RECONSTRUCTING AND CORRECTING 3D BUILDING MODELS USING ROOF TOPOLOGY GRAPHS

Biao Xiong

Examining committee:

Prof.dr. M.J. Kraak	University of Twente
Dr. N. Kerle	University of Twente
Prof.Dr.-Ing. M. Kada	University of Osnabrück
Prof.dr. M.J. van Kreveld	University of Utrecht

ITC dissertation number 264
ITC, P.O. Box 217, 7500 AA Enschede, The Netherlands

ISBN 978-90-365-3810-7
DOI 10.3990/1.9789036538107
Cover designed by Biao Xiong
Printed by ITC Printing Department

ITC
UNIVERSITY OF TWENTE.
FACULTY OF GEO-INFORMATION SCIENCE AND EARTH OBSERVATION

RECONSTRUCTING AND CORRECTING 3D BUILDING MODELS USING ROOF TOPOLOGY GRAPHS

DISSERTATION

to obtain
the degree of doctor at the University of Twente,
on the authority of the rector magnificus,
prof.dr. H. Brinksma,
on account of the decision of the graduation committee,
to be publicly defended
on Friday 5 December 2014 at 12:45 hrs

by

Biao Xiong

born on October 10, 1985

in Jiangxi, China

This thesis is approved by
Prof.dr.ir. M.G. Vosselman, promoter
Dr.ir. S.J. Oude Elberink, assistant promoter

To my parents **Meilan Nie and Jingen Xiong**
and my sister **Kuan Xiong**

Summary

The reconstruction of building models has been heavily researched in the field of photogrammetry and computer vision. The two most used types of input data are lidar points and images. The goal of this thesis is to find an automatic method to reconstruct Level of Detail 2 (LOD2) building models from airborne lidar scanning data. A LOD2 building model contains distinctive roof structures and larger building installations like balconies and stairs. The state-of-the-art approaches can automatically reconstruct about 80% buildings, mainly in residential areas. This means a lot of complex buildings still cannot be automatically reconstructed. This PhD research is to increase the automation rate and provide a methodology which could be used to reconstruct buildings at a national scale. In order to achieve this goal, a sequence of problems has been solved. The two main contributions are:

1. ***Flexible building primitives*** have been proposed to solve the problem that many complex buildings cannot be represented by the state-of-the-art building primitives. The building primitives are key elements in the model-driven methods to reconstruct buildings. However, the current building primitives are all shape-limited. An open problem in the literature is whether there is a set of building primitives that is compact but flexible enough to represent complex buildings? Based on the finding that the roof topology graphs have basic sub-graphs, each of them represents a type of building primitives. Because any roof topology graph can be decomposed into such basic sub-graphs, building primitives based on them are able to represent any type of building. Meanwhile, the sub-graphs are non-redundant, thereby, the building primitives are compact.
2. ***A graph edit dictionary*** has been introduced to automatically correct the errors in roof topology graphs. A roof topology graph is the adjacency topology graph of roof patches of a building. Roof topology graphs are essential for inferring building structures and searching for building primitives. However, because of noise, data deficiency, and imperfection of pre-processing steps, roof topology graphs inevitably have errors. In order to improve the automation, the errors should be automatically corrected. By analysing over 1000 buildings, errors are found to be repeating. By learning the repeating errors, a graph edit dictionary, which stores pairs of errors and correction rules, is constructed to find and correct the errors in roof topology graphs.

In order to solve the problems accompanying the two main contributions, several important contributions have been made:

1. ***Several Shape regularities*** of buildings are introduced in addition to summarizing and analysing the regularities used in the literature. As man-made objects, buildings have strong shape and structure regularities. However, they are not equally important and cannot be used

in all situations. This thesis analyses the characteristics of those regularities, their advantages and disadvantages, and their application situations.

2. ***Efficient interactive tools*** have been designed for correcting roof topology graph errors. The proposed automatic method based on graph edit dictionary can only correct part of the errors. In order to reconstruct all building models, the remaining errors should be corrected. Additionally, the graph edit dictionary is constructed based on learning errors in the roof topology graph. The correct model should be provided before the learning. The interactive tools enable us to reconstruct building models efficiently.
3. ***Adaptive Constraints*** are applied to improve the model quality. The buildings are decomposed into building primitives, which contain constraints for guiding the reconstruction. Constraints are not forced obligatorily. A decision is taken as to which constraints are necessary and which constraints could be left aside based on both inferred building primitives and input data.
4. ***A single quality criterion*** is proposed for evaluating a complete building model without the use of reference building. The building models could only be used when their quality is accepted. However, it is difficult and expensive to achieve reference data for 3D building models of large area. The proposed criterion is based on the comparison of input lidar points and reconstructed building models.

By progressing step by step, this PhD research provides methods that can automatically reconstruct about 95% acceptable building models. The other 5% buildings can be reconstructed manually efficiently by the developed interactive tools. The building models are available for large scale applications. The most common applications will be urban planning, tourism, navigation, security, training, and visibility analysis.

Samenvatting

De reconstructie van gebouwmodellen is een welbekend onderwerp in het onderzoeksveld van de fotogrammetrie en computer vision. De twee meest gebruikte databestanden om de modellen te maken zijn laseraltimetriegegevens en luchtfotos. Het doel van dit proefschrift is een automatische methodie te presenteren die 3D gebouwmodellen genereert op basis van laseraltimetriegegevens. De gebouwmodellen worden met een hoge mate van gedetailleerdheid gegenereerd en bevatten dakvormen en elementen zoals dakkappelen en balkons. De bestaande methodes kunnen tot 80% van de gebouwen correct modelleren, vooral in woonwijken is het succespercentage hoog. Dat betekent dat vooral complexe gebouwen nog niet gereconstrueerd kunnen worden. Dit PhD onderzoek heeft als doel om het succespercentage te verhogen, en om een methode te maken die op een landelijke schaal ingezet zou kunnen worden. Om dit doel te bereiken, moeten er een aantal problemen opgelost worden. De twee belangrijkste bijdragen van het proefschrift zijn:

1. De inzet van ***flexibele gebouwprimitieven*** zorgt ervoor dat zelfs complexe gebouwen gemodelleerd kunnen worden. Gebouwprimitieven zijn belangrijke bouwstenen in modelgedreven reconstructiemethoden. Het probleem is dat de meeste primitieven een vaste vorm hebben, waardoor variatie in gebouwvormen lastig aan te brengen is. De vraag is dus of er een methode bedacht kan worden die flexibel genoeg is om complexe gebouwen te kunnen modelleren. Een manier om dat te bewerkstelligen is gebruik te maken van daktopologien. Zo’n daktopologie hebben we beschreven als een combinatie van simpele gebouwprimitieven, zoals een enkel dakvlak. Deze simpele gebouwprimitieven hebben we als uitgangspunt voor gebouwprimitieven genomen, omdat ze zowel eenvoudig als flexibel genoeg zijn om allerlei complexe gebouwen te modelleren.
2. Een zogenaamde “***graph edit dictionary***” zorgt ervoor dat foutjes in de topologie van dakvlakken worden opgespoord en gecorrigeerd. De topologie van de daken is een beschrijving van hoe dakvlakken aan elkaar grenzen. Deze daktopologie is handig om de structuur van het dak, en dus het dakmodel, te herkennen. Helaas kan de daktopologie niet altijd correct gevonden worden in de laserdata, ofwel door ruis in de data, of door verkeerde aannames in een van de data verwerkingsstappen. Het is belangrijk dat deze foutjes automatisch opgespoord kunnen worden. Dit is gedaan door eerst handmatig meer dan 1000 gebouwen te analyseren, en de computer te leren welke foutjes nu aanwezig zijn. Op deze manier is een “graph edit dictionary” opgebouwd die de combinatie opslaat van hoe het foutje eruit ziet en hoe het er eigenlijk uit zou moeten zien. Dus, deze ‘dictionary’, letterlijk een vertaalwoordenboek, vertaalt de manier waarop een gebouw er in

de data uitziet en hoe het er eigenlijk uit zou moeten zien om er een correct 3D model van te maken.

Om bovenstaande mogelijk te maken, zijn de volgende stappen uitgevoerd:

1. Een ***aantal repeterende*** patronen in gebouwvormen zijn geïntroduceerd naast reeds bestaande oplossingen uit de literatuur. De meeste gebouwen vertonen een min of meer regelmatig patroon, bijvoorbeeld qua symmetrie. Alleen is soms het ene patroon belangrijker dan andere. In dit proefschrift worden de verschillende karakteristieke patronen geanalyseerd op voor- en nadelen en hoe ze gebruikt kunnen worden in de gebouw modellering.
2. ***Efficiënte interactieve*** methodes om fouten in de daktopologie te corrigeren. De graph edit dictionary kan een deel van de fouten automatisch corrigeren. Het overige deel moet op een andere, niet automatische manier opgevangen worden. Dat gebeurt op een interactieve manier waarop tegelijkertijd de graph edit dictionary aangevuld wordt. Deze interactieve methodes maken het mogelijk om efficiënt gebouwen te modelleren in 3D.
3. ***Optionele geometrische*** voorwaarden worden toegepast om de kwaliteit van gebouwmodellen te verbeteren. De modellen zijn opgebouwd uit meerdere gebouwprimitieven, die elk geometrische voorwaarden bevatten om de dakvorm te beschrijven. Deze voorwaarden worden niet verplicht gesteld, maar optioneel. Er wordt gekeken naar welke voorwaarde noodzakelijk zijn, en welke overbodig. Dit wordt getoetst op zowel het uiteindelijke 3D model als de input laser altimetriegegevens.
4. Een ***eenduidig kwaliteitscriterium*** wordt geïntroduceerd om de kwaliteit van elk 3D model te beoordelen zonder dat er referentiemateriaal voor nodig is. Op basis van dit kwaliteitscriterium wordt bekeken of het 3D model goed genoeg is of niet. Het criterium is gebaseerd op de vergelijking tussen de input laseraltimetriegegevens en de geproduceerde gebouwmodellen.

De onderbouwing en de implementatie van bovenstaande stappen is beschreven in dit proefschrift. Aan de hand van deze stappen is het mogelijk om 95 % van alle gebouwen automatisch te reconstrueren. De overige 5% kan efficiënt met de interactieve methodes gemodelleerd worden. Dit maakt het mogelijk om op landelijke schaal gebouwmodellen te produceren. De belangrijkste toepassingen zijn te vinden in stedelijke planning, toerisme, navigatie, veiligheid, training en zichtbaarheidsanalyses.

Acknowledgements

Sitting in front of my laptop, which has already companied me four years, and is almost old enough to retire, enjoying the sunshine outside the window and the ten degree temperature in August, I am finally prepared to finish my thesis and look back to my Ph.D life in Enschede. But I still do not know where to start after tried to start writing many times. The last four years are either too long, or too short, but are really very important for me. I have experienced a lot, thought up a lot of ideas, and also have been helped by many. When looking at the new students and colleges, I clearly remember I, just like them, were full of curiosity and ignorance about the new life in the Netherlands and ITC. I still clearly remember my idea at that time: try my best to climb over the giant's shoulder and to create a new space with my own hands. I thought lidar is very near to be widely applied. Before that, several crucial technical problems should be solved. Automatic modelling of 3D buildings is one of them, and is the high ground: if we can solve it, other research problems and industrial applications will be promoted strongly. I was so lucky that I can start research on 3D building modelling in such a good environment. I must cherish this opportunity. If I have achieved something, I cannot make it without helps from a lot of people. Here I would like to give my special thanks to some of them.

George, you are the most knowledgeable, rigorous, and efficient person I have worked with until now. Without your hard work, I cannot progress any inch and publish any word. I can always learn more from you. You go very deep to problems and ask questions to the point. You have built up a perfect research framework on cutting-edge scientific problems. You push me to clearly express myself, and to prove my idea with strong experiments, though it is really hard to convince you. You are always my first reviewer, first customer and the first step leading to the world. I get the confidence that if I can convince you, I can convince the whole world.

Sander, you are so kind, patient, and helpful, at any time when I knock your door, even with really tiny and stupid problems. Your helps on my scientific writing are especially impressive to me. Prof. Wolfgang Förstner, even though I only emailed several times, your accurate and profound suggestions helped me out of the graph correction problem that I had pondered more than half year. Without you, I definitely cannot move on that quickly.

Dear dad, mom, and sister, without your supports, I even cannot go do Ph.D out of China. Every time I called you, I felt so happy and warm in deep of my heart. Dad, it is your hard shoulders to support the whole family, and to support sister and me to get high education. Mom, you always teach sister and me to try our best, do the best at any position, and never be satisfied. Your personal example as well as verbal instruction makes me who I am now.

Sister, you are always the guide in every step of my life. I learnt from you that where should I go, and how I could reach there quickly with less detours. Because I have you to take care of our parents, I can freely brave the journey, and take risks.

Yang Dong, you are the one with who I can share all ideas and problems that I have. You are always there to support me, to encourage me, to make me brave, and to make me confident. All my naive ideas that became matured in the last four years were under the discussion with you. Without you, I do not know how to persevere in the last four years.

Markus, Kourosh, your comments improves my work all the time. Adam, you always act as my big brother, teaching me how to smile loudly and face the hard of life. Bashar, you show me the responsibility of man and father. Emily (Xiao Jing), Zhou Liang, you demonstrate me how to live in a foreign country and how to balance work and life. Adugna, lets share more secretes, boys secretes, between us :D. Terresa, Leos, your patient help always makes my life easier. I would also like to say many thanks to all other EOS colleges, including Francisco, Salma, Julia, Mila, Anand, Mu Jingfeng, Li Mengmeng, Tang Feifei, Feng Yan, and so on. The discussion and chatting with you always inspire and rejoice me.

Xu Sudan, Wu Xiaojing, Ye Xinping, Zheng Donghai, and Huang Ying, it is really very nice to share the Ph.D journey in ITC with you. My Ph.D life is never lonely with you. Sudan, your brave in front of hard problems always encourages me. Xiaojing, five years ago, we together started apply the positions, went through a wide range of complex and various procedures, and even took the same airplane to the Netherlands.

Chinese community is the big and warm family for me. I would like to thank all my Chinese friends I met in ITC and with whom I enjoyed an unforgettable time. It is not possible to list you all here, please accept my sincere appreciations if you are not mentioned here: Wang Tiejun, Su Bobo, Si yali, Chen xuelong, Zeng Yijian, Zhang Xiang, Zhao Xi, Chen Fangfang, Hao Pu, Yu Fangyuan, Xiao Wen, He fangning, Han qifei, Jiang yanbing, Yang Fengfan, Tan Chengxiao, Gu Fengchao, Li Xiaoxu, Wu Gaoyan, Yuan Xu, Lv Shaoqing, Cui Mengyi, Jing Ying, Zhang Kaixi, Li Lingling, Wang Zhihui and so on. A lot of good memories bear in my mind: happy parties, warm Chines festivals, exciting travelling, and nice cooking and so on. They contribute to my beautiful life in the last four years. I would further express my special thanks to my good friend, Gao Zhihong, Fu Dongjie, Mao Feiyue, Li Xuanqi, and Wang Lanjun, for their thoughtful discussion, invaluable suggestions and everlasting friendship.

Table of Contents

List of figures

List of tables

Chapter 1

Introduction

From the time when human started to record information on papyrus, parchment roll, and silk, we have been trying to record the 3D world on the 2D media. A simple but essential reason is that we are living in the 3D world, not on a 2D planet. Many technologies have been invented for this purpose, including painting and mapping technology. Only until the information era came, we were able to store, manage, and use the information of the 3D world easily. Digital 3D worlds play an important role in fields like flood risk management, infrastructure constructing and management, water supply and quality, nature resource conservation, geologic resource assessment and hazard mitigation, navigation, and so on (Snyder, 2012). The lidar (LIght Detection And Ranging) technology, which provides accurate and direct 3D information, enables us to collect accurate 3D information. Moreover, the capability in vegetation penetration makes it the perfect method to measure objects under vegetation, e.g., ground. The flexibility of lidar provides multi-angles and multi-distances measurements, including satellite-based, airborne, terrestrial, handheld laser scanner. Laser scanners are developing fast in the last 10 years, providing high density, high accuracy, and full waveform data.

Algorithms for processing lidar data and reconstructing 3D maps, however, are delayed behind the hardware development of data acquirement technology. In fact, 3D reconstruction has become the bottleneck of the widely ranged application of high quality point clouds. Because buildings are the main part of cities, their reconstruction has drawn huge attention to researchers from photogrammetry and computer science. Though many scholars have done research on this topic and have acquired impressive results in the last two decades, they are still restricted to relatively simple buildings. Many complex buildings are left to be manual reconstructed. General and relatively simple buildings can be automatically reconstructed with the recent technologies. Nevertheless, architectures that have complex structure or incomplete data are still difficult to reconstruct and become a hot research topic nowadays. Due to the high percentage of those complex buildings, modelling of big city now is still expensive and time consuming. Automated algorithms that can deal with high quality data are in demand for application requirements

Building shapes are regular but as well diverse. Comparing to natural objects, e.g. trees, buildings are man-made objects and have their own shape regularities. But when comparing to other man-made object, e.g. lamp, traffic lights, buildings do not have that many repeat patterns. Those shape regularities can help us to interpret and reconstruct buildings. Knowledge based algorithms are the best solution for automatic reconstruction of 3D building. Human operators could immediately place and integrate buildings from mass data because they have rich knowledge about buildings and city environments. This knowledge and experience includes geometry topology, semantic information. The human brain can integrate this knowledge in different data processing stages and even improve the knowledge database by adding new discovered information of buildings. In the interpreting process, observed objects are matched with models and knowledge, which are learned and stored in brain. Due to the interaction of reconstruction and interpreting process in the intelligent system, research on utilization of knowledge on 3D model acquirement will not only improve reconstruction of 3D architecture model but also deepen the understanding of human reasoning.

1.1 Research problems

Although building reconstruction techniques have dramatically improved recently (Brenner and Von Goesseln, 2004; Haala and Kada, 2010; Hu et al., 2003; Musialski et al., 2013; Vosselman, 2002; Wang, 2013), many problems still prevents it from comprehensive utilization. Those problems stem from the characteristics of laser scanning data as well as the complexity of buildings and environments. It is helpful to list and discuss problems for determining further research.

a) Complex scene

Various kinds of objects exist in city scene, including vehicles, pedestrians, trees and street lamps. The first step of building reconstruction is to distinguish building points from other points. A 2D topographical map is very useful to complete this task. However, topographical maps are not always accessible. In many situations, we need to complete building reconstruction without a map. On the other hand, objects neighbouring to buildings are important context information that could be used to categorize building and refine building model. For instance, ground planes could be employed to determine footprint of building.

b) Lack of data

The laser scanning data is often imperfect for a building because of its surface material and environment. A laser scanner cannot receive data from roofs and walls covered by non-reflecting materials. For example, a laser beam is transmitted through glass, absorbed by slate and water surface, and mirrored by smooth surfaces. On the other hand, some surfaces have steep slopes or are occluded by other objects. So they cannot be scanned by laser scanner or the scanning data are sparse. Those buildings cannot be reconstructed by detecting and assembling surfaces. In addition, a large part of vertical walls is partly scanned, so the reconstruction of those building walls depends on 2D maps. For example, no ridge line could be detected from lidar points of the churches shown in Figure 1-1.

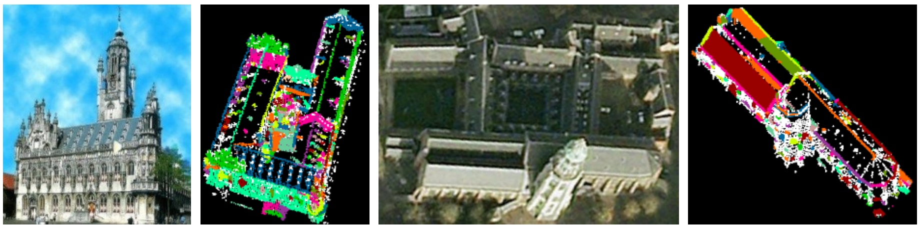


Figure 1-1. Buildings with many lidar points lost.

c) Complex building shapes

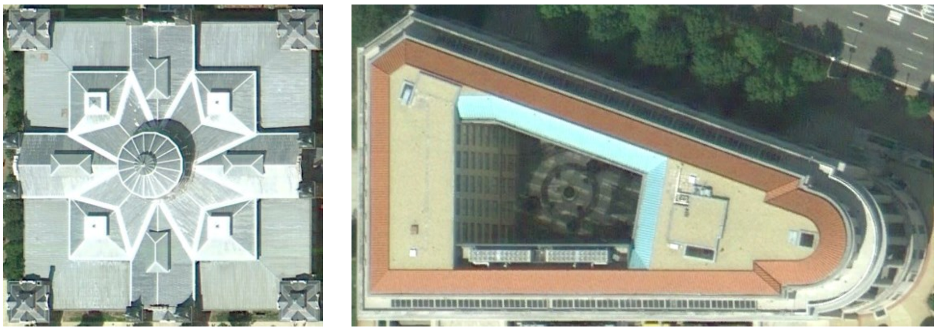


Figure 1-2. Buildings with complex shape.

Many buildings have complex shapes, which are difficult to be abstracted into a topological database. A part of buildings is constructed by curved surfaces, e.g. most large gymnasium use curved surface as roof. In addition, building surfaces constructed by planes also have manifold structures. Topological graphs are employed to introduce constraints to improve reconstruction results. However, these topological graphs are hard to be enumerated and pre-stored in knowledge database. As shown in Figure 1-2, the building shapes could not be represented by assembling primitive building shapes. On the other hand, if a larger number of topological graphs were listed, searching target graph for one specific building would be more difficult. One object could be matched with more than one graph.

d) Complex Boundary

Edges connected by two faces are stable and simple to detect. Their parameters are fixed by intersecting the two adjacent faces. Nevertheless, other edges are hard to fit due to discreteness and noise of the lidar points near the building boundary. The working manner of laser scanner makes the laser scanning points near the building boundaries unstable. The error of pre-processing, except data noise, such as filtering and segmentation, will result in a poor boundary description by the lidar points. As we can see from

Figure 1-3, it is hard to detect the boundary shape and fit boundaries for the segment marked by purple points. As a result, we need extra information on the geometry of the outer edges to help determine the roof face boundaries.

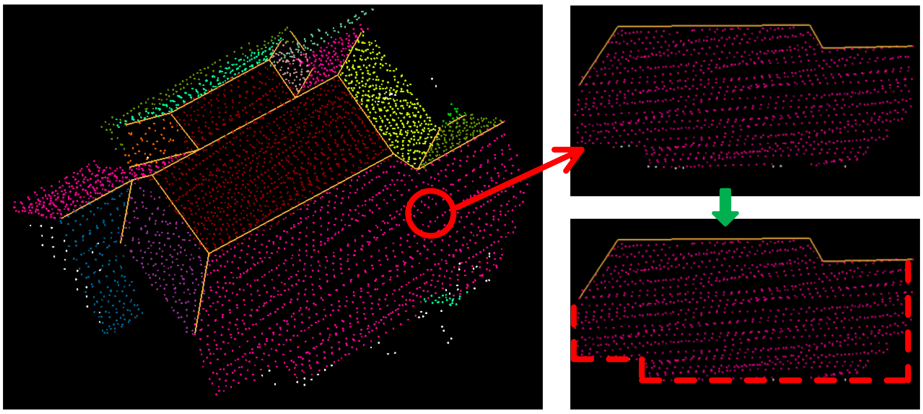


Figure 1-3. Complex outer boundaries.

1.2 Goal and objectives

This PhD research is to provide an automated approach that reconstructs Level of Detail 2 (LOD2) building models from airborne laser scanning data. A LOD2 building model has distinctive roof structures and larger building installations like balconies and stairs (Kolbe, 2009). This research will focus on representation, learning, and utilization of knowledge on building shapes for 3D building model acquirement. It will first try to find the grouping type and dependences between geometric elements and regularities behind building structure. It will attempt to develop strategies on how such dependences and regularities are used in automatic algorithms for 3D building model acquirement.

The general goals are narrowed to the following objectives:

- To find a set of building primitives that is compact but flexible enough to represent arbitrarily complex buildings.
- To define measures to evaluate building model quality. Considering the current situation that ground truth data is hardly available, the measure should not dependent on the ground truth data.
- To design and implement an automatic method that can correct erroneous building models.
- To provide a set of efficient tools to manual correct errors in reconstructed models.

1.3 Building knowledge

Building shape knowledge mainly includes geometry, topology and semantic information. This information can be transformed into reconstruction rules and introduce constraints for building modeling. Therefore, in this project, the first work to be done is to explore which knowledge about buildings could be used for building reconstruction and how to represent and acquire such knowledge. When the human brain interprets data to recognise buildings, it will inference all information from low-level data information and translate this information into reasonable buildings to match the original data. This information may be as follows:

Age: Buildings in different ages have different structures and styles. Citizens design the architecture according to the economic condition, fashion, culture, and other factors. According to the building age, there are generally several building styles: Romanesque style, Gothic style, Baroque style, Rococo style, and so on. Every building style has its own features. E.g., Gothic style has Latin cross plane, the pointed arch, high height, and rose windows. Romanesque architecture is famous with its massive quality, its thick walls, round arches, sturdy piers, decorative arcading, and so on. In a short period, buildings also may have different types. For instance, buildings, which were built before and after World WarⅡ, have significantly different characters in the Netherlands.

Usage: Buildings could be classified into different styles by its usage, for example, industry, residences, comitial, sports, religious, education, office, and so on. Generally, a residence building has a relative small area and two or three floors, while office buildings are larger and higher then residence buildings. A religious building often has a cross roof and tower.

Street pattern: A city area is segmented into different plots by streets. Buildings are placed on these plots so their positions, directions form to street patterns. Besides, buildings in same area have similar structure and heights. By analysing street pattern and its neighbour buildings, one building’s coarse position, direction, height and structure could be detected (Figure 1-4).

Figure 1-4. Building styles vary with time, culture, location. (dudviscomm, 2014)

Given building categories and building models, we can learn statistical data and mine building knowledge. By acquiring more and more buildings, the set of buildings from which the general knowledge on building shapes can be inferred will gradually expand. This could then lead to an increasingly reliable hypothesis generation to support further building model acquisition. The vital information about building style is height, roof type and slope, facade elements types and topology graph, and so on. In the reconstruction process, features, characteristic information, are acquired from the point cloud and sent to trained classifiers to determine the building type. Then the structure pattern for a specific building type could be used to reconstruct architectures. Building knowledge should also be learned from experts in the area of architecture and computer graphics.

1.4 Dissertation Outline

This dissertation consists of six chapters. The two core chapters (4 and 5) focus on the two main problems and have the main scientific contributions.

Chapter 1 first described the reason behind the selection of the research topic. The concepts used, the objective and related research questions were presented. Lastly, the outline of the dissertation was given.

Chapter 2 gives a literature overview on building reconstruction and procedural modelling, including roof, façade, indoors, from diverse source data. The aim of this review is to find the geometry regularities behind the object surfaces and determine when to use which regularities in building model reconstruction.

Based on the overview given in Chapter 2, Chapter 3 summarizes the regularities and suggests the suitable situations where and which regularities could be applied and the factors which need to be considered. This chapter also propose several new regularities and discuss three ways to represent building structures. Chapter 3 gives theory foundations for the methods shown in following chapters.

Chapter 4 presents flexible building primitives which are able to represent any complex building. This chapter also solves several related issues that should be considered to use the new building primitives for building reconstructions, including primitive search, constrained least-squares fitting, outline enclosing, and the manual correction of roof topology graphs. This chapter ends with the experiments on testing the building reconstruction algorithm based on the developed flexible building primitives.

Chapter 5 extends the work shown in Chapter 4, which uses roof topology graphs for searching building primitives. Roof topology graphs often have errors, which seriously influence the reconstruction results. Chapter 5 gives an automated method, which is based on a graph edit dictionary, to correct these errors. Accordingly, this chapter presents the definition, construction and application of the graph edit dictionary. Experiments are also presented to test the graph edit dictionary and demonstrate the improvement in automated 3D building modelling.

Chapter 6 concludes the works of this dissertation and give recommendations for further research directions.

Chapter 2

Literature review

This chapter gives a literature review of building reconstruction and procedural modelling, including roof, façade, indoors, from diverse source data. The aim of this review is to find the geometry regularities behind the object surfaces and determine when to use what regularities in building model reconstruction. There are several review papers on the building and urban reconstruction methods (Brenner and Von Goesseln, 2004; Haala and Kada, 2010; Hu et al., 2003; Musialski et al., 2013; Vosselman, 2002; Wang, 2013).The methods of building reconstruction and modelling are fundamentally different according to the level of model details and data source. However, the research objects, buildings, and their geometry and topology behind do not change. By reviewing the literature the author would like to acquire a simple but uniform knowledge (rules, or regularities) that can be used to model any building with LOD 2 model. This review only considers the reconstruction algorithm of building geometries, therefore the detection of buildings are not discussed, as well as the acquisition of the input data, like DEM from stereo image pairs, SAR, lidar, and 3D lines and 3D point clouds from image matching.

The assumptions about the regularities of building geometry are different when considering the quality of goal model and input data. Normally, the lower quality of input data requests higher level of assumptions, while the lower model quality needs looser geometry assumptions. No matter what kind the input data is, we should keep in mind that we are reconstructing and modelling the same objects: the architectures. The works can be classified according to the used input data, goal objects, and geometry assumptions. As the thesis is to reconstruct roofs from airborne data, the literature review will put more emphasis on roof reconstructions.

Table 2-1 . The types of input data, goal objects and geometry assumptions that are used in the object model reconstruction.

Input data	Single image	Stereo images	lidar	KINECT	SAR	2D map
Goal objects	roof	façades	indoor scene		small objects	
Geometry assumptions	hierarchic	line and face alliance		templates	Simple	region

Table 2-1 gives input data types, goal objects and geometry assumptions that are usually commonly used in the literature. The objects mentioned here are mainly buildings, including roof, façade, and indoor, according to the level of details. Some small objects, like industry products and furniture, are skipped, because their reconstructions are very similar to buildings and some ideas and technologies can be borrowed to reconstruct buildings. The geometry assumptions chosen here are those frequently used in literature. Hierarchical regularities are defined to describe the decomposition and grouping relationship of objects. The line and face regularities are about the geometry regularities of single or a group of lines and faces. Templates provide uniform shapes for objects within a same type. The objects are modelled by matching a predefined geometry template to the data. The region regularities are the most basic characters that the valid geometry shapes need to meet.

The input data include single images, stereo images, lidar, RGBD (from Kinect), SAR and topographic maps. Normally, a building reconstruction method uses several types of input data and combines different geometry assumptions. But topographic maps are usually used as auxiliary data, therefore could not be used individually. Stereo images can be used to derive several different data source, including digital elevation model (DEM), photogrammetric point cloud and 3D line segments. Each of them could be used for object reconstruction independently or combined with other data. Though images can be applied to reconstruct all kinds of objects, some data is only for one, depending on data characters. E.g. Kinect only has 5m effective detection range and is sensitive to light conditions therefore could not be used in outdoor modelling, while SAR is designed to be installed on airborne aircraft, therefore only suitable for outdoor modelling. The lidar data could be acquired by different scanners, airborne, terrestrial or hand-hold scanners. Therein, they are used for modelling different objects, airborne lidar for roof and façades, terrestrial lidar for façades and indoor scene and hand-hold lidar is for industry objects.

Procedural modelling uses a lot of building knowledge and can create near real city scenes (Kelly and McCabe, 2006; Vanegas et al., 2010; Watson et al., 2008). Therefore procedural modelling is also discussed in this chapter to analysis the possible building knowledge that can be used in the reconstruction. To synthesize a virtual city is easier than to reconstruct a city which is already there. Because the synthesis only needs to model a reasonable scene but the model does not have to coincide with the actual situation. However, the rules used in the synthesis should coincide with the reality. We need to refine the rules so that they can be used in reconstruction. The procedural modelling has very few interests on roof structures. It may be because a simple CSG combination of a set of simple roof shapes is already enough to simulate divers building types for visualization. However, the synthesized roofs are not diverse enough to represent the real buildings.

The model reconstruction by mesh models and the procedural modelling are first discussed because they are widely used in other methodologies, like roof, façade and indoor modelling. The mesh model and its optimized model do not get enough emphasizes in the photogrammetry community. It is generally considered that they are too simple to need research. The mesh model is simple and flexible to represent arbitrary structures. However, it is still difficult to construct a complete mesh model from noisy and incomplete point cloud and to simplify and optimize the mesh model. On the other hand, the ideal simplified mesh model is a B-rep model. Therefore this chapter discusses the reconstruction of mesh models in detail.

The other literature is divided according to the goal objects that the reconstruction algorithms aim at. The goal objects dramatically affect the selection of structure regularities and reconstruction strategies. Then each group is partitioned based on input data types. This organization enables a clearer understanding of the reconstruction algorithms. If the organization is only based on the input data, the reconstruction methods have large overlap. For example, the roof reconstruction from DEM and aerial lidar data sometimes are mutually used, and the indoor modelling from Kinect data, photogrammetric point cloud and terrestrial lidar data as well are almost equal. Though the input data may have large differences, the same goal objects make the selection of patterns and regularities of buildings to be almost the same. Even though a proper organization is chosen, there are still some works can locate in several groups. Those papers are categorized to the most related group. The literature can also be divided according to the geometry assumptions. But one geometry assumption can be used in the reconstruction of different objects from different input data.

2.1 Procedural modelling of urban scene

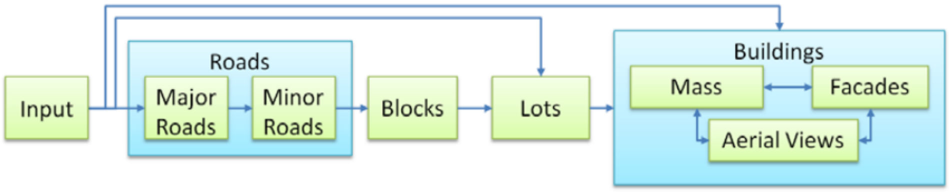


Figure 2-1. General pipeline for modelling urban spaces. (Vanegas et al., 2010)

Computer graphics practitioners have successfully used procedural modelling to generate non-urban and urban contents. Kelly et al. (2006), Watson et al. (2008), Smelik et al. (2009) and Vanegas et al. (2010) give overviews of the state of art on generating urban environment, terrain modelling (Figure 2-1). In non-urban content, L-system grammars generate plants (Prusinkiewicz et al., 1988), while agent-based particle systems (Reeves, 1983) model fuzzy objects such as fire and smoke. Perlin’s noise (Perlin, 1985) simulates clouds and natural textures, while Reynolds apply agent-based methods to animate flocks, schools, and herds (Reynolds, 1987).

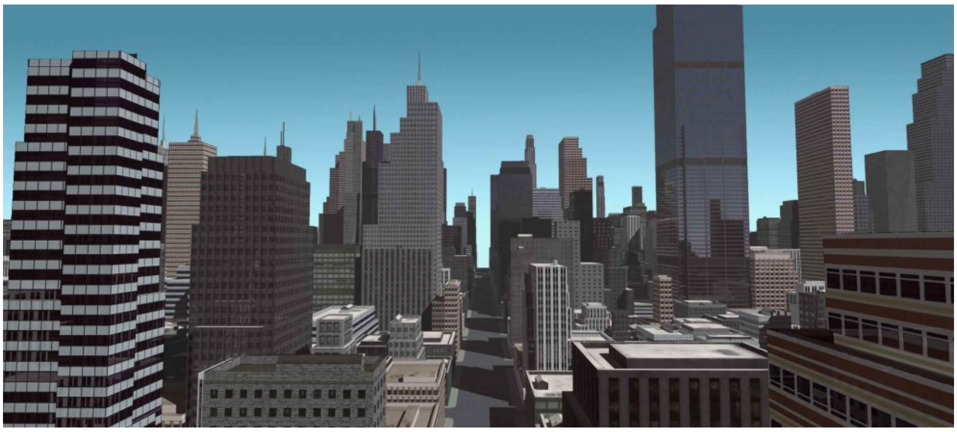


Figure 2-2. A city scene procedural modelled by CityEngine. (Parish and Müller, 2001)

Procedurally generation of cities is commonly starting from creating a dense road network and afterwards identifying the polygonal regions, which are enclosed by streets. The process of city generation could be divided into several stages. A variety of methods can be used to generate road networks for cities, e.g. L-systems(Parish and Müller, 2001), agent simulations (Lechner et al., 2003) and tensor fields (Chen et al., 2008). Vanegas et al. (2010) generate roads based on weighted anisotropic shortest path algorithm by considering the slope of the terrain, natural obstacles. Polygonal regions surrounded by streets are subdivided into lots, which determine the architecture position. In order to enrich these lots with buildings, the lot shape can be used directly as the building footprint. Another way is to fit a building footprint on the lot. A city with skyscrapers and office buildings can be generated by simply extruding the footprint to a random height. To obtain building shapes more detailed, several approaches have been invented.

Greuter et al. (2003) generate a floor plan by combining several primitive 2D shapes and extrude them to different heights, and then generate a complete building model by combining several floor sections. Parish and Müller (2001) apply an L-system to refine the building iteratively based on rectangular floor plans (Figure 2-2). Merrell (2007) generates buildings (and many other shapes) using a texture-synthesis-inspired technique. Cabral et al. (2009) reshape and combine existing textured models to model architectural scenes. They manipulate the geometry and texture in a tightly coupled way. Whiting et al. (2009) procedurally model realistic structural buildings considering physical interaction.

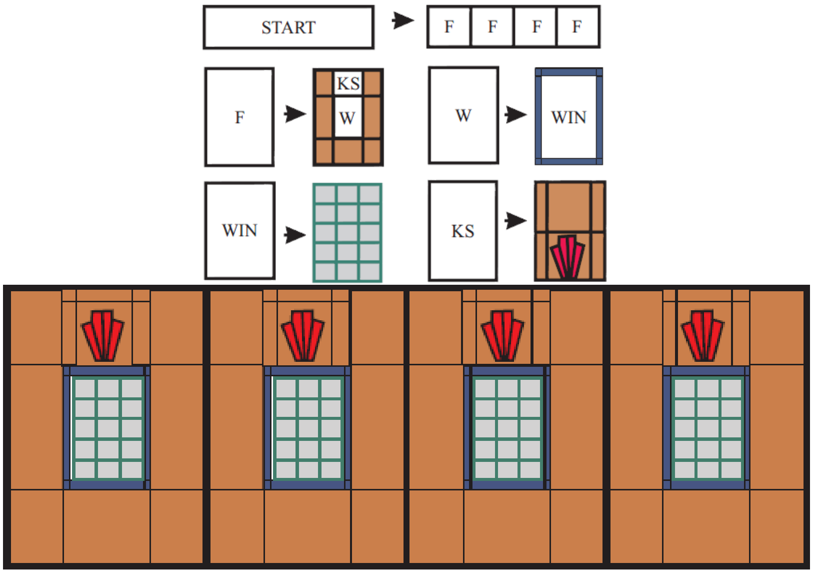


Figure 2-3. The split grammar of façade (Wonka et al., 2003).

Creating detailed façades is of key importance to simulate buildings when perspective is near the building. After generating mass model of buildings, many researchers introduce façade grammar for synthesizing building façades as the next step. Wonka et al. (2003) iteratively split a façade into detailed façade elements (Figure 2-3). Larive and Gaildrat (2006) use a similar grammar, called a wall grammar, which enable them to generate building walls with additional details, such as doors, windows and balconies. Müller et al. (2006) apply context-sensitive rule grammar, named CGA (Computer Graphic Architecture) shape grammar.

Automatic creation of indoor layouts and complete building models is also of high interest for computer graphics and agriculture planning. Merrell et al. (2010) stochastically construct floor planes given a set of high-level of requirements and use them as inner structure to construct a complete 3D building (Figure 2-4). Leblanc et al. (2011) procedurally generate complete buildings with coherent interior and exterior by combining spatial and semantic components. Tutenel et al. (2011) offer a framework to create buildings with consistently inter-related exteriors and interiors making use of a library of semantic objects and constraints. Chojnacki (2012) provides custom procedural mechanism based on a scoring function to automatically arrange and create business interior models.

Though aiming at creating virtual environment, a large set of regularities of real world are applied. The synthesized objects are so real that a man cannot see that they are artificial. Those regularities can also be used to reconstruct the real world, even though hard.

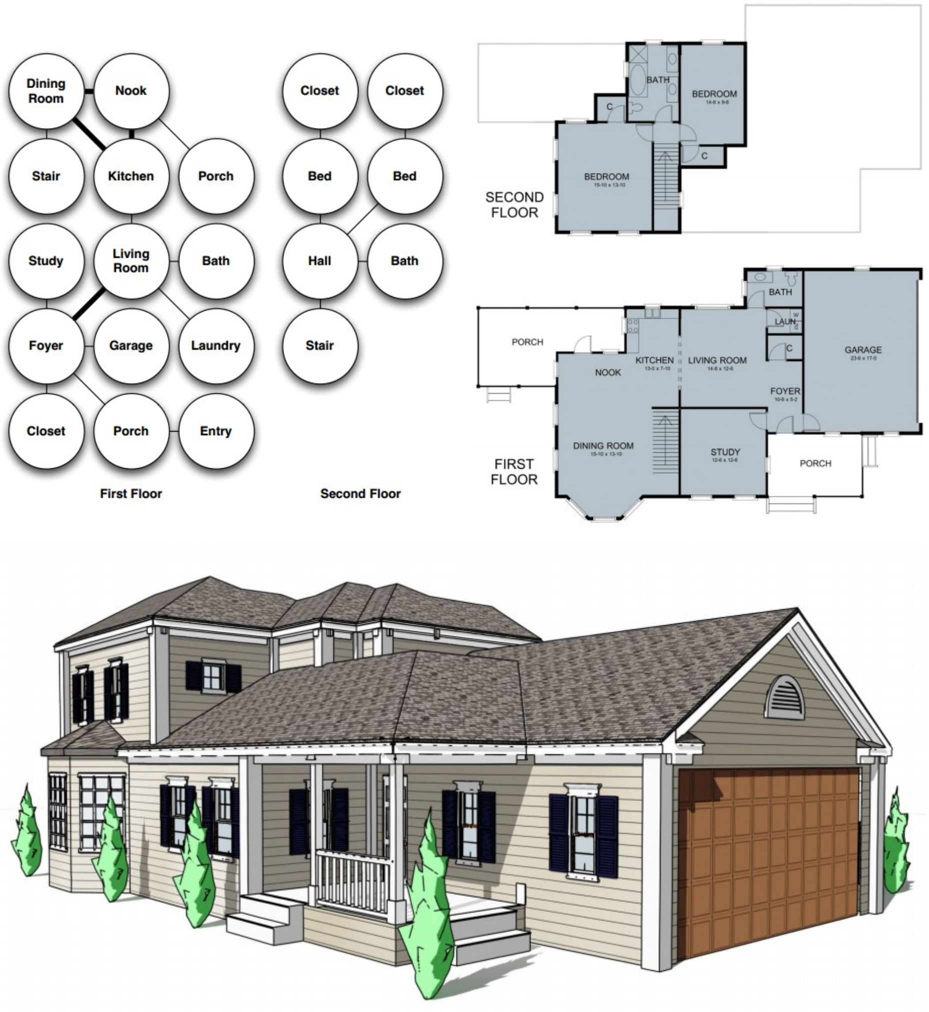


Figure 2-4. Stochastical construction of floor plans given a set of high-level of requirements and use them as inner structure to construct a complete 3D building. Merrell et al. (2010)

2.2 Reconstruction of mesh models

Surface meshes are commonly used to represent shapes in many computer graphics applications. These meshes are usually generated by scanning devices. Such processes, however, are inevitably erroneous, and the resulting raw meshes are hardly acceptable. Most of the time, they are oversampled and contain many redundant vertices. This complexity stimulates a considerable amount work of mesh simplifications. On the other hand, the scanning pattern is hard to be adjusted to coincide with sharp edges and corners of the model, which are therefore removed by the sampling process. The sharp features are replaced by irregularly triangulated chamfers in the final model. The mesh simplification needs to reserve the sharp features. When considering transforming redundant triangle mesh into a B-rep model, leaving the minimum number of triangles, the mesh simplification is equivalent to the reconstructions of polyhedron models, which are usually the aim of building reconstruction.

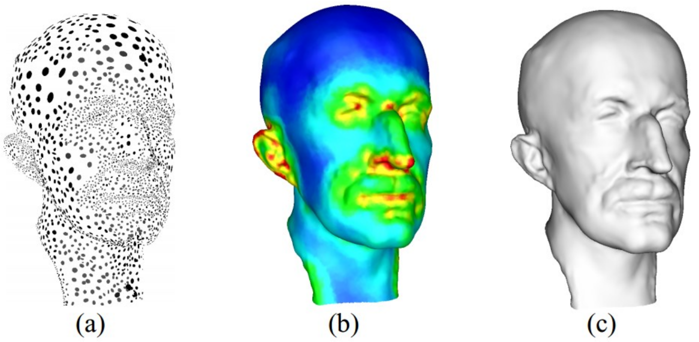


Figure 2-5. Adaptive MLS reconstruction of the Max Planck bust. (a) input point cloud with 5,413 points, (b) continuous sampling density map, (c) reconstructed MLS surface. (Pauly et al., 2003b)

The reconstruct of mesh models from unstructured point cloud is a classic topic in computer graphics field. Edelsbrunner and Mücke (1994) extend the alpha-shape algorithm from 2D to 3D to detect the shape of a point cloud with the parameter alpha to control the detail of level. A triangle is accepted as surface mesh if it can be touched by a ball with user defined radius. Similarly, Bernardini et al. (1999) use a Ball-Pivoting Algorithm to efficiently and robustly compute a manifold subset of an alpha-shape. Kolluri et al. (2004) use a variant of spectral graph to participate the tetrahedrons inside and outside the original object while removing outliers.

However these algorithms do not aim at averaging out noise. Pauly et al. (2003b) obtain the hybrid geometry representation by combining the unstructured point with the implicit surface definition of the moving least squares approximation (Figure 2-5). Usually the local facet is estimated by fitting plane using local points within a certain distance criteria. The estimation of an optimum neighbouring radius is a scale selection problem: the recovered surface should be smooth but detail enough. Guibas et al. (2004) find the optimal neighbourhood size based on the local information from the noisy point cloud. Pauly et al. (2003a) use the size of the local neighbourhoods as a discrete scale parameter to analysis line-type features at multiple scales. There is also a lot of research on optimizing the normal field by a global or semi-global penalty. The level of noise and outlier differ inheriting to sensor types, distance to scanner, acquisition condition and light-material interaction. Considering the variable level of noise and outlier, Giraudot et al. use a noise-adaptive distance function to infer the shape (Digne et al., 2013; Giraudot et al., 2013).

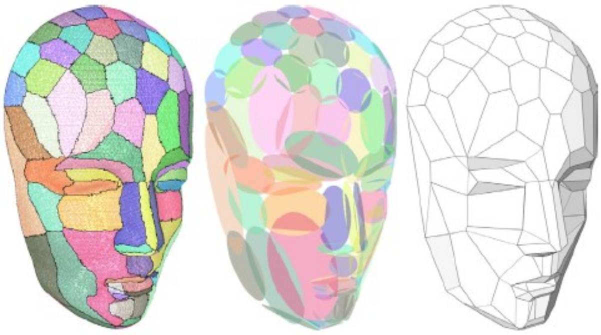


Figure 2-6. Repeatedly cluster faces into local smooth regions and construct approximating polygonal meshes. (Cohen-Steiner et al., 2004)

By the technology mentioned above, object models are created from point cloud. Nevertheless, the models are highly detailed and redundant, especially for objects with regular shapes. To control processing time and storage space, it is desirable to use approximation models instead of extremely detailed models. Garland and Heckbert (1997) simplify models by iteratively contracting vertex pairs and maintain surface error approximations. However, severe alias artefacts can be observed at sharp features on the extracted surfaces. Two main solutions are used to keep sharp features while simplifying models. One is to detect edges and corners directly on point cloud, sometimes with the estimated normal of local facets, and relocate the vertex to nearby sharp features. Kobbelt et al. (2001) utilize a marching cubes algorithm to extract sharp features when recovering surface from volume data. Machine learning based on statistic information is also used to classify sharp features (Cohen-Steiner et al., 2004; Weber et al., 2010). Another one is to partition the mesh into a set of smooth regions and thereby sharp features are detected as border edges connecting different regions (Attene et al., 2003). Fleishman et al. (2005) use a forward-search paradigm method to classify local smooth regions which guarantee sharp features and remove outliers. Ohtake et al. (2003) detect local piecewise quadratic surfaces and blend them with a weighting functions to get sharp features on edges and corners. Cohen-Steiner et al. (2004) repeatedly cluster faces into local smooth regions and construct approximating polygonal meshes (Figure 2-6).

The (decimated) mesh models are commonly used to represent the shapes of animals, statuses, industry products, and other relative small objects. With the development of equipment which can measure lager objects, the mesh models are used to model bigger objects, including a single building and even a whole city scene. Frueh et al. (2005) generate textured façade meshes of cities. Hähnel et al. (2003) create compact models for indoor and outdoor environment from mobile laser scanning data. Marton et al. (2009) fast create triangle models in real time for an indoor scene model from laser scanning data. Nie et al. (2013) model large scene from consumer depth camera data. Wahl et al. (2008) first detect semantic shapes and use them as constraints in the simplification to derive city models from digital surface models (DSM). Walls are rarely scanned by the airborne laser scanner. Therefore large holes exist between roof layers. A typical mesh simplification will over simplify small parts, like chimneys will be lost, and create zigzagging boundaries of eave boundaries. Zhou and Neumann use 2.5D dual constraints to recover vertical walls and create simplified roof models while reserving sharp features (Zhou and Neumann, 2010; Zhou and Neumann, 2011)(Figure 2-7). Lafarge and Alliez (2013) create and simplify mesh models from the structured points which are created from the noisy point cloud and regular distributed points on the detected planes and ridges. Lafarge (2013) provides a hybrid way to represent compact objects while preserving details: regular surface with primitives models and irregular surfaces with mesh models.

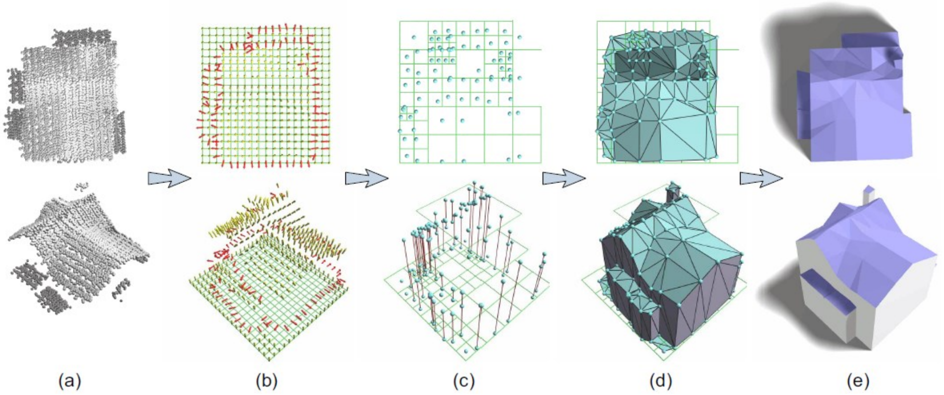


Figure 2-7. 2.5D dual constraints is used to recover vertical walls and create simplified roof models while reserving sharp features (Zhou and Neumann, 2010)

2.3 Roof model reconstruction

The reconstruction of architecture roof models has developed a lot in the last two decades. There are two main methods to automatically reconstruct buildings, first one is the reconstruction from 3D features, including line segments, corners, matched from image sequence, and second one is the reconstruction from 3D point clouds, including those acquired by laser scanner and image matching. In the early period of building reconstruction, only coarse model of building could be reconstructed because of the low resolution of data, including satellite images and airborne laser scanning data. Many systems provide user interface to interactively acquire basic model of architecture (Debevec et al., 1996; Sinha et al., 2008; Ziegler et al., 2003). They usually need users label corners, lines or planes, and provide primitive models types.

Thanks to the recent development in laser scanning technology and image matching, dense point clouds with high precision and density can be cheaply and quickly acquired and open a great door for 3D modelling. SIFT operator presented by David Lowe provides scale invariant feature which enable us to reliably match different views of an object or scene (Lowe, 2004). Snavely and colleagues provide bundler adjustment to automatically detect interior and exterior orientation for large unstructured collections of photographs (Snavely et al., 2006) and then 3D point clouds are reconstructed simultaneously (Agarwal et al., 2009). Hirschmuller (2008) presents a linear complex dense match algorithm, Semi-Global Matching, resulting in sub-pixel accuracy, which employs Mutual Information for pixel-wise matching and semi-global constraint for path-wise smoothness constraint. Nowadays computer vision techniques can rapidly create detail 3D point clouds based on bundle adjustment and dense matching from unconstructed photo collections (Agarwal et al., 2010).

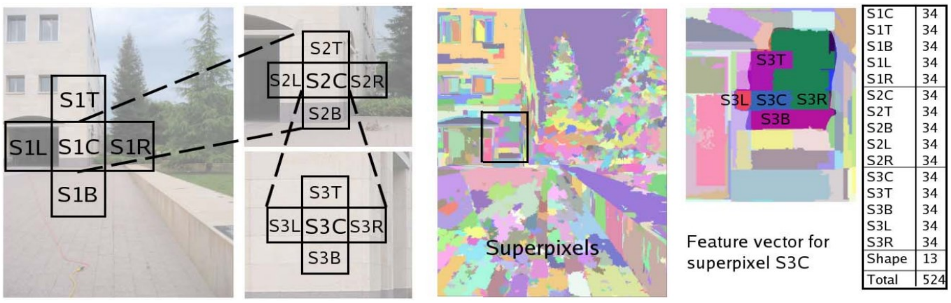
Roof models can be reconstructed from images, lidar data or the combination of them. The reconstruction from single information source is described in following sections. Images provide accurate edge and corners, while lidar data provide stable surface segments. Their combination offers more complete data for building reconstruction (Hu et al., 2004; Sohn and Dowman, 2003; Sohn et al., 2013). However, the two different data conflict sometimes and bring difficulties to use them simultaneously. The category maps of buildings are also used to locate buildings, give precise footprints, and analysis building interior structures (Brenner, 2000; Brenner and Haala, 1998; Haala et al., 2006; Taillandier, 2005; Vosselman and Dijkman, 2001).

2.3.1 Image based roof model reconstruction

Single image

3D structure reconstruction from a single image is an ill-conditioned problem and is very difficult to solve. The imaging process from 3D world to 2D image loses 3D information. Multi-view images are usually required to retrieval the lost information. Many monocular information and geometric structures also give hints about camera parameters and scene depth. The structure of scene components, if not repeat, usually needs human input to interpret. To keep the interaction simple, the user only manipulates in the image space to mark out various architecture components such as walls and roofs. When no 3D geo-information given, the interpretation of scene structures is an image understanding problem and still open for further research. The research on 3D reconstruction from single image can be divided into two groups:

1) ***Depth statistics method.*** This method link the depth information of image patches with monocular image cues, like brightness, texture, shapes, and maze and so on, by supervised learning over a large dataset of nature images. Then by a global optimization method, the 3D orientation and position of each planar patches of an input image are computed as well as the 3D scene that it presents. A serial of works on this method has been done by a team from University of Stanford (Saxena et al., 2008; Saxena et al., 2007a; Saxena et al., 2007b; Saxena et al., 2009)(Figure 2-8). Except assuming that the environment is made up of a number of small plane patches, this model makes no other explicit assumptions, therefore has wide adaptability. The models presented by TIN have accurate quality and pleasing visualization, however, the occluded parts are not considered in the reconstruction.



S1C	34
S1T	34
S1B	34
S1L	34
S1R	34
S2C	34
S2T	34
S2B	34
S2L	34
S2R	34
S3C	34
S3T	34
S3B	34
S3L	34
S3R	34
Shape	13
Total	524

Figure 2-8. The feature vector for a superpixel, which includes immediate and distant neighbors in multiple scales. (Saxena et al., 2009)

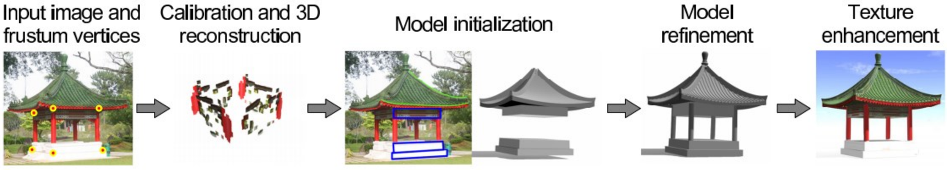

Input image and frustum vertices
Calibration and 3D reconstruction
Model initialization
Model refinement
Texture enhancement

Figure 2-9. Reconstruction of 3D shapes of symmetric piece-wise planar objects via a Markov random field to recover the depth map based on the symmetric line pairs. (Tianfan et al., 2011)

2) ***Geometric structure information.*** This method firstly uses the abound structure information in the image, including vanishing points, vanishing lines, parallel lines, parallel planes, perpendicular lines and perpendicular planes, to calibrate camera and obtain the rotation matrix, focal length, and scale, secondly incorporate the structure information to compute depth information, then finally use prior geometric constraints to reconstruct 3D object models. This method requests the scene and objects

have strong geometric regularities, and therefore cannot always be applicable. The scenes without repetition need human input to interpret. Li et al. (2001) reconstruct photorealistic scene models with freeform geometry from painting or photographs by assuming the scene as a non-uniform, piecewise continuous grid and optimizing the grid with a hierarchical transformation method with user given constraints. Sturm (2000) reconstructs 3D piecewise planar objects from single panoramic images based on user provide coplanarity, perpendicularity and parallelism constraints. Based on manual given object labels, Russell and Torralba (2009) recover implicit geometric information of relationships between objects and camera parameters and create high quality 3D information from a single image. Satoshi et al. (2012) model 3D scenes including a ground plane and walls as background and other objects as foreground based on a boundary between the ground plane and wall planes given by users. By manually given line drawing of objects, Zou et al. (2013) recover focal length and use a set of reference grid to find the precise 3D locations of corners and then recover object wireframe automatically. Just based on a few 2D control points interactively inputted, Zhou and Li (2011) model cone and cylinder objects.

As the manual editing is time consuming, many works automatically infer the scene structures and reconstruct 3D models. The automatic methods typically need more strong geometric assumptions about scenes and objects than semi-automatic methods. Guillou et al. (2000) use vanishing points for camera calibration and box shaped object reconstruction from a single image. Wilczkowiak et al. (2001) calibrate camera and reconstruct 3D scene from a single image using the strong geometry regularity of parallelepiped, such as such as parallelism and orthogonally. Lee et al. (2009) interpret and reconstruct the 3D indoor scene from a collection of line segments automatically extracted from a single image with a set of geometric constraints between groups of line segments (Figure 2-10). Zheng et al. (2011) obtain 3D building models by automatically extracting building contours from building photo and match the contours with the projection contour of the 3D models from different views predefined in the model database. Jung and Kim (2012) estimate the 3D structure of a scene by decompose the 3D world space into a set of geometrical primitive subspaces which are easy to solve and reliable even with the presence of occlusion or clutter, without loss of generality. Tianfan et al. (2011) reconstruct the 3D shapes of symmetric piece-wise planar objects via a Markov random field to recover the depth map based on the symmetric line pairs (Figure 2-9). Vanishing points, repetition, symmetry regularities (including mirror symmetry, rotation symmetry) are often used in 3D reconstruction from single image (Diego and Javier, 2008; Sinha et al., 2012; Wang et al., 2005).

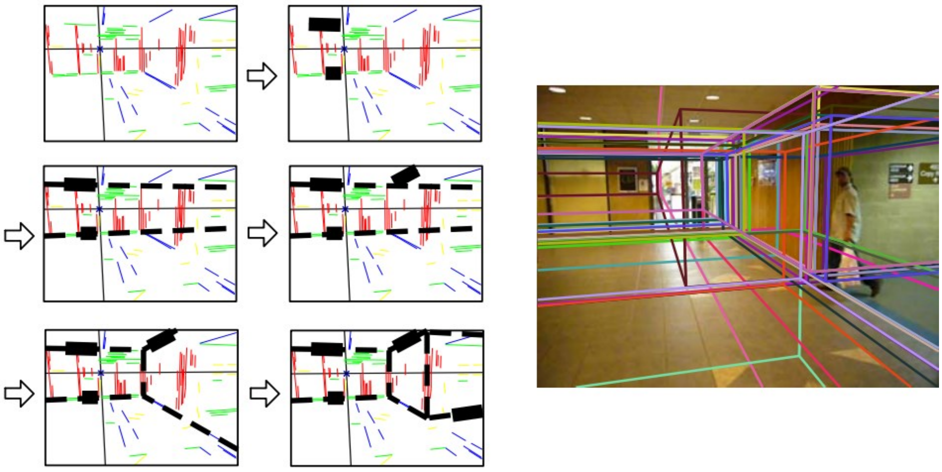


Figure 2-10.Generating hypotheses of grouping line segments with geometric constraints. (Lee et al., 2009)

Stereo images

Stereo images provide detail structure information about buildings including 3D points (Agarwal et al., 2009; Hirschmuller, 2008), 3D lines (Hirschmuller, 2008; Ok et al., 2012) and 3D planes (Micusik and Kosecka, 2009; Mičušík and Košecká, 2010). The extracted information could be used individually or combined to derive building models. Some algorithm for building reconstruction which is based on photogrammetric point clouds has very little difference with the one of lidar point cloud, so is discussed together with lidar point cloud in section 2.2, though the photogrammetric point cloud is less accurate.

3D scenes can be directly represented by triangle mesh models from dense matching point cloud (Esteban and Schmitt, 2003; Pollefeys et al., 2004). However the triangle mesh models are noisy and redundant. Users prefer to B-rep models. Fischer et al. (1997b) show a model-based approach to automate 3D extraction of buildings from aerial images (Figure 2-11). 3D corners, denoting a group of vertex, lines and planes, play key role in this method, in which 3D corners connect low-level features such as point, edges, with building part model. 3D corners are detected from matching 2D corners and then used to find building parts, which have same classifications with its sub-graph corners. Then building models are reconstructed by merging building parts if their plug faces are the same type. Steinhage et al. (2013) incorporate the DSM and footprint and use a similar strategy to reconstruct buildings.

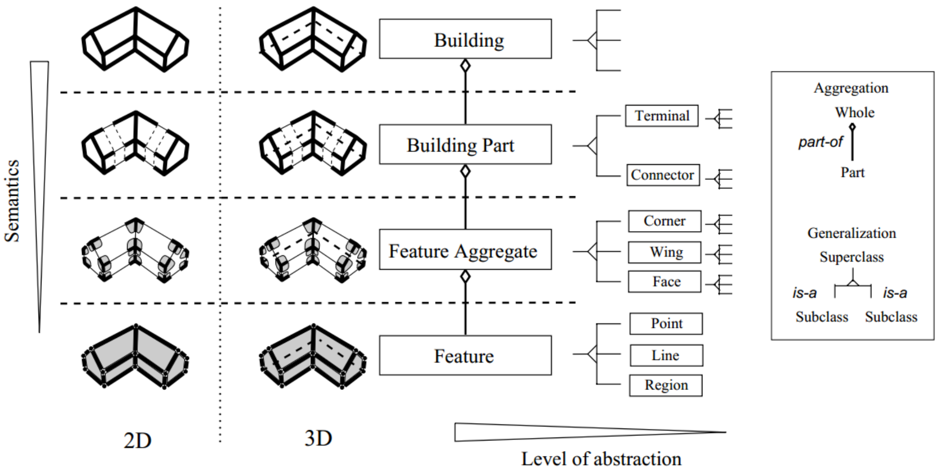


Figure 2-11. A generic hierarchical model for integrating of 2D and 3D reasoning. (Fischer et al., 1997b)

Taillandier and Deriche (2004) thoroughly search the possible plane arrangement based on the detected 3D planes and get the most possible building hypothesis that balance data adequacy and caricature needs. Taillandier (2005) generates plane candidates from each polygon segment of cadastral maps, create building hypotheses and choose the best representation which fit well to the digital elevation model computed from aerial images. Zebedin et al. (2008b) use graph cut to combine DSM and 3D line segments to generate building models. The 3D lines are used to partition the building area into regions, each assigned to a primitive (plane or surface of revolution).

Because automatic systems fail to fully understand the building structures, many semi-automatic systems have be developed. Debevec et al. (1996) give an easy-to-use photogrammetric modelling system to interactively recover the basic geometric model of architectures. They creates novel views by view dependent texture mapping, and automatically recover additional geometric detail through stereo correspondence. Gülch et al. (1999) provide a semi-automatic system to support operators to adapt parametric models and their arrangements to multiple overlapping images. Sinha et al. (2008) present an interactive system for generating 3D models. Users draw outlines on 2D images and 3D structures are automatically computed by combing multi-view geometric information. Arikan et al. (2013) use O-Snap to automatically snap the polygons derived from point segments based on optimization (Figure 2-12). The users are involved to modify the coarse polygons if the initial polygons are not good enough to get a correct surface adjacency topology graph, e.g. in the area with large hole.

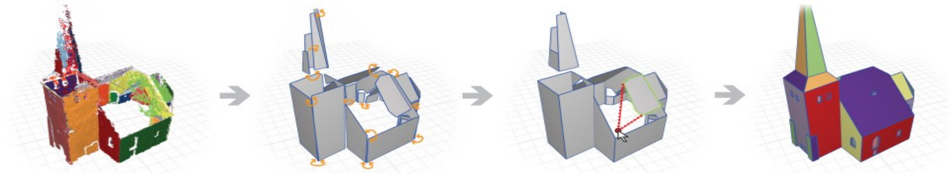


Figure 2-12. O-Snap for automatic optimization-based snapping of the polygons derived from point segments. (Arikan et al., 2013)

2.3.2 Point cloud based roof model reconstruction

Three types of approaches have been proposed to reconstruct LoD2 building models from point clouds and DSM: mesh simplification, fitting of building shape primitives, and segment based methods. Mesh simplification is initially used to speed up visualization, delivery and storage (Garland and Heckbert, 1997). Wahl et al. (2008) use it to rapidly generate 3D city models. In order to keep sharp features (like plane intersections) and topological relationships, topology constraints are introduced (Verdie et al., 2011; Wahl et al., 2008; Zhou and Neumann, 2011). The mesh models are geometrically close to the raw data, and when textured, are close to the reality. Because no assumption about building shapes is needed, the mesh model is able to represent diverse building structures. This representation, however, does not contain semantic information as one roof face would be represented by several triangles. Besides, the models sometimes include artefacts caused by outliers that are included in the triangulation.

Fitting of building shape primitives, which has few parameters and predefined topologies, is used to model buildings in low resolution data in the early researches. The point clouds from dense matching or lidar were noisy and sparse at that time. Therefore, the building roof structure could not be inferred from the original data by bottom-up methods. Additional data and building knowledge is necessary. Haala et al. (1998) use building primitives, like pent, flat, gable, and mansard roof, to represent building parts. Usually a 2D map with building footprints is used as extra information. However, these basic primitive shapes are not flexible enough to generate precise building models with complex shaped footprints. To solve this problem, Taillandier (2005) extrudes given footprints to a uniform building eave height. Building footprints are decomposed into cells. Each one is used as the footprint of a building primitive (Haala et al., 2006; Kada and McKinley, 2009b). 3D primitives are constructed for all cells, and combined in a CSG-like manner to form a complex building model. Lafarge et al. (2010) use a Markov Chain Monte Carlo sampler (MCMC) and simulated annealing to find the optimal configuration of building primitives (Figure 2-13). Huang et al. (2011) narrow down the possible moves in the jump routine by only allowing change of a limited number of parameters in each step under some rules. This strategy speeds up the search. But as it is a kind of blind search, the convergence of search is rather slow. Suveg and Vosselman (2004) combine cell decomposition and primitive searching for the optimization of a whole building model. Henn et al. (2013) propose to use RANSAC and a supervised classification method to search simple building primitives in sparse lidar data (1.2 points/m^2). Fitting building primitives can be slow because types and parameters of building primitives for each cell are determined by an exhaustive search. Furthermore, due to the complexity of building structures and inaccuracies in the available building footprints, it is hard to generate exact decompositions that fit well with roof shapes.

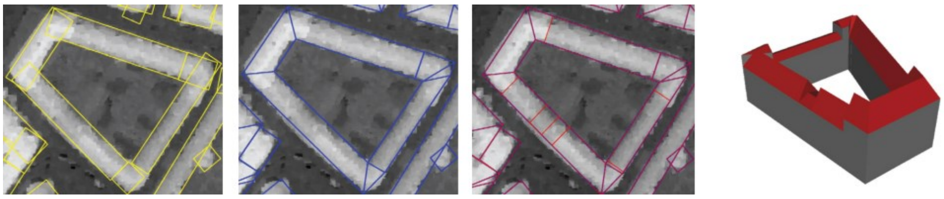
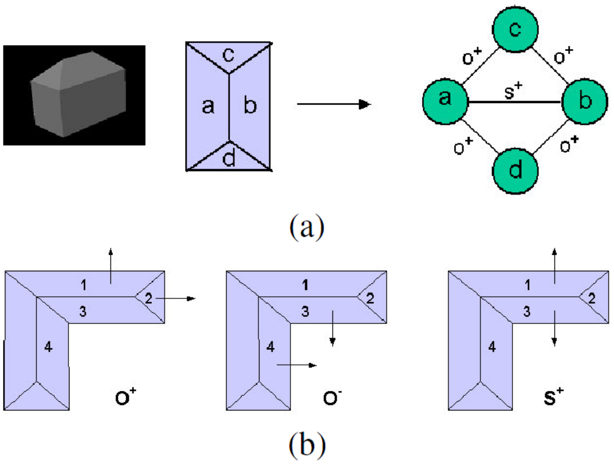


Figure 2-13. Searching for the optimal configuration of building primitives by Markov Chain Monte Carlo sampler (MCMC) and simulated annealing. (Lafarge et al., 2008)

With the further development of accuracy and density of point clouds obtained by laser scanning or dense image matching, roof faces can now be extracted more reliably from point clouds (Schnabel et al., 2007; Vosselman et al., 2004) and thereby can be taken as the basic unit for building modelling. By combining and intersecting roof faces of basic shape, a polyhedral model can be reconstructed (Brunn and Weidner, 1997; Lafarge and Mallet, 2011; Maas and Vosselman, 1999; Sampath and Shan, 2010; Sohn et al., 2008; Taillandier, 2005). The topological relations between roof faces are very useful in finding ridges, step edges, as well as roof structures. The roof topology graph is used to infer simple building primitives, like I, L, and U shaped primitives, which introduce geometric constraints for further improving models (Milde and Brenner, 2009; Oude Elberink and Vosselman, 2009; Verma et al., 2006) (Figure 2-14). The topology graph is a powerful representation of the inner structure of building roofs and is easy to combine with prior knowledge. It is a low-level feature and gives hints about the structure to speed up the search. However, a roof topology graph may have errors if the point cloud segmentation fails because of outliers or low point densities on poorly reflecting surfaces (lidar) or textureless surfaces (image matching). The erroneous roof topology graph will result in incomplete interpretation and therefore a wrong model (Oude Elberink and Vosselman, 2009; Sampath and Shan, 2010). The geometry of roof segments can be used to derive the roof topology graph. Brédif (2010) corrects the roof topology graph considering the 3D dual property. Lafarge and Alliez (2013) reconstruct polyhedron models from inexact adjacencies of surface planes.

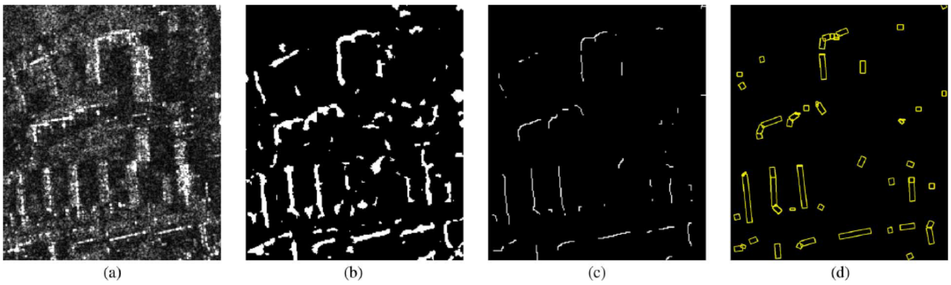


(a)

(b)

Figure 2-14. Inferring building types from roof topology graphs and building primitives. (Verma et al., 2006)

2.3.3 SAR



(a) (b) (c) (d)

Figure 2-15. Construction of building candidates and footprints from a set of low-level features from VHR SAR images. (Ferro et al., 2013)

The improved ground resolution of synthetic aperture radar (SAR) sensors provides possibilities of its applications in building reconstruction. SAR systems are active survey systems and are highly attractive due to their daytime and weather independency. Besides, SAR has the advantage of low cost and high acquisition speed. It is as well valuable for rapidly acquiring area-wide information of regions hit by disasters such as flooding, landslides, or earthquakes. However, building reconstruction from SAR or InSAR data suffers from consequences of the inherent oblique scene illumination, such as foreshortening, layover, occlusion by radar shadow and multipath signal propagation. These shortages bring large problems in building reconstruction and block its possible applications.

Experimental airborne synthetic aperture radar (SAR) systems reach spatial resolutions of approximately 10 cm, whereas the new space borne very high spatial resolution (VHR) SAR sensors onboard the TerraSAR-X and COSMO-

SkyMed satellites achieve spatial resolutions down to 1 m. Bolter (2000) calculates the position and height of all walls facing away from the sensor from the position and length of the building shadows in a single slant range image, and then reconstruct the building geometry by assuming the building has orthogonally intersecting walls. Brunner et al. (2009) estimate heights of man-made objects from one aspect or a combination of two aspects of single VHR SAR imagery. Ferro et al. (2013) extract a set of low-level features from VHR SAR images, on their combination to more structured primitives, and derive building candidates and footprints.

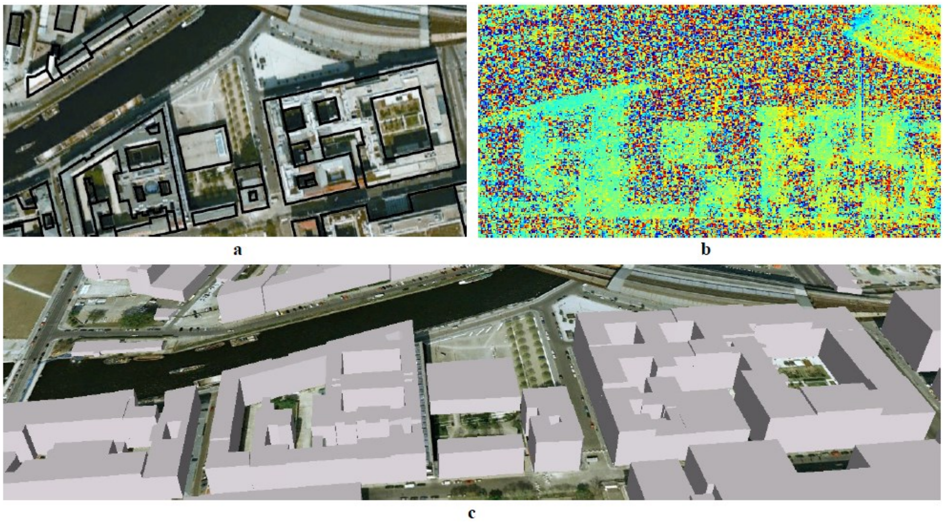


Figure 2-16. 3D building reconstruction by fusion of InSAR and GIS data. (Thiele et al., 2010)

The inherent oblique scene illumination makes it hardly possible to reconstruct buildings based on single SAR or InSAR data sets alone, especially in built-up areas. Thiele et al. (2007) extract line features from SAR images and group them into hypothesized buildings with rectangle footprints. The hypothesized buildings are then used to simulate SAR image and are compared with real image to filter false hypothesizes and obtain building heights. Brunner et al. (2010) estimate building height by hypothesizing the height of buildings with flat or gable roof structures until the matching of simulated SAR image with the actual SAR image is optimized. They achieve 2cm resolution for VHR airborne and 1m resolution TerraSAR-X SAR scenes. The GIS data which contains building footprint is reliable to provide building initiatives. Thiele et al. (2010) simulate building hypotheses based on the 2D footprint and estimate building heights (Figure 2-16). The hypothesized buildings are used to synthetic InSAR phases and the building models are improved in an interactive adjustment between simulated and measured InSar phases. The inherent oblique scene illumination gives hints about the building structures and heights. Based on the observation that gable-roofed buildings give salient pairs of parallel lines of bright magnitude, Thiele et al. (2008) locate gable-roofed buildings by considering SAR-specific effects and estimate the building height from geometric parameters from the amplitude image. TomoSAR uses multiple data stacks from different viewing angles and generates 4D point clouds from space with a point density comparable to lidar (Zhu and Bamler, 2010). Characterized by the side-looking SAR geometry, the TomoSAR point clouds have rich façade information and allow automatic reconstruction of building façades (Zhu and Shahzad, 2014).

All in all, the state-of-art of building reconstruction from SAR could only provide footprints, LoD1 building models and some other very simple buildings. The precision of SAR is not high enough to support fine building structure reconstruction. 2D footprint maps are needed to analysis the inter structure of buildings and building primitive libraries are appreciated for reconstructing buildings with the model-driven methods. The primitives used in building reconstruction from SAR are still quite simple, like flat and gamble. The input SAR or InSAR image could not directly provide accurate DSM, and need hypothesis building types and simulate a SAR image to match with the actual SAR image. It is an iteration problem and therefore high complexity buildings are very hard to be reconstructed.

2.4 Façade modelling

Since the L-system has been successfully applied to procedurally model façade and city scene (see the section 2.1 on procedural modelling), the façade grammar has been of high interest to model real façades. In general, architecture façades can be horizontally divided into floors and vertically divided into rows of windows. Façade elements, like windows, doors, ledges and balconies, have strongly repeating patterns. The regularities provide prior knowledge for detecting façade grammar and reconstructing façade elements. The learnt grammar is also used to procedurally model new façades. Façade grammar detection could be divided into three steps: coarse detection of façade elements, finding possible relationship between façade elements, and refinement of façade elements. Some algorithms combine the last two steps together in an iteration schedule. The coarse detection is highly dependent on the available features that data provides. A large set of algorithms used for image and point cloud classification can serve as tools for the coarse detection. The input data including images and point cloud is usually not detailed enough to reconstruct fine models of façade elements. A library of these models is manually created beforehand, linked with the detected façade elements, and used to represent them in the final façade models. Façade reconstruction can also be divided into two sub-classes: image based detection and point cloud based detection.

2.5.1. Image based modelling

In order to detect façade elements, images are usually segmented into regular grid (Koutsourakis et al., 2009; Tylecek and Sara, 2011; Weissenberg et al., 2013), irregular lattices (Dai et al., 2012; Riemenschneider et al., 2012) and sometimes just irregular patches (Martinovi et al., 2012; Peng et al., 2012; Tyleček and Šára, 2013). The façade elements can also be directly searched on the image via supervised classification (Wenzel and Förstner, 2008) and unsupervised classifications (Fröhlich et al., 2013; Xiao et al., 2008; Xiao et al., 2009). Ripperda and Brenner (2006) use a process based reversible jump Markov Chain Monte Carlo (rjMCMC) to guide the construction of model tree. Dick et al. (2004) utilizes a Bayesian and model based approach to automatically reconstruct 3D architecture models from short image sequences. Debevec et al. (1996) present an interactive system to extract building grammars which facilitate the visualization and quick modification of architectural structures. Müller et al. (2007b) present an algorithm to automatically reconstruct 3D models with high visual quality from single façade images (Figure 2-17). Teboul et al. (2010) combine shape grammars, supervised classification and random walks for the perceptual interpretation of building façades. However, the repeat elements usually have variations (e.g., windows in different open or close positions) which pose challenges to the façade analysis algorithms. AlHalawani et al. (2013) propose a semi-automatic framework to recover both repetition patterns of the elements and their deformations.

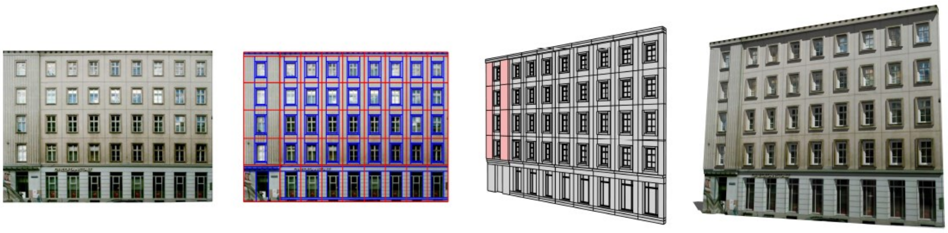


Figure 2-17. Automatic reconstruction of 3D models with high visual quality from single façade images. (Müller et al., 2007b)

The methods mentioned above are all based on rectified orthoimages. The images are rectified based on the assumption that the main structural of walls are planar. Therefore if the walls have curved structure or have large convex/concave structures, the images cannot be well rectified. The façade reconstruction directly in 3D space is more reasonable. On the other hand, the overall structure of façades should be reconstructed before the modelling of detail façade elements. Werner and Zisserman (2002) first reconstruct the coarse planar model of the main scene, and then use these facets to guide the reconstruction of façade elements. Schindler and Bauer (2003) reconstruct the coarse polyhedral model from the photogrammetry points and then refine the geometry of coarse model with predefined shape templates. Dick et al. (2004) utilizes a Bayesian and model based approach to automatic reconstruct 3D model of architectural from short image sequences. Based on 3D points and 3D edges from video images, Tian et al. (2010) detect planar surfaces and group them to building model by integrating building structure knowledge. Ceylan et al. (2012) use 3D lines to simultaneously detect symmetric line arrangements and refine the estimated 3D models (Figure 2-18).

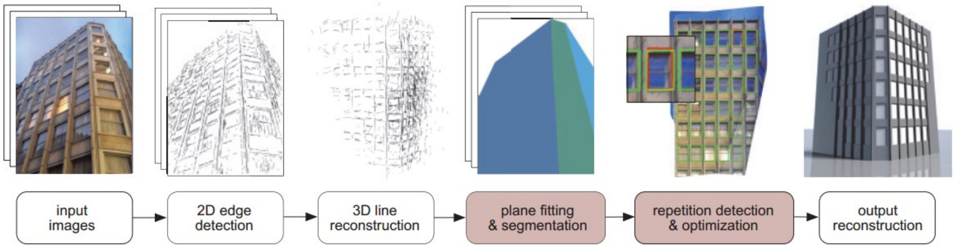


Figure 2-18. Simultaneously detect symmetric line arrangements and refine the estimated 3D models. (Ceylan et al., 2012)

2.5.2. Point cloud based

Zheng et al. (2010) use the repeat façade elements to consolidate the imperfect data. The consolidation provides robust de-noising and allows reliable completion of missing parts. Automatic façade analysis is typically restricted to globally rectilinear grids. Shen et al. (2011) adaptively determine the splitting direction, the number and location of splitting planes. Becker and Haala (2007) present a grammar based façade reconstruction to automatically generate façade structures in regions where no or only limited lidar points are detected (Figure 2-19). Pauly et al. (2008) propose a computational framework to discover regular or repeat geometric structures in point or mesh based models. They introduce an optimization method for detecting such uniform grids specifically designed to deal with outliers and missing elements. Friedman and Stamos (2011) do real time detection of repeat elements from each scan line. By nearest-neighbouring clustering on feature space, Kerber et al. (2013) divide the façade elements into different types and detect partial symmetries. This method can handle very large scenes with many different façade grammars. As the point cloud is not detail enough to construct fine models, Nan et al. (2010) use a smartbox to interactively search and model façades.

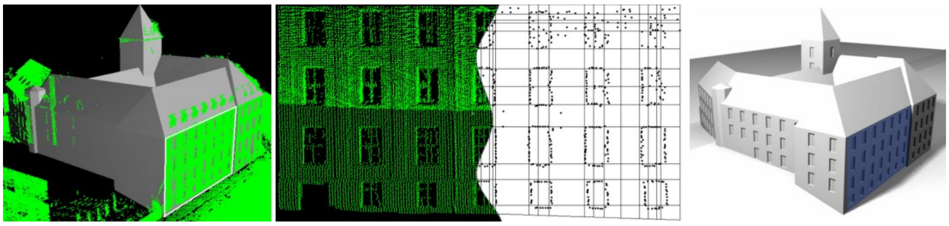


Figure 2-19. Grammar based façade reconstruction in regions where no or only limited lidar points are detected. (Becker and Haala, 2007)

Most research focuses on the façades with strong regularities, however most resident buildings only have a few doors and windows. Pu and Vosselman (2009) present a knowledge based method for automatic reconstruction of façade models of resident buildings from terrestrial laser scanning data (Figure 2-20). Demantke et al. (2013) provide a regularity-prone method to reconstruct façade with a deformable 2.5D grid. The method can be applied to a large amount of data regardless of the diverse building types.

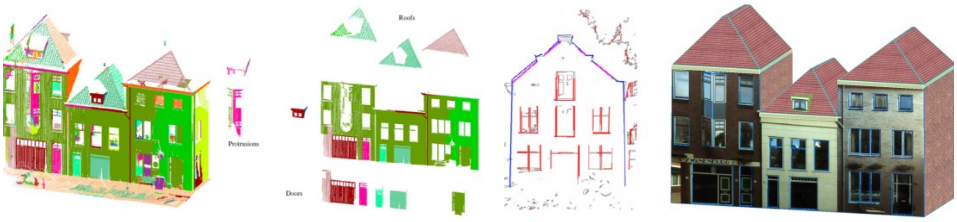


Figure 2-20. Automatic reconstruction of façade models of resident buildings from terrestrial laser scanning data. (Pu and Vosselman, 2009)

2.5 Indoor Modelling

The indoor environments are essential for the life and work of human beings. Without the indoor models, the digital 3D cities are not complete. The indoor mapping for big public facilities are urgently needed now, like airports, train stations and museums. The location based serves can navigate people to accurate address but cannot help us to find the exact rooms. In order to monitor and maintain the facilities, factories and historic buildings also need to be reconstructed by reverse-engineering. 3D models of building interior scenes also play an important role in fields such as autonomous mobile robotics, unmanned aerial vehicle (UAV), smart home systems, and computer games. Compared to the 3D mapping of outdoor, indoor mapping shows a different set of technical challenges and requirements (Huber et al., 2011; Tang et al., 2010). Lack of GPS signal, crowded environments and unreflecting surfaces make the acquired data inaccurate and incomplete. The imperfection of the data acquisition explicitly affects the results of interpretation and modelling. are The data used to interpret and reconstruct the indoor environment include lidar data, images, Kinect data and their combinations (El-Hakim et al., 2005; Izadi et al., 2011; Liu et al., 2010; Mozos et al., 2012; Rusu et al., 2008; Smisek et al., 2011).

The indoor environment is quite complex: walls are interspersed with windows and doorways, parts may be occluded by furniture, pictures, and clocks and so on. The voxel representation and occupancy map is heavily used for decreasing data volume, fusing measurement from different scanners, detecting occlusion and gaps (Biber et al., 2004; Gutmann et al., 2005; Turner and Zakhor, 2013; Wurm et al., 2010). No matter it is for modelling or interpretation, the semantic information of objects is essential. The main parts, including walls, floors, and ceilings are usually firstly identified and modelled (Valero et al., 2012b). Based on them, doors, windows, furniture and other objects are detected, spatially and semantically related, and modelled (Adan and Huber, 2011; Huber et al., 2011; Koppula et al., 2011; Valero et al., 2012a; Xiong et al., 2013).

To allow for an accurate reconstruction such that the space decomposition is aligned with permanent structures, Oesau et al. (2013) use Hough Transform to detect wall segments and use graph cut to predict inside/outside cells of the space decomposition with the data consistency. Turner and Zakhor (2012) separate lidar points for each floor, fit curves and line segments to walls and develop a watertight representation of the walls for each scanned area. Budroni and Boehm (2010) segment the point cloud into planar patches and create the ground plan through cell decomposition by trimming the 2D ground space using half-space primitives. The building models are created by extending the ground contours along the height direction. Mesh models including triangle and grid are flexible to represent complex indoor scenes so are often used. Rusu et al. (2009) reconstruct hybrid semantic 3D indoor household environments by triangular meshes (Figure 2-21). Turner and Zakhor (2013) use voxel presentation to reduce the data amount and to infer interior or exterior, and generate triangulated surfaces which are watertight and sharp feature preserving by analysing planar surfaces.

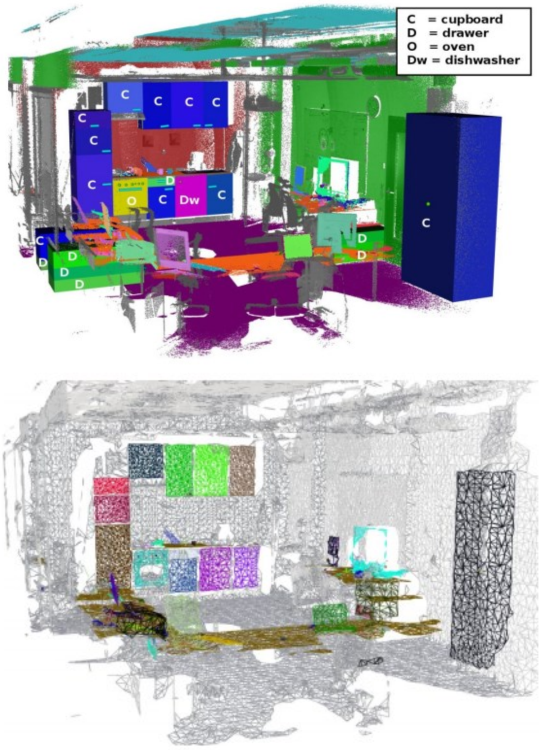


Figure 2-21. Reconstruction of hybrid semantic 3D indoor household environments by triangular meshes. (Rusu et al., 2009)

The indoor scene presents a strong pattern: a building is composed by floors, a floor is grouped by rooms and corridors, and floors are connected by stairs. This pattern is used to design, to procedural model, as well as to reconstruct the indoor environment. Marton et al. (2009) use triangle meshes to model indoor scene using data resampling and a robust triangulation algorithm in near real-time. Okorn et al. (2010) detect floors and celling via a histogram of height data and infer walls by detecting line features on the 2D density map of the remaining data. Ochmann et al. (2014) segment points for each room and locate doors between adjacent rooms. Turner and Zakhor (2014) triangulate a 2D sampling of wall positions, separate theses triangles into interior and exterior sets, and participate the interior volume into rooms. Oßwald et al. (2011) reconstruct stairs for robust climbing by segmenting the 3D point cloud into planar patches. The stair models are reconstructed by simply intersecting the vertical and horizontal planes for intersection lines, determining their end points and connecting end points of parallel lines.

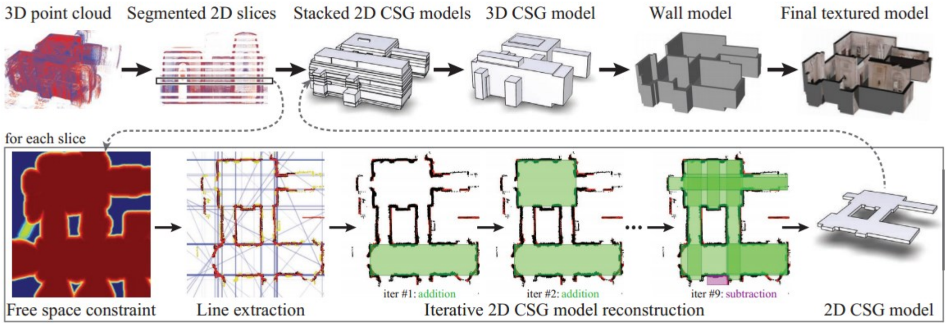


Figure 2-22. Reconstruction of textured museum models from ground-lvel photographs and 3D laser points with cuboid assumptions. (Xiao and Furukawa, 2012a)

Considering scene complexities and data imperfection, geometry assumptions are helpful for the reconstruction. Manhattan world assumption is a simple but effective one. A Manhattan world scene is a term used to describe a real world scene based on the Cartesian coordinate system. The scene is dominated by lines which are parallel with one of the X, Y or Z axes. The Manhattan World assumption was first discussed by Coughlan and Yuille to determine the orientation of the viewer given the scene structure and also to detect target objects that are not associated with the principle direction of the scene (Coughlan and Yuille, 1999; Coughlan and Yuille, 2000). Furukawa et al. use the Manhattan world assumption to improve the multi-view stereo algorithm and produce simplified 3D models (Furukawa et al., 2009a; Furukawa et al., 2009b). Using the Manhattan world assumption Neverova et al. (2013) estimate geometry of a whole environment from a single Kinect RGB-D image. Following the same direction, Xiao and Furukawa (2012a) assume that the indoor scene can be represented by Constructive Solid Geometry (CSG) representation consisting of cuboids and reconstruct textured museum models from ground-level photographs and 3D laser points (Figure 2-22).

The scene configuration, including the relationship between walls, furniture and other objects, are also useful to interpret and model scenes. Lee et al. (2010) volumetrically reason about objects and surfaces and estimate the spatial layout of rooms from single image. Jia et al. (2013) interpret 3D scene from RGB-D images by reasoning the volumetric relationship between object blocks. Yun et al. (2013) notice that the indoor objects serve for human beings and therefore have strong semantic and spatial relationships between them. These relationships are derived and applied to label 3D indoor scenes. Zhao and Zhu (2013) find that the object geometry is also designed to serve its functions and formulate the nature of object functions into a stochastic grammar model to interpret the indoor scenes. Nan et al. (2012) interpret cluttered indoor scene by a search-classify approach which iteratively interleaves segmentation and classification. Kim et al. (2012) use the repetition of objects (tables, chairs, monitors) to model 3D indoor environments (Figure 2-23). The 3D models of frequently occurring objects are learned firstly from a few Kinect RGBD data and used to identify and model objects from a single scan of a new area.

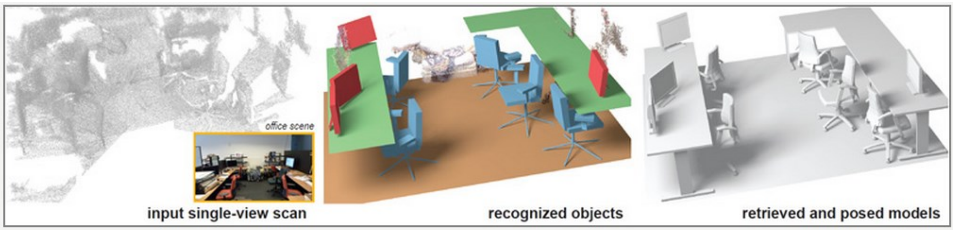


Figure 2-23. Modelling 3D indoor environments with repeat furniture.(Kim et al., 2012)

2.6 Conclusions

This chapter reviews the literature on building model reconstructions. The literature can be divided according to input data, goal objects and geometry assumptions. In order to get a clear organization, the literature review groups the prior works according to goal objects on the first level and according to the input data on the second level. The work on building reconstruction has tried different input data, different goal objects and different methodologies. The data and goal objects are such diverse that no overall comparison of the methodologies could be concluded. The specific discussions for each subgroup have been made in each section. As this thesis focuses on building modelling from airborne lidar data, the author would like to emphasize the problems remaining in literature on this topic: 1) The primitive model libraries given in literature are all not complete, therefore the model-driven methods can only reconstruct buildings with simple shapes and some more complex buildings; 2) Roof topology graphs stand in the middle level between point clouds and building models. They play the linking role in modelling. However, the derived roof topology graphs are always erroneous. There is no work in literature to correct the errors in roof topology graphs.

An extensive literature has accumulated on the structure knowledge which can be used to guide building reconstruction. In the next chapter, the author will organize the structure regularities proposed in the literature and present several novel regularities proposed by him. The author would like to summarize general rules and trends. Generally speaking, there is no way adaptive to all the data and objects. With the improvement of data quality, the requirement of the quality and detail of models are as well increasing. It is still an open question to automatically model diverse objects with high detail and high accuracy.

The 3D point cloud and 3D line segments with a good distribution, high density and accuracy directly provide 3D geometric information. They therefore provide the best possible input to analyse geometric structures and reconstruct 3D building models. The data acquisition ways are improving and the data quality increase day after day. Therefore we can see that in literature there is an increasing number of reconstruction algorithms which rely more on data and less on assumptions. However, the data are still noisy and incomplete, and will remain so. The model reconstruction algorithm should be robust enough to the data. On the other hand, structures and regularities of objects do not change with the data. Finding and employing the geometric regularities with wide adaptability are the indispensable means to reconstruct high quality models.

The geometry assumptions sometimes just improve the model quality but not guarantee a complete model. More simple and general regularities of buildings need to be extracted. If we can infer regularities directly from data, it is unnecessary to heavily and randomly pick primitives from the primitive library to meet data. Most regularity based reconstruction methods request strong constraints to which reconstruction result needs to conform strictly. Those constraints, sometimes, are not followed by reality. Such situations require soft constraints, which could be switching on/off, thus are able to deviate from their ideal shapes. One way to introduce constraints for refining detection result is model-based methods, which can enforce model reconstruction according to defined shapes. However, as many researchers mentioned, model-driven techniques are not able to reconstruct arbitrary complex buildings. The option to expand the model library is necessary if some complex building shapes are not stored in it.

Chapter 3

Regularities of Building Structures

As man-made objects, buildings have strong shape and structure regularities. The regularities are often used to simulate and reconstruct the buildings models. These rules are applied more or less, explicitly or implicitly in the literature. This chapter summarizes the regularities and suggests the suitable situations where regularities could be applied and the factors that need to be considered. The regularities have four types: hierarchical regularities, line and face regularities, template regularities, and simple region regularities, see Figure 3-1.

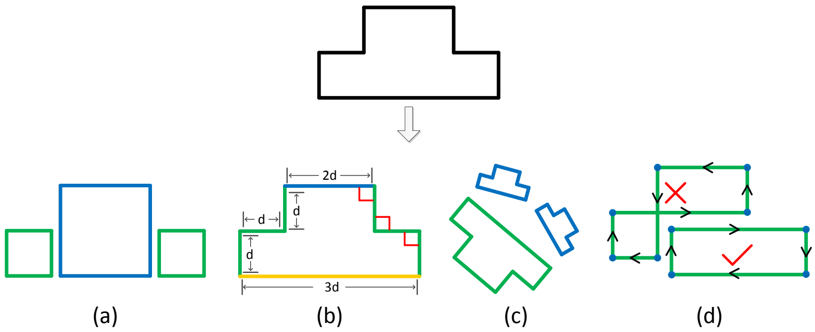


Figure 3-1. Four types of regularities. (a) Hierarchical regularities; (b) Line and face regularities; (c) Template regularities; (d) Simple region regularities.

These regularities are presenting the same structures but from different levels and perspectives. The hierarchical regularities present the relationships between objects and their parts, e.g. the black polygon could be hierarchically composed by three squares, see Figure 3-1 (a). Line and face regularities directly describe the orientation and alignment patterns of the geometry elements, e.g. the black polygon has six equal length edges, and eight right angles, see Figure 3-1 (b). Templates integrate regularities of a group of lines and faces. Illustrated in Figure 3-1 (c), the three polygons have same shape, but different positions, size, and orientation. The templates are at a higher level than lines and faces. Therefore the relationship between faces and templates can be seen as hierarchically related. The simple region regularities are to ensure the valid and watertight models. E.g., see Figure 3-1 (d), although the two polygons have similar line and face regularities with the case in Figure 3-1 (b), and can also be hierarchically composed to the black polygon as in Figure 3-1 (a), but this polygon is not simple. The simple region regularities are the most basic requirement that a model needs to meet. The definitions of the regularities will be given in the following sections.

This chapter is based on the literature review in Chapter 2 and is the foundation of the following chapters. In addition to summarize and analyses the regularities used in the literature, the author proposes several novel regularities. The regularities discussed in this chapter will be partly used in the building reconstruction methods shown in this thesis. The representation methods heavily influence the strategy selection for building reconstruction. Therefore their relationships are as well interesting to analyse. Three ways are proposed to represent surface structures in the literature: constructive solid geometry (CSG) model, binary space partitioning (BSP) tree, and topology graphs. This chapter first discusses the three types of regularities and then compares the three representation methods. By analysing the graph structure of polyhedrons, this chapter also proposes the definition of the minimum sub-graph basis and uses it to prove the definition of simple polyhedrons (cf. section 3.4).

3.1 Hierarchical Regularities

Hierarchical regularities are defined to describe the decomposing and grouping relationship of objects. All objects have components and also serve as components for the higher level objects. A building is usually composed by façades and roofs when only considering the outside. Buildings also group to a block, an area and a city. The hierarchical regularities contain the composition relationship between father and children and the alignment relationship of objects of the same level. A typical building has the hierarchical relationship shown in Figure 3-2. The interior and exterior are usually modelled independently because most application usually only consider one of them, either interior or exterior part.

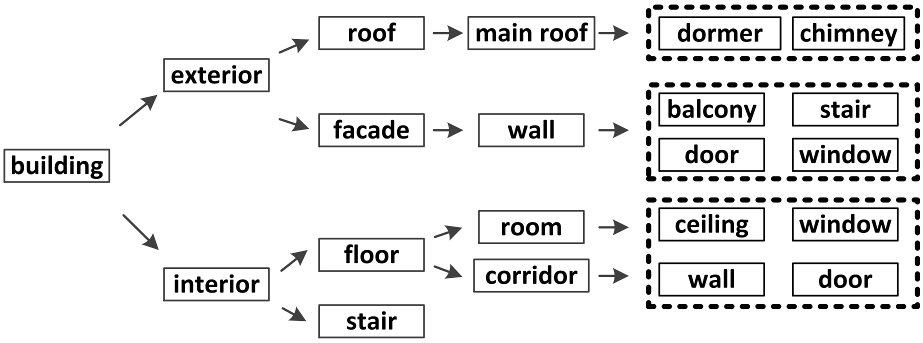


Figure 3-2. Hierarchical structures of buildings.

L-systems are an efficient and concise way to describe the hierarchical regularities. In 1968 Lindenmayer introduced L-systems or Lindenmayer systems to describe the behaviour of plant cells and to model the growth processes of plant development (Lindenmayer, 1968; Prusinkiewicz et al., 1990). An L-system is composed of a collection of symbols that are used to make strings, a set of production rules that expand each symbol into larger strings of symbols, an initial axiom string where the construction starts, and a mechanism for explaining the produced strings into geometric structures.

L-systems are widely applied in computer simulation (Parish and Müller, 2001; Prusinkiewicz et al., 1990) and model reconstruction (Müller et al., 2007a). When given all the parameters of an L-system and its translation mechanism, a photo-realistic object can be modelled. However, it is still an open question, given a structure, to find an L-system that can generate that structure. The difficulty of this problem is threefold. Firstly, it is very difficult to accurately reconstruct the objects from the imperfect data. Data inevitably has noise and gaps because of sensor accuracy, occlusion, sample resolution, and surface properties causing light absorption and mirror reflection. Secondly, the hierarchical level that certain information belongs to is ambiguous. The scale or the size of objects is not the only factor to check the hierarchical level. For example, the convex and concave parts of walls, although smaller than windows, are belonging to walls, but not windows. Therefore in the automatic interpretation, it is very hard to determine the hierarchical level of a certain object. Thirdly, different types of objects have different L-systems. The selection of a proper L-system is an object interpretation problem, which is also an open problem now. The buildings can be divided into several large groups: office buildings, residential buildings, historic buildings, and so on. Every group can also be hierarchically divided into combinations of smaller groups. All groups need a specific L-system to model.

On the other hand, the advantages of hierarchical regularities and L-systems are obvious. They provide a way to integrate information from multiple levels. According to the data and interested level of detail, a certain level of hierarchy regularities should be chosen for guide building reconstruction. For example, the alignment regularities between buildings are rarely applied in building reconstruction. An object is supported by the information of lower level and verified by information of higher level.

The hierarchical regularities can be represented by an attribute graph grammar. Figure 3-3 shows a hierarchical model for a toy building. From top to bottom, the three layers are, respectively, the compositional objects, the part templates and the generalized geometric primitives. The elements in one level are compositions of elements in the lower level. Several essential factors should be fully considered in the hierarchical model. Firstly, objects are decomposed into their constituent parts and then modelled by visual vocabularies. Secondly, components at each level are composed to form large structures, and two different components may share common parts at the lower level. Thirdly, the compositions are specified through a set of attribute grammar rules. Each rule is associated with a number of hard or soft constraints on the attributes of the components, so as to model the pictorial relations. For example in Figure 3-3, a building façade is formed by grouping a door, two windows, and a wall by an "enchasing" relationship. Finally, each compositional building is represented by a hierarchical graph which is a graphical representation for the attribute grammar.

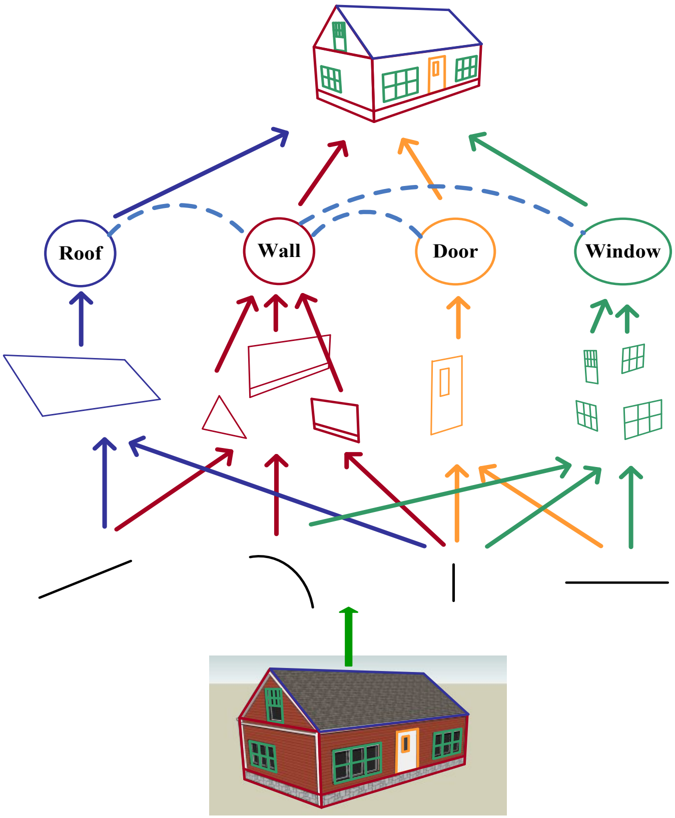


Figure 3-3. Hierarchical structures of the exterior of a building.

Learning and reconstruction of a hierarchical model can be combined in an on-line learning algorithm, in which the learned model is employed as prior knowledge to instruct building reconstruction. In return, the reconstruction result can be used as training data to improve building models. As the hierarchical graph is an explicit model, it can be constructed and revised feasibly by human. Primitives, such as door, window, would be labelled by operators for component model training. Their models are presented as child nodes in the whole graph. Constraint rules between primitives are learned from training data and part comes from expert architects.

In order to reconstruct a single building, basic components or primitives should be detected by bottom-top detectors and then grouped into parsing graph with context constraint. The context constraints are verified and are used to prune or predict sub-components by top-bottom Bayesian Inference. E.g. a wall with rows of windows is detected and the relationship of windows is used to improve the detection of windows. In order to detect a single component (e.g. door and window), lines, surfaces and junctions should be firstly detected as basic elements and then organized into planes and objects within some fundamental rulers. An object entity is described by its line and surface parsing graph and statistic features of texture and colour. By comparing with learned object models, new detected objects could be identified and then refined or excluded and scattered into elements for grouping another object.

3.2 Line and face regularities

Lines and faces are the most basic elements in geometry shapes. All solids are composed by lines and faces. Here lines include straight lines and curved lines and faces include planar faces and curved faces. The line and face regularities are used to call the geometry regularities of single line and face or a group of lines and faces. The regular structures of buildings are designed for function, beauty and construction simplicity. The walls are constructed to support the roofs, therefore are usually vertical to hold the weight of itself and the supported roofs. Many buildings are designed to have symmetry and repeat structures, making the buildings a strong impression of beauty. Besides, the regularities would save construction time and materials. Because lines are formed by intersecting faces, some line regularities are included in the face regularities.

The face regularities have been widely used to improve the model qualities in reconstructing models. Except buildings, other objects with the regularities are also reconstructed with the help of face regularities (Li et al., 2011; Werghi et al., 1999; Zhou and Neumann, 2012). Table 3-1 shows the frequently used geometry regularities. Although the categories and names are different, the constraints used in the three papers have a large overlap. The regularities of parallel and orthogonal relations, equal angle relations, coaxial relations, coplanar relations, equal radii, and equal distances are the most frequently used constraints. The regularities used by Zhou and Neumann (2012) are specially for roofs, so are a bit different with the other two sets. The single face regularities, like horizontal and vertical, are only valid for buildings. Ridge-height-equality and ridge-position-equality are also specially designed for buildings. Normal objects can be freely rotated, therefore the absolute geometries are usually not considered.

Table 3-1. **Line and face regularities.**

(Li et al., 2011)	***Orientation alignment:*** Parallel and orthogonal relations Equal angle relations	***Placement alignment:*** Coaxial relations Coplanar relations	***Equality alignment:*** Equal radii Equal distances
(Zhou and Neumann, 2012)	***Orientation regularities:*** Horizontal Vertical Orthogonal Parallel Opposite	***Placement regularities:*** Ridge-height-equality Ridge-position-equality	***Roof Boundary Regularities:*** Parallelism Orthogonality
(Werghi et al., 1999)	***Relative orientation:*** Parallelism Orthogonality Equal angle	Coincidence Inclusion	Relative separation

The objects we want to reconstruct can usually be represented by watertight polyhedrons. Edges are formed by intersecting faces. Hence edges do not need to be reconstructed independently, and no edge regularities are needed. But when reconstructing building roofs, walls are not included. Eaves need special care in processing. Therefore the edge regularities are applied. And 2D models only have edges, the edge regularities are the only possible constraints. The edge regularities are frequently used to regularize the building footprints (Keqi et al., 2006; Sohn et al., 2007; Weidner and Förstner, 1995).

Two important issues need to be considered when applying the line and face regularities. The first one is the way to search regularities. Brenner (2005) introduces the idea of soft constraints. The regularities are forced only when they are supported by the data. The conflicts between regularities should be found in the searching. For example, in the case of a polygon with five angles not all angles can be right angles. Werghi et al. (1999) ask operators to manually choose regularities. Li et al. (2011) and Zhou and Neumann (2012) automatically derive the possible regularities from data. The second one is the enforcement of regularities. The problem is a constrained least-squares fitting problem. Li et al. (2011) and Zhou and Neumann (2012) split the constraints into orientation regularities and alignment regularities, and independently and iteratively enforce the two groups of constraints. However, the two groups of the constraints are not linearly independent. In this way they cannot achieve a global optimization result but will bring conflicts in the optimization. When the concurrency regularities are used in the optimization, the regularities must be enforced simultaneously. The problem can be solved by the method of Lagrange multipliers, in which the geometric regularities are used as constraints. Brenner (2005) points out that the constraint optimization can be solved in two ways. One is to treat the Lagrange multipliers as unknowns and solve them simultaneously with other unknowns. However, the constraint functions are not linear. The objective functions cannot be directly solved. Another one is to give the Lagrange multipliers very large values, however running the risk that the objective function is ill-conditioned. Werghi et al. (1999) introduce an iterative processing to solve the constrained least-squares fitting algorithm. The optimization algorithm iteratively increases the value of the Lagrange multiplayers and solves the objective functions. This algorithm can guarantee the objective function well-conditioned and the constraints enforced.

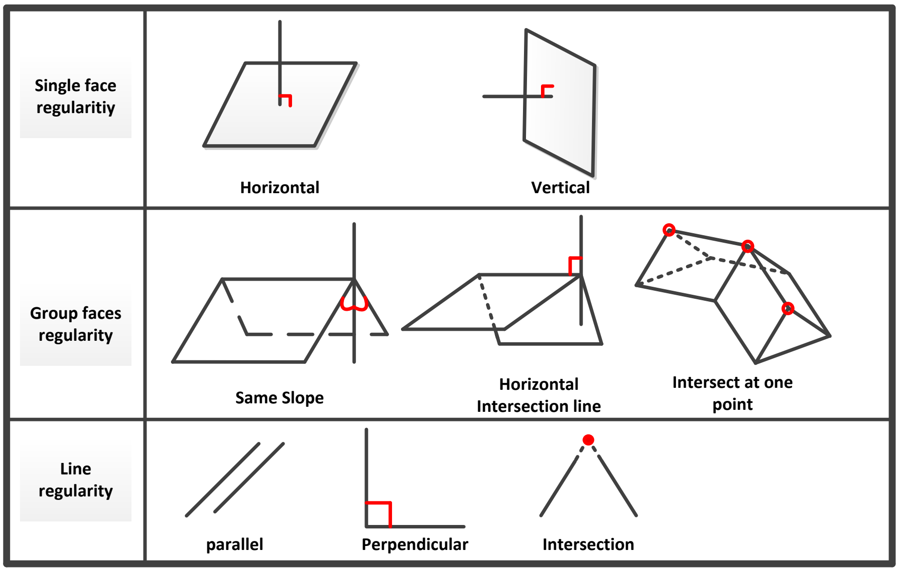


Figure 3-4. Geometry regularities.

Figure 3-4 shows the regularities used in this thesis. The regularities are carefully chosen to satisfy most of the cases. The constraints include roof plane and boundary constraints. The roof plane constraints enforce symmetry and horizontal and perpendicular intersections. The boundary constraints ensure identical heights, perpendicularity and parallelism of outlines. Apart from the often used constraints, the concurrent, a very important constraint, is also used in this thesis. The term concurrent is to express the situation that several planes intersect at one point. This constraint is necessary to force a watertight polyhedron. If only enforcing other constraints, roof faces will not necessarily intersect at one point.

3.3 Template Regularities

Geometry templates provide uniform shapes for objects within a same type. The objects are modelled by matching a predefined geometry template to the data. This method is powerful when the objects repeat and when the data is not accurate or dense enough to derive the fine models. For example, when reconstructing the façade models, the lidar points and images only have the resolution to present the window frame model. Glass, finer structures and decorations cannot be modelled. When reconstructing the railway furniture from terrestrial lidar data, including the signal lights and power towers, the data does not completely cover the object surfaces due to the occlusion. However, most of the furniture can be divided into several groups, and each group has exactly the same shape. A simple identification of the objects will enable their complete modelling.

The basic geometry shapes, like plane, cone and sphere, can also be seen as the templates. Their detection, like RANSAC algorithm, is to fit the predefined shape primitives to the data. However, the templates discussed here usually constitute a set of basic geometry shapes, which are composed with a certain orientation and position relationship. They are at a higher level than basic geometry shapes. For example, a simple window constitutes several cuboid frames and rectangular glass.

A set of geometric templates should be defined before their application. The models can be drawn by typical CAD software or by a semi-automatic tool like SmartBox (Nan et al., 2010) and O-snap (Arikan et al., 2013). The semi-automatic tool is a more practical way than normal CAD software. It can efficiently produce accurate models with the help of data and manual input. The object models, once modelled, can be used as templates. The semi-automatic tool should have the ability to automatically detect basic geometry shapes and their combinations and also provide the function to manually refine models.

The templates could be divided into two groups according their flexibility: rigid templates and elastic templates. The rigid templates request the reconstructed model exactly meet with predefined templates. The templates can be rotated and transformed. For example, the windows of a modern skyscraper building have an exact same shape. The elastic templates only ask the type of subparts and their topologies same, E.g., in reconstructing models of human faces, the topology of eyes, mouse, nose and ears is same, but their shape, size and relative positions vary with people. Some relative positions of the parts are fixed, like the two eyes are roughly horizontally placed and the mouse, nose, the mid of the two eyes and the mid of the two ears are roughly vertically placed. As shown in Figure 3-5, when modelling the rail track and its furniture, the power tower can be modelled by a rigid template while the rail track and the power line can be modelled by elastic templates.

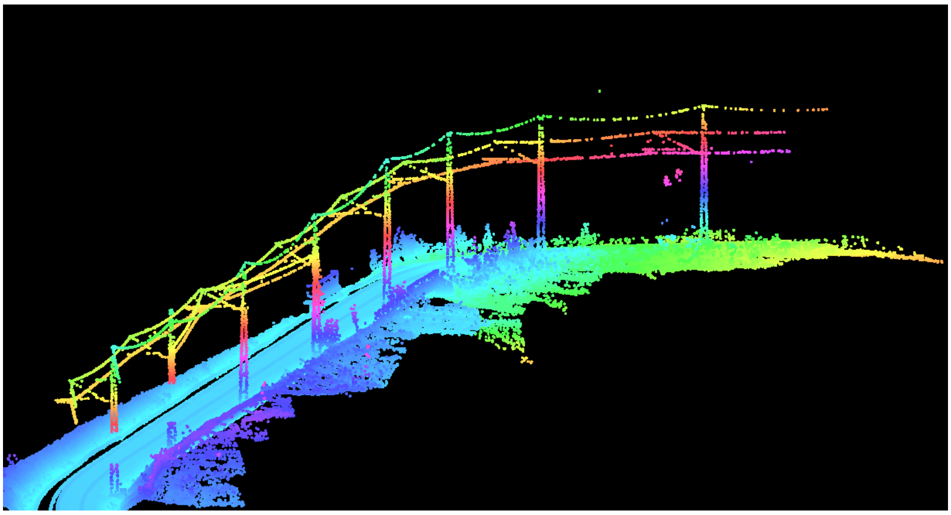


Figure 3-5. Rigid and elastic templates. When modelling the rail track and its furniture, the power tower can be modelled by a rigid template while the rail track and the power line can be modelled by elastic templates.

Another group of objects is that the main parts are the same but small parts are different. The small parts can also be modelled by solid templates. The whole object is modelled by combining a main part and small parts. Or the group is divided into several sub-groups according to the templates of the small parts. Each sub-group is manually defined by a rigid model. Then the modelling procedure is to directly identify the sub-group types.

The template matching can be done by a fully model-driven method or a combination of data and model-driven method. The model-driven method is to directly fit the template to the data. But the parameter space is very large, and the searching is high time consuming, especially when many templates are optional. The data-model combination method is to derive features in the data and then use the features to select candidate templates. The feature extraction is data driven, and the template matching is done in the feature space, therefore the whole strategy is faster. The features sometimes are wrongly detected. Therefore in the template matching stage a global optimization strategy should be taken to supress those errors. The data-model combined method requests the data is good enough to extract features. When the data is too poor, the model-driven method is a better alternative.

Template primitives are frequently used in roof reconstructions (Kada, 2009; Lafarge et al., 2008; Oude Elberink and Vosselman, 2009). The templates are usually called building primitives, which are elastic templates. The exact parameters of the building models are fixed when fitting to the data, like the height, length of the eaves and the intersection angle between roof faces. The building primitives are sub-part templates. A building is template matched part by part and combined to a complete building. This design is able to model a large variety of buildings with only a few building primitives. However, the state-of-the-art building primitives only provide a sub-set of the possible primitives. Many structures are not defined therefore cannot be accurately modelled. Chapter 4 will provide a complete library of building primitives based on the theory of the roof topology graphs.

3.4 Simple region regularities

Simple region regularities, including regularities of simple polygonal regions (2D) and simple polyhedral regions (3D), are the most basic regularities in modelling. They are the necessary qualities that valid geometry shapes must meet. Therefore they are frequently used in the self-checking of model qualities and model repairing (Brédif, 2010; Ledoux, 2013; Ohori et al., 2012). The regularities are guaranteed in the process of translating strings into geometric structures in the procedural modelling. However, these regularities did not get enough attention in the field of object reconstructing, especially in building reconstruction. Nowadays the range sensor and the technology that create edges and point clouds from images are good enough to produce dense and accurate data, which enable a much more data-driven strategy. The simple polygonal and polyhedral region regularities are met by all the valid models, therefore, can be uniformly used. In addition, the simple polyhedral regularities guarantee the models are watertight, enabling the definition of a complete building primitive library.

3.4.1 Simple polygonal region regularity

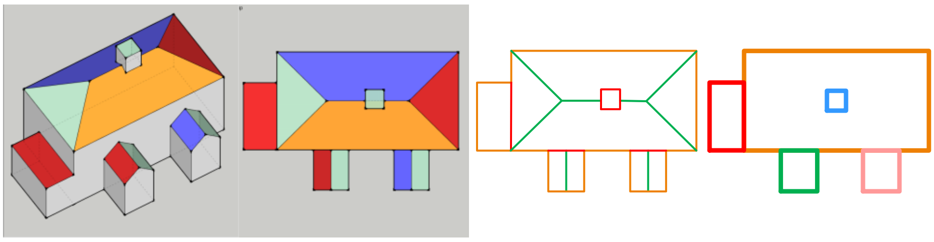


Figure 3-6. 2.5D polygonal regions, all outlined by simple polygons except the region outlined brown in the right image which has a hole.

In geometry, a simple polygon is a flat shape consisting of straight, non-intersecting line segments or "sides" that are joined pair-wise to form a closed path. A simple polygon divides the plane into two regions. The interior region is called a ***simple polygonal region,*** which is a connected region. All polygonal regions used for modelling buildings will be outlined by simple polygons. However, we also allow regions to contain holes outlined by simple polygons. A ***planar partition*** of a polygonal region is a sub-division of the region into non-overlapping polygonal regions. Planar partitions are frequently used in GIS to model concepts such as land cover, the cadastre, or the administrative boundaries of a given country. As each region has different height, the GIS polygons have 2.5D properties: in 2D they are partitioning the X-Y plane but they are located at different Z-levels. The 2.5D planar partition is used to model building parts without height jumps (Zhou and Neumann, 2010; Zhou and Neumann, 2011). See Figure 3-6 (right side), a building has several roof parts that are disconnected in 3D. Each part may be composed of one or more connected roof faces. All roof parts have no overlap in 2D and together form a complete building region. The partitioning of the X-Y plane is also frequently used to decompose a building region into sub-regions, which used to guide the independent reconstruction of sub-buildings(Haala et al., 2006; Lafarge et al., 2008; Vosselman and Dijkman, 2001; Zebedin et al., 2008a).

3.4.2 Simple polyhedral region regularity

In geometry a polyhedron is a 3D solid with straight edges and flat faces. A simple polyhedron is usually defined to be a polyhedron that is topologically equivalent to a sphere and whose faces are simple polygons. Therefore a polyhedron consists of just one piece. It cannot, for example, be made up of two (or more) basically separate parts connected by only an edge or a vertex. However, the constraint that the simple polyhedron is topologically equivalent to a sphere request the polyhedron has no hole inside. This constraint implies that many buildings cannot be represented. The simple polyhedron is defined in the view of connected region:

A simple polyhedral region is a connected 3D region that is enclosed by simple polyhedrons.

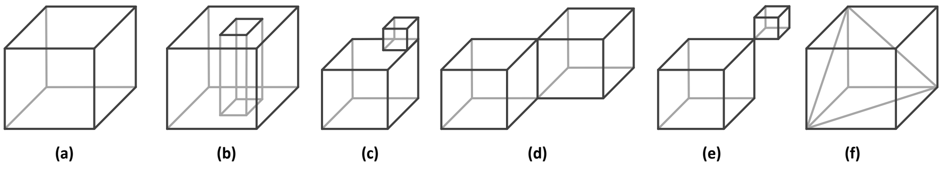


Figure 3-7. A simple polyhedral region and combinations of simple polyhedral regions.

The simple polyhedral region regularity is used to infer the adjacency relationships between roof faces and to reconstruct meaningful buildings. The roof planes are extracted from input data, including lidar point cloud, photogrammetric point cloud, DSM, or 3D lines, and then enforced to be a polyhedron model. Taillandier and Deriche (2004) thoroughly search the possible plane arrangement based on the detected 3D planes and get the most plausible building hypothesis that balances fit of the model to the data and the model's complexity. Chauve et al. (2010) adaptively decompose the 3D space into polyhedral cells by iteratively splitting the space along planes detected in the point cloud. Brédif (2010) corrects the roof topology graph of the input building models when fitting to new DEM. Lafarge and Alliez (2013) reconstruct polyhedron models from inexact adjacencies of surface planes.

The simple polyhedral region regularity ensures that each object has one connected 3D region. A building when considering both the outer and inner surfaces usually does not meet this constraint. However, each component can be seen as a simple polyhedral region. A building is usually divided into several corridors and rooms, which respectively have connected 3D regions (assuming that doors are closed). Therefore each corridor and room is modelled as a simple polyhedron.

Two adjacent simple polyhedrons that share a face can be joined to a single polyhedron (Figure 3-7 (c)). However, two adjacent simple polyhedrons that only share a vertex or a line cannot be joined to one simple polyhedron (Figure 3-7 (d) and (e)). Besides simple polyhedrons we will also use polyhedrons with holes to model buildings (Figure 3-7 (b)). A complex polyhedron is composed by several simple polyhedrons, like polyhedrons (d) and (e). A simple polyhedron does not allow a surface cut through it, like Figure 3-7 (f). It also means that the face cutting a vertex, an edge, or a face should not exist. The definition of simple polyhedral region makes that the vertex-based, the edge-based, and the face-based graph representations are equivalent. An illegal simple polyhedral region will break this equivalence. Topology graphs are a concise way to represent the structures between objects and their parts, without considering the geometry. Each topology graph has a minimum sub-graph basis, composed by a maximum group of linear independent sub-graphs, which have physical meanings. The following sub-sections first introduce this minimum sub-graph basis and afterwards discuss the physical meaning of the independent sub-graphs.

Minimum sub-graph basis

A topology graph G has three basic elements: loose nodes, loose edges, and minimum cycles. A loose node has no adjacent nodes, and a loose edge cannot be a part of cycle. A cycle of G is any closed walk of G with no repeated vertices and edges, where a closed walk is a sequence of vertices and edges with a common first and last vertex. A maximal set of linearly independent cycles is called a cycle basis (Berger et al., 2004). The linear combination of sub-graphs is defined as the sum of all the sub-graphs minus the overlapping edges and the corresponding nodes. All cycles can be represented by linear combination of the cycles in the cycle basis. While a graph may have many cycle bases, a cycle basis where the sum of the edges of the cycles is minimal is called a minimum cycle basis of G. The cycles which constitute the minimum cycle basis are called minimum cycles. We define the minimum sub-graph basis of G as the set of all loose nodes, loose edges, and minimum cycles. Figure 3-8 illustrates the linear combination of cycles, the cycle bases and the minimum cycle basis. The cycle basis in red dash frame is the minimum cycle basis on the right of Figure 3-8. Figure 3-9 shows an example of the minimum sub-graph basis of a topology graph.

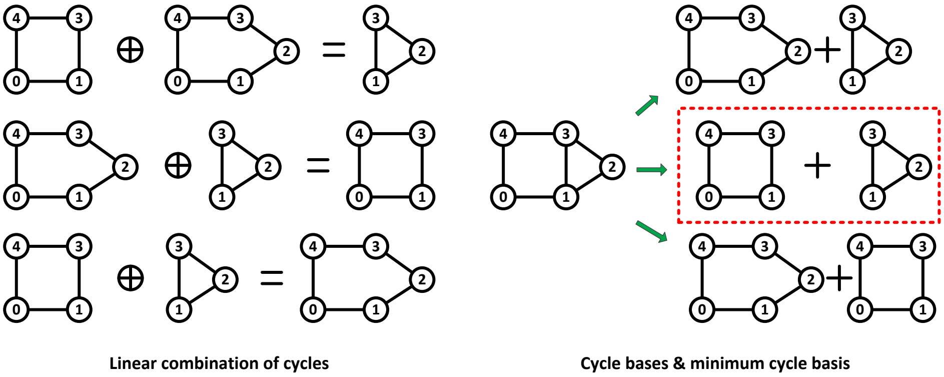

Linear combination of cycles
Cycle bases & minimum cycle basis

Figure 3-8. The linear combination of cycles, the cycle bases and the minimum cycle basis. The cycle basis in red dash frame is the minimum cycle basis.

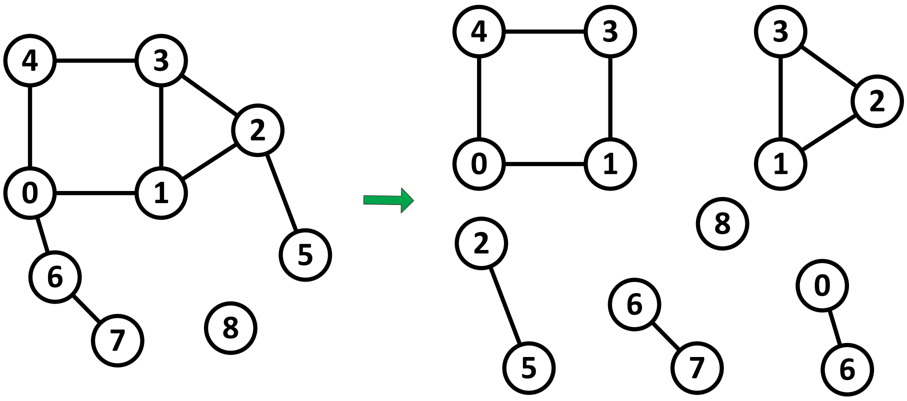


Figure 3-9. Minimum sub-graph basis of a topology graph. A minimum sub-graph basis is constituted by loose nodes, loose edges, and minimum cycles.

Physical meaning of minimum cycles

Minimum cycles have essential physical meaning in the topology graph representations of object faces. With the exception of information which is explicitly represented by nodes and edges, the other information is implicitly represented in the minimum cycle. There are three ways to represent building structures by a topology graph, Figure 3-10. The first one is face-based, in which object faces are represented as graph nodes, and edges intersected by faces are represented as graph edges. The second one is vertex-based. The vertexes are graph nodes and the edges between vertexes are graph edges. The last one is edge-based topology graph, in which edges are represented as graph nodes, and the adjacent relationships between edges are represented as graph edges. The three ways are mutually interchangeable. In the face-based topology graph, a minimum cycle (MC) is a vertex, such as MC $F_1F_2F_3$ is vertex V_2 in Figure 3-10 (b). And in the vertex based topology graph, a minimum cycle is a face, for example, MC $V_1V_2V_3$ is surface F_1 in Figure 3-10(c). In the first two representations, edges are explicitly represented as graph edges, whereas in the edge-based topology graph, only the edges are explicitly described, and the vertexes and faces are all implicitly represented as MCs. For example, in Figure 3-10(d), the MC $E_{12}E_{24}E_{14}$ is the face F_2, and the MC $E_{12}E_{24}E_{14}$ is the vertex V_1. If we can infer the MCs, then one expression can be transformed to other representations. This would mean that anyone can be used for a specific application.

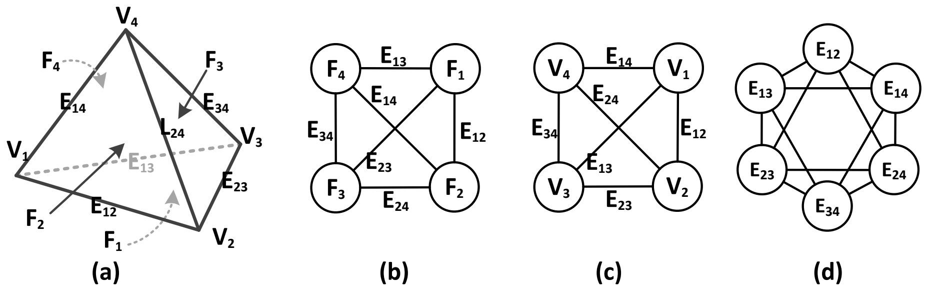


Figure 3-10. Three topology graph ways to represent an object. (a) a tetrahedron; (b) face based topology graph; (c) vertex based topology graph; (d) edge based topology graph. The three ways are mutually exchangeable. With the exception of the explicit nodes and edges, the other information is implicitly presented in the minimum cycle (MC).

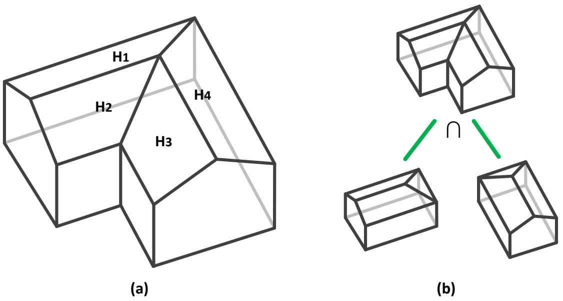
Given that a polyhedron always encloses a space, we get the rule:

The topology graph of a polyhedron does not have loose nodes and edges.

It means a polyhedron does not have loose vertexes, edges and faces. All the elements should form cycles in the topology graph. However, the roof model does not contain walls, therefore does not form polyhedron. The roof topology graph does not only constitute cycles. If the walls and the ground polygons are included, the building polyhedron will follow the rule. On the other hand, the inner corners of the roofs are always minimum cycles in the roof topology graphs.

A suitable topology graph should be chosen according to the applications and the availability of data. In the building reconstruction from airborne lidar data, the point cloud has more information on roof faces. Because roof faces can be detected reliably, the surface based topology graph is usually used to present the roof structures (Oude Elberink and Vosselman, 2009; Verma et al., 2006). The corners could then be inferred from the MCs. Fischer et al. (1997a) use vertex based graphs which contain 3D corners to integrate the roof structures, because in images the corners and lines are more reliably detected. Gruen and Wang (1998) use vertex based topology graph to group manually measured corners. This representation way is natural because corners are the input elements. Line based topology graph is rarely used. This maybe because lines are not easy to detect, as well as faces and vertexes are all implicitly presented in the vertex based topology graph. Too much information is hidden in this expression.

3.5 Three ways to interpret surface structure



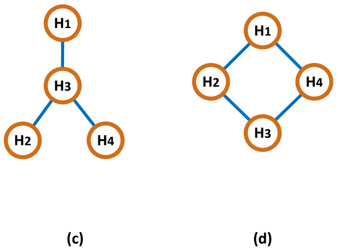


Figure 3-11. Three ways to interpret building structure. a) the building to be integrated; b) CSG model; c) BSP-tree; d) roof topology graph.

Three ways are used to represent building structures: constructive solid geometry (CSG) model, binary space partitioning (BSP) tree, and topology graphs (Figure 3-11). CSG model comes from the field of computer aided design. CSG allows an operator to create a complex surface or object by using Boolean operators to combine primitive shapes, which are usually cuboids, cylinders, prisms, pyramids, spheres and cones. The CSG model is not suitable to be used in the inverse engineering problem, reconstructing models from the observed data. The main problem is that the Boolean operations are ambiguous to detect. A complex shape has several ways to combine primitive shapes. And the structure of the CSG tree will also be complex, especially for the buildings which are composed by planar surfaces. For example, a simple L-shape building has four CSG representation ways with only the union operation, see Figure 3-12. It is because that the size of the primitives and their relative positions are unknown if they are freely composed. Therefore their combination can create diverse structures. In the CAD software, all the parameters and the Boolean operations are manually given therefore operators can get the exact model that they want. But meanwhile, by adjusting the parameters and the Boolean operations, the operators can also get any models based on the flexibility of the CSG tree. Shapiro and Vossler (1993) convert B-rep to CSG for a curved solid object by separating half-spaces, in which construction of a sufficient set of separators is a difficult task. And it is also a complex problem to interpret the CSG trees in the reverse engineering applications.

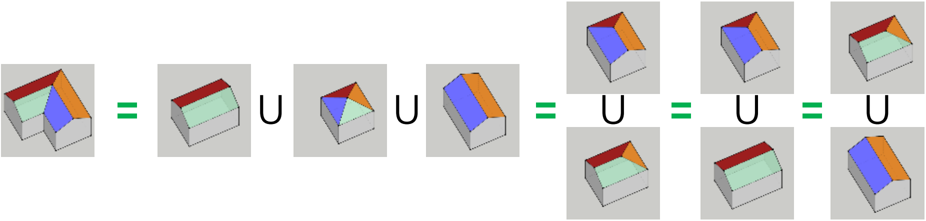


Figure 3-12. Ambiguity of CSG tree decomposition. A simple L-shape building has four CSG representation ways with only the union operation.

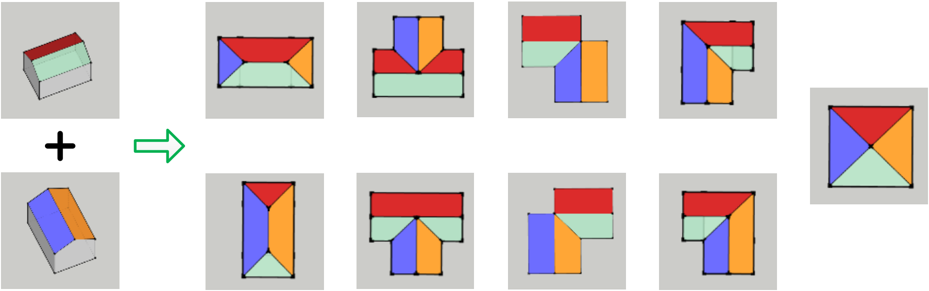


Figure 3-13. The diversity of the CSG model. A simple combination of two gable buildings creates nine building models by adjusting the size of the two gable buildings and their relative positions.

Another problem is there are not enough primitive shapes to represent all buildings. The buildings in the real world are not always regularly shaped. When only considering planar surfaces, the primitive shapes used in the literature are not diverse enough to represent the structures. For example, in the reconstruction of indoor scene, many works assume the Manhattan-world rules (Furukawa et al., 2009a). They assume the scene is constituted by piecewise planar with orthogonal dominant directions. Therefore the scene can be reconstructed by CSG model with primitive shapes of rectangle in 2D and cubic in 3D (Rusu et al., 2008; Xiao and Furukawa, 2012b). However, the assumption is hard to meet. The objects with arbitrary intersection angles could not be modelled. Because the types of primitive shapes are not enough, the CSG model will fall into the dilemma of model-driven methods: the predefined primitive buildings are always not enough.

The BSP is traditionally used in 3D computer graphics to increase rendering efficiency, including solid modelling, hidden surface removal, shadow generation, visibility orderings, and image representation. BSP uses (n-1)-dimensional hyper planes to recursively subdivide n-dimensional space is into homogeneous convex sets. Thereby a complex special scene is represented by a consequence of this hierarchical subdivision by a binary tree data structure. Sohn gives another meaning to it, and use the BSP tree as an object reconstruction tool to produce prismatic models of buildings (Sohn and Dowman, 2003; Sohn et al., 2008). Labatut et al. (2009) use BSP to build a hierarchical description of the scene. BSP is able to extend detected shapes to areas with holes, allowing the reconstruction of areas which may be missed. However, the partitioning will have different results if a different sequence of organizing is applied. And the corners, the essential elements of the models, are implicitly presented. As illustrated in Figure 3-11 (c), the corner of L-shape building has three layers of trees.

A given model can be structurally interpreted by the three ways. However, a model only has one unique topology graph. In fact, the topology graph is used to analyse the CSG or BSP-tree structures. The two methods are dual to the topology graphs. See the Figure 3-11, the topology graph is unique and unambiguous, and easy to be derived. In the vertex-based topology graph, faces and intersection lines are detected from raw data, and corners are inferred from the graph, a polyhedral model is then explicit presented. Therefore the topology graph is the recommended structure analysing ways in the reverse engineering. Its problem is that when the data has gaps and errors, the detected topology graph would have errors. Therefore the topology graph should be used only when the data is good enough. And we should always keep the idea in mind that the application should be robust to the errors in the topology graph or some special processing should be applied to correct the errors. The other two ways can also be used to improve the interpretation of the topology graph.

3.6 Conclusion

This chapter analyses the structure regularities of buildings, discusses their advantages and disadvantages, and gives hints about how to use these regularities. These regularities can be used to help building reconstruction and to improve the model qualities. They are viewed at different level therefore are largely overlapped. The line and face regularities are at lowest level and most flexible. The simple polygonal and polyhedral regularities are the necessary request to guarantee the reconstructed models to be valid. The template regularities enforce the observed data to be a few predefined primitives. It is a kind of model-driven methodology, therefore are a strict regularities. The hierarchical regularities are a bridge to connect the information from different levels. When only considering one level of detail, it is not as valuable as to reconstruct models with several levels of information. This thesis only reconstructs the LOD2 models therefore does not use the hierarchical regularities. Three ways to interpret the model structures are also discussed and compared. The author would like to recommend the topology graph, an unambiguous way, to interpret the model structures.

Chapter 4

Flexible Building Primitives and Their Applications in Building Modelling

Because of their high complexity of structures, no fully automatic system is currently available for producing LoD2 models of buildings. In order to simplify the problem, a lot of research focuses only on particular buildings shapes, and relatively simple ones. Based on the analysis of the properties of topology graphs of object surfaces, roof topology graphs are found that they have three basic elements: loose nodes, loose edges, and minimum cycles. These elements have interesting physical meanings: a loose node is a building with one roof face; a loose edge is a ridge line between two roof faces whose end points are not defined by a third roof face; and a minimum cycle represents a roof corner of a building. This chapter introduces a library of flexible building primitives based on the regularities of simple polyhedron and minimum sub-graph bases. Building primitives, which introduce building shape knowledge, are defined according to these three basic elements. Then all buildings can be represented by combining such building primitives. The building parts are searched according to the predefined building primitives, reconstructed independently, and grouped into a complete building model in a CSG-style, see Figure 4-1. The shape knowledge is inferred via the building primitives and used as constraints to improve the building models, in which all roof parameters are simultaneously adjusted.

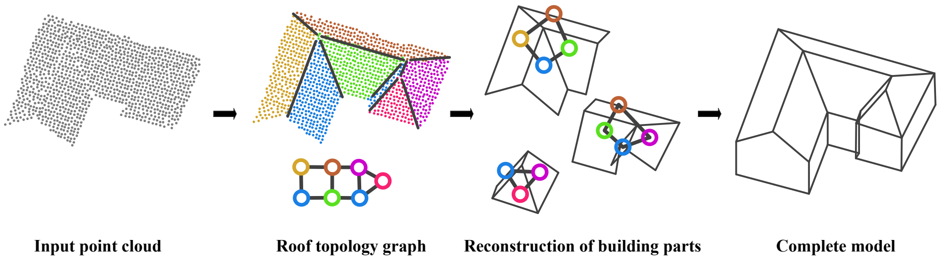


Figure 4-1. Workflow of the building model reconstruction from a point cloud by applying the new building primitive library. The point cloud of buildings is segmented into roof patches and the roof adjacency graph is thereby constructed. The building parts are searched according to the predefined building primitives, reconstructed independently, and grouped into a complete building model in a CSG-style.

The remainder of this chapter is organized as follows: Section 4.1 gives a definition of a building primitive library and introduces an algorithm that searches for building primitives. The building roofs follow some geometric regularities, which are applied to improve the quality of the reconstructed models. Section 4.2 gives a set of geometry regularities, shows the inference method, and presents the algorithm for applying constraints. Section 4.3 demonstrates how to obtain enclosed roof polygons by grouping the inner corners and inner boundaries, and the inferred outer corners and outer boundaries. As the whole algorithm is dependent on the roof topology graph, any error in the roof topology graph would seriously affect the quality of the model. Section 4.4 proposes a manual method of correcting any errors in the roof topology graphs. Section 4.5 discusses the results of the proposed method as applied to a dataset of Vaihingen (Germany) and Enschede (the Netherlands), including lidar point cloud and photogrammetric point cloud. The chapter ends in Section 4.6 with concluding remarks, a discussion of open questions and suggestions for further work.

4.1 Building primitive library

Model-driven methods are widely applied in reconstructing LoD2 building models (Dorninger and Pfeifer, 2008; Huang et al., 2011; Karantzalos and Paragios, 2010; Oude Elberink and Vosselman, 2009; Verma et al., 2006; Zebedin et al., 2008b). Model-driven methods reconstruct buildings with incomplete data and reconstruct all roofs at the same time. However, the usage of model-driven methods is limited because they require a library of prior defined models. Many buildings cannot be represented because of the high complexity of building structures. Verma et al. (2006) describe how a building can be represented as a group of assembled building primitives by analysing the roof topology graphs. They define primitive shapes as I-shaped, L-shaped and U-shaped, and search for these using a roof topology graph. Primitive shapes are limited so only relatively simple buildings can be described. Oude Elberink and Vosselman (2009) expand the topological primitive library to include gable, hip, gambrel, etc. The primitive shapes defined by Oude Elberink are finer in granularity than those defined by Verma and are, therefore, more flexible. However there are still many building shapes are not described in the library. A lot of other studies also use building primitives to integrate building structures (Hammoudi and Dornaika, 2010; Huang et al., 2011; Kada and McKinley, 2009a; Lafarge et al., 2010).

This chapter goes further and thoroughly divides buildings into basic elements of the topology graph: loose nodes, loose edges and minimum cycles. As any roof topology graph can be decomposed into such basic elements, buildings with a planar structure can be assembled according to the primitive shapes defined. To allow the largest possible flexibility, a building primitive library is designed so that the graph of every primitive corresponds to the minimum sub-graphs. The building primitives are also used to introduce geometric constraints to improve the model quality in the reconstruction process. This section shows the construction of a roof topology graph, provides a definition of the building primitive library, and indicates how to search for the building primitives.

4.1.1 Roof topology graph (RTG)

In the point cloud, roof segments can reliably be detected, so the face-based topology graph has been applied. A roof topology graph is constituted of roof segments (graph nodes) and their adjacent relationships (graph edges). A node represents a roof face, and an edge represents the adjacency relationship of two roof faces. If an intersection line exists between two roof segments, their corresponding nodes are connected by an edge. In order to precisely identify building types, the edges are labelled by attributes. The roof topology graph is an attributed relational graph, denoted by the 3-tuple $G=(V,E,\mu)$, where V is the set of vertices (nodes), E is the set of edges, and $E\subset(V\times V)$, μ is the attribute set of edges. The roof topology graph is illustrated in Figure 4-2. Roof segments are coloured randomly and represented by graph nodes. The graph edges are coloured according to their attributes.

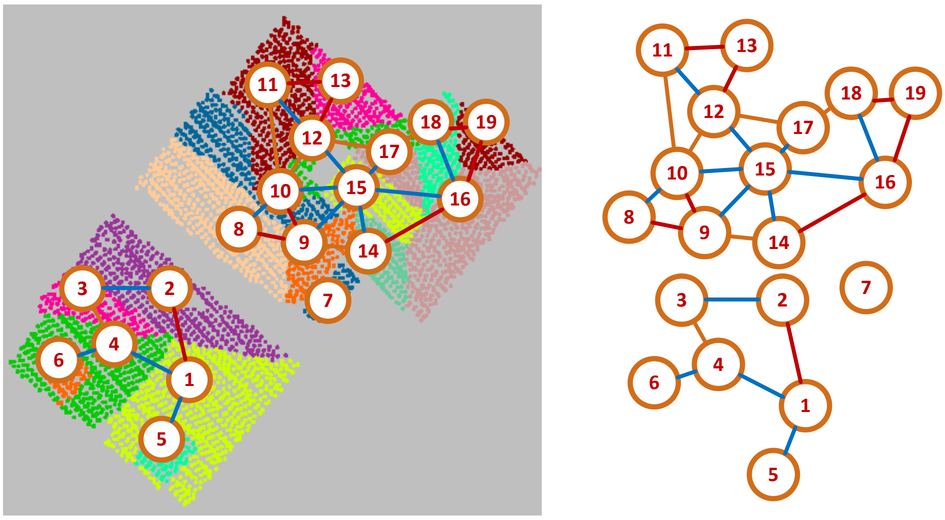


Figure 4-2. Roof topology graphs.

4.1.1.1 Roof Segments

In city scenes, most buildings can be modelled as a group of planar roof faces. A surface growing method (Vosselman, 2012) is used to segment the point cloud into homogeneous planar patches, which are called roof segments. The input point cloud is classified beforehand and only contains building points. Therefore, all segment patches are roof segments. The Hough transform is applied to the neighbourhood of a randomly selected point. When the point set fits well to a plane it is taken as a seed surface. Then points adjacent to the seed and within a given distance to the plane are accepted as surface extensions. This process iterates until no more points can be added to this surface. The remaining points are searched to form other planar patches in the same way. To correct overgrowing by the greedy surface growing, points initially assigned to a surface may be re-assigned to another surface when this surface better fits to the point and its neighbourhood. This surface growing algorithm is fast and robust even in a complex scene. The parameters of a roof plane are calculated by a least squares fit of a plane to all the points in the roof patch.

4.1.1.2 Intersection Lines

For each pair of nearby segments an intersection line is hypothesised. The hypothesis is verified by checking whether both segments have enough points close to the intersection line and by checking the length of the intersection line. The length of the intersection line is calculated by taking the overlap of two line segments computed on each roof segment. The points within a buffer around the intersection line are projected onto that line. Then the outmost points of the projected points are set as endpoints of the line segment. The thresholds of the number of laser points within the buffer area and the length of the intersection line are set to 10 points and 0.5m respectively. The buffer size is set to 0.5 m in our experiments. The thresholds and the buffer area size can be adjusted according to point density.

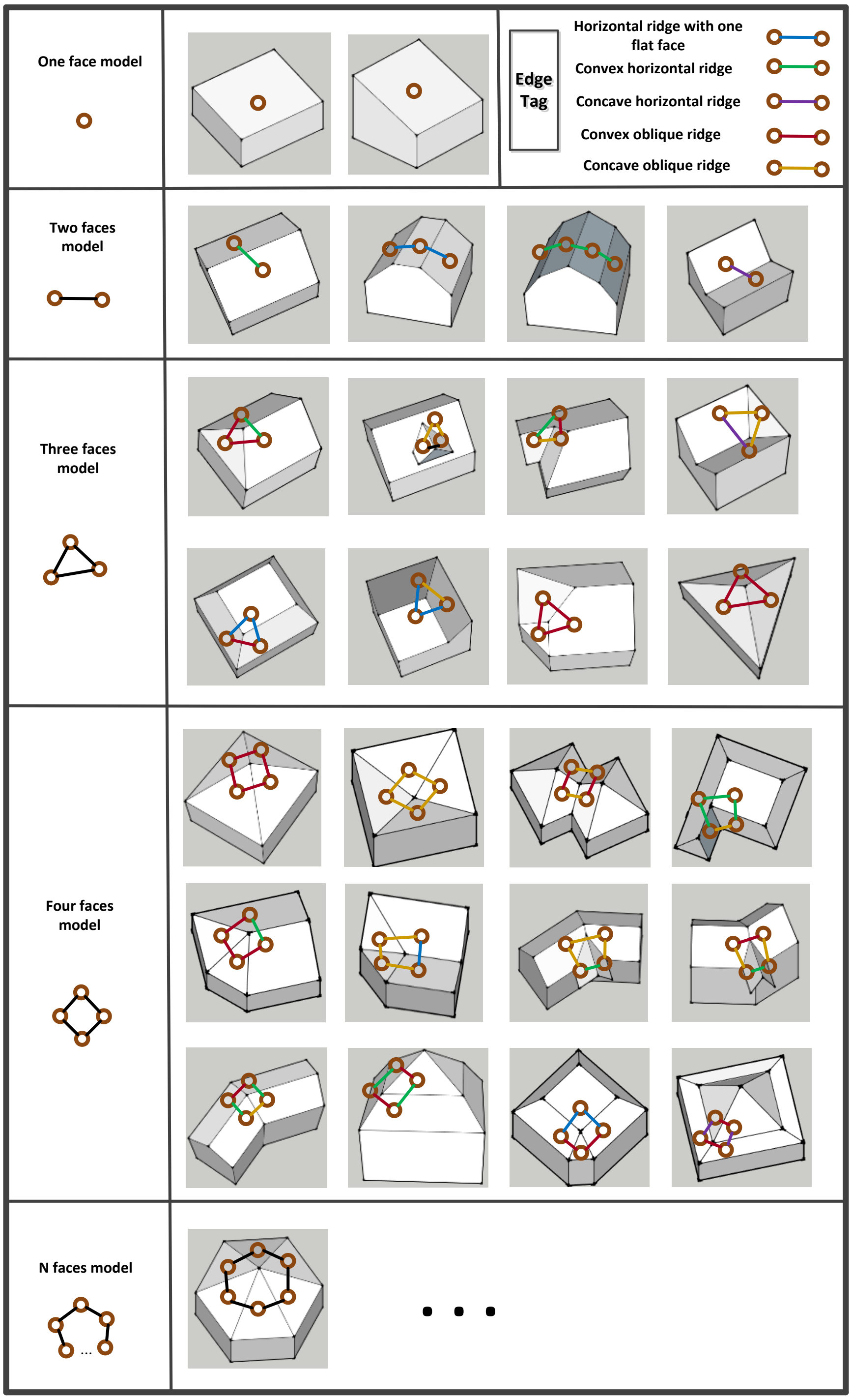


Figure 4-3. Definition of building primitives.

4.1.2 Building primitive definition

It is difficult to define a finite library of building primitives that can be used to describe all building structures. The complexity of building structures is extremely high when considering the various ways of composing roof planes and the unknown number of roof facets. As discussed in the Section 3.3.4, a building topology graph only has three basic elements: loose edges, loose nodes and minimum cycles. This means that all buildings can be divided into three basic types. Building primitives are defined according to these basic elements. The problem of building reconstruction is thus translated into the detection of building primitives and their composition. Roof segments that form a minimum cycle intersect at one point in 3D space. Consequently, building primitives are divided into corner building primitives and non-corner building primitives. A corner building primitive forms minimum cycles in a roof topology graph, while non-corner building primitives only form loose nodes or loose edges.

The enumeration of all loose nodes, loose edges and minimum cycles generates a building primitive library that is minimum but nevertheless complete. The building primitives are grouped according to the number of nodes, shown in Figure 4-3. The multiple instances of loose edges and cycles differ in the edge attributes. These attributes describe the relative geometry of the adjacency roof planes. The attributes are further applied to infer the geometry constraints which will be applied to improve model qualities. Five attributes are used here (shown in Figure 4-3). The first three attributes are for horizontal intersections, which are differentiated by roof geometry: the horizontal ridge has one horizontal face, the convex horizontal ridge has two sloped roof faces which form a convex angle, and the concave horizontal ridge has two sloped roof faces which form a concave angle. The last two attributes are for sloped intersection lines resulting from roof faces with a convex angle and those with a concave angle respectively.

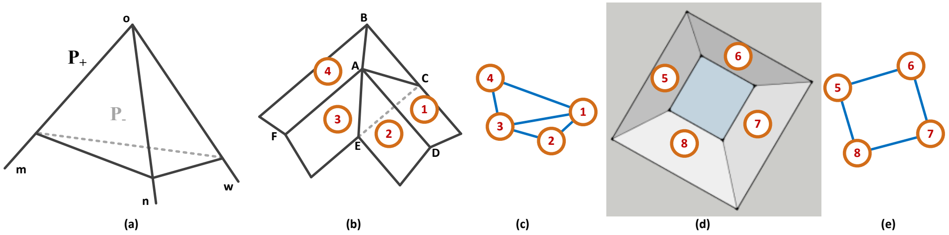


Figure 4-4. Cycle and non-cycle. (a) the space is divided into P+ and P- by corner O; (b) The erroneous edge between face 1 and 3 causes faces 1-2-3-4 are not considered to be a cycle; (c) the topology graph of model shown in (b); (d) Faces 5-6-7-8 do form a cycle, but should not form a single inner corner; (e) the topology graph of model shown in (d).

A corner building primitive can only be formed by a minimum cycle. This means that a minimum cycle sub-graph is a ***necessary*** - but ***insufficient*** - condition for a corner building primitive. The 3D space is separated by the half-planes that form the vertex into two parts: inside the corner and outside the corner. Illustrated in Figure 4-4 (a), the corner $S_{mon}S_{now}S_{mow}$ divides the space into two partitions $\boldsymbol{P_{+}}$ and $\boldsymbol{P_{-}}$.

***Necessary condition*:** If a group of roof segments forms a cycle but not a minimum cycle, then a sub-set of them can form a corner. The corner constructed by the sub-set of roof segments also divides the 3D space into two parts. And because the separation does not match the one formed by the whole set, the 3D space is then divided into more than two parts. In Figure 4-4 (b), planes 1, 2, 3 and 4 form a cycle. Because plane 1 and plane 3 intersect at line L_{AC}, planes 1, 2, 3 and planes 1, 3, 4 also form cycles, see topology graph shown in Figure 4-4(c). The space $\boldsymbol{S}_{ABF-AFE-AED-ABD}$ formed by cycle 1-2-3-4 is divided by plane $\boldsymbol{S}_{AEC}$ into two parts. The whole 3D space is split into three parts by planes 1, 2, 3, and 4. Therefore, a topology graph of a corner building can only be a minimum cycle.

***Insufficient condition*:** The roof segments form a finite area and do not extend infinitely as planes. If the roof segments do not extend to the position where they should meet, they will not form a corner even though they are a minimum cycle in the roof topology graph. In Figure 4-4 (d), planes 5, 6, 7 and 8 have a limited area and do not extend to their final intersection, resulting in topology graph shown in Figure 4-4 (e). This means they do not divide the 3D space into two parts. When searching for corner building primitives, the roof segments of a minimum cycle need to be checked as to whether they really do extend to the intersection point.

4.1.3 Search for building primitives

After constructing a roof topology graph, a building can be decomposed into building primitives by searching for loose edges, loose nodes and minimum cycles. As discussed in last section, a corner building primitive is always a minimum cycle in a roof topology graph, but a minimum cycle is not always a building primitive. Therefore, the search for minimum sub-graphs is a way of carrying out an initial search for building primitives. The exact building primitives are recognized by checking the geometry of corners and ridge lines. The minimum cycle (MC) search is for speeding up the search, because of its fast speed in topology space. The roof segments of a minimum cycle that has been found are checked to examine whether they really meet at one point. If the input lidar points are noisy, then it is certain that the roof planes do not meet at one point, and a buffer area is set in order to check the consistency. Even if the roof planes do intersect at one point, they may not form a building primitive as roof segments are not infinite surfaces. The roof segments are checked as to whether they extend to the intersection point. If a minimum cycle is recognized as a corner building primitive, the roof planes will be adjusted to be concurrent during subsequent refinement. As defined in the building primitive library, the property of edges is used to identify types of primitives precisely.

It is vital to search for minimum cycles in the application of minimum sub-graph basis. The search for loose nodes and loose edges is an easy process. What remains in the search for a minimum sub-graph basis is the search for the minimum cycles. An undirected graph has a cycle if - and only if - a depth-first search finds an edge which points to an already-visited vertex (a back edge). However, the minimum cycle could not be searched in this way. In order to speed up search, loose edges and nodes are first removed.

A breadth-first spanning tree ***T*** is constructed to help search for all minimum cycles which contain the node of interest (see Figure 4-5). The node of interest is set as the root of its spanning tree ***T***. If a node can form an MC with a node of its sibling branch, the cycle that its children form is not minimal. For example, in Figure 4-5 (b), cycle $\mathrm{C}_{\mathrm{N}_4\mathrm{N}_3\mathrm{N}_5}$ is an MC, but $\mathrm{C}_{\mathrm{N}_4\mathrm{N}_5\mathrm{N}_6\mathrm{N}_1\mathrm{N}_3}$ is not, because N_3 and N_5 are connected in the graph (Figure 4-5 (a)). So when searching, if a node cannot form a cycle, the algorithm will iteratively go to its children. After all MCs that contain node ***n*** are searched for, node ***n*** is removed from the graph ***G***, as well as other loose edges and nodes, for searching for remaining minimum cycles. The search continues until no edge and node is left in the graph ***G***.

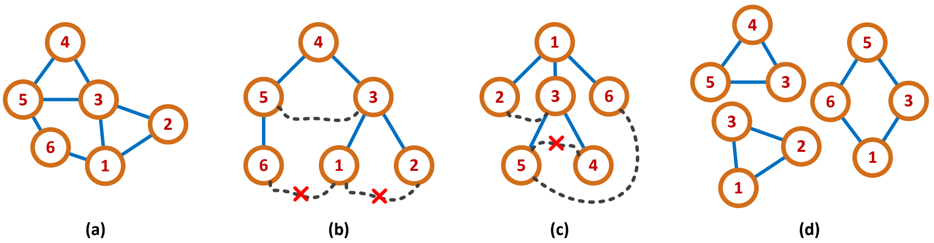


Figure 4-5. Search for Minimum Cycles. (a) Topology graph; (b) Spanning tree of Node N_4; (c) Spanning tree of Node N_1. (d) Searched minimum cycles. A dashed dark line represents a cycle, but only the one that is not crossed is minimal.

```
Algorithm: Search for minimum cycles in a graph
Input: graph $G = (N, E)$;
Output: minimum cycles $C$

For each valid node $n$ of $N$
----Invalid all loose nodes and edges
----Build spanning tree $T$ for $n$
----Initialize stack $S_1, S_2$

----For $l$ = all nodes of $n$ in $T$
--------Push $l$ into $S_1$
--------Push all other valid children of $n$ into $S_2$

--------While $S_1$ is non-empty
------------$l$ := pop $S_1$

------------While $S_2$ is non-empty
----------------$m$ := pop $S_2$
----------------If $m$ has edge with $l$
--------------------Construct $c$ with $m, l$, their ancestors,
--------------------up to root node $n$
--------------------Add $c$ into $C$
----------------Else
--------------------Push all children of $m$ into $S_2$
------------End while

------------if $l$ does not form cycles
----------------Push all children of $l$ into $S_1$
--------End while
----End for

----Invalid $n$
End
```

Table 4-1. Algorithm for minimum cycles search.

4.2 Geometry regularities and their inference

Architectures have strong geometric regularities. These regularities, which are also called structure knowledge, are fully used when the human brain interprets the observed world. However, we still do not know which regularity plays what role in that interpretation, and we do not know how human brains identify specific regularities. One reason for this is that there are too many regularities. As discussed in Section 3.3.3, this thesis only uses building geometry regularities that can be satisfied by most cases. The constraints include roof plane and boundary constraints. The roof plane constraints enforce symmetry and horizontal, concurrent and perpendicular intersections. The boundary constraints ensure identical heights, perpendicularity and parallelism of outlines. A constrained least-squares fitting (CLSF) algorithm is applied to adjust the roof planes according to the constraints for which a search is carried out while minimizing fitting residual. The boundary constraints are used to infer the outer boundaries and enclose the roof polygons, which is discussed in Section 4.3.

Roof planes can be refined if they satisfy the geometric constraint criteria. Roof constraints are divided into two types: 1) single roof constraints, 2) group roof constraints. Single roof constraints denote geometric regularities of a single roof, which can be vertical or horizontal. When the normal of a roof plane has a small angle, e.g. 5 degrees, with the plumb line or the horizontal plane, this roof will be taken to be horizontal or vertical. Group roof constraints mean group roofs satisfy some regularities. These constraints include the fact that intersection lines of two neighbouring roofs are horizontal or vertical, two connected roofs have equal slopes, and a group of roofs intersect at one position. The constraints used hare are common for the roofs of buildings. These features are searched for according to roof topology graphs. Every roof and building primitive for which a search has been carried out is checked as to whether it satisfies some geometric constraints.

The constraints are treated as adaptive constraints. This implies that the constraints are not strictly enforced based on the predefined models and are inferred according to the input data. The constraints are searched for inside the primitive models. They are local constraints, not global ones like GlobFit (Li et al., 2011), because local constraints are sufficient to improve reconstruction accuracy. More global constraints, like those inferred from repeated structures in buildings and from neighbouring buildings, are ignored as they would lead to a much higher complexity in the search for constraints.

4.2.1 Constrained Least-Squares Fitting (CLSF)

After the constraints have been found, roof faces should be adjusted simultaneously with the guidance of constraints to make all the roof planes fit the data well. As the noise of lidar points representing the roof plane is a Gaussian deviation, the solution of Least-Squares Fitting can provide the highest probability results. However, the CLSF cannot result in a linear solution due to the non-linear character of the constraints mentioned above. So an iterative process is reasonable. Werghi's optimization method (Werghi et al., 1999) is used here to solve the CLSF problem and extend it to include the concurrent constraint. The method is briefly introduced first, closely following (Werghi et al., 1999).

Suppose a set of surfaces is denoted $s_1, s_2, \dots, s_n$ and the set of parameter vectors is denoted as p. A surface fit error criterion J, e.g. the residual square sum, has to be minimized by each vector p associated with the surface S. The object function to be minimized by the set of parameter vectors is:

$$J = J_1 + J_2 + \cdots + J_n \tag{4.1}$$

Each surface s_i can be represented by a polynomial description of the surfaces in homogeneous coordinates:

$$h_i^T p_i = 0 \tag{4.2}$$

I.e., for a roof plane defined as $\boldsymbol{ax + by + cz + d = 0}$, the measurement vector becomes $\boldsymbol{h} = (\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z}, \boldsymbol{1})^{\boldsymbol{T}}$; and parameter vector $\boldsymbol{p} = (\boldsymbol{a}, \boldsymbol{b}, \boldsymbol{c}, \boldsymbol{d})^{\boldsymbol{T}}$.

The objective function minimizing the point to plane distances is written as

$$J_i = \sum_{j=0}^{M_i} (h_{ij}^T p_i)^2 = p_i^T \left(\sum_{j=0}^{M_i} h_{ij} h_{ij}^T \right) p_i = p_i^T H_i p_i \tag{4.3}$$

When all parameter vectors $\boldsymbol{p_i}$ are concatenated into one vector $\boldsymbol{p} = [\boldsymbol{p_1^T}, \boldsymbol{p_2^T}, \dots, \boldsymbol{p_n^T}]^{\boldsymbol{T}}$ the objective function becomes:

$$F(p) = J = p^T Hp \tag{4.4}$$

$$\mathrm{H} = \begin{bmatrix} \mathrm{H}_1 & 0 & \cdots & 0 \\ 0 & \mathrm{H}_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \mathrm{H}_n \end{bmatrix} \tag{4.5}$$

Under the above form, the data and the parameters are separately listed in the objective equation. An off-line computation can be done to get data matrix H before optimization. Because the matrix H is positive, this objective function is convex. If we need to minimize the objective function F subject to constraints C, then the problem is a CLSF problem. Theoretically, the problem can be solved by determining the set $(p^T, \lambda_1, \lambda_2, .., \lambda_M)$ that minimize the following equation:

$$E(p) = F(p) + \sum_{k=1}^{M} \lambda_k C_k(p)^2, \quad \lambda_k > 0 \tag{4.6}$$

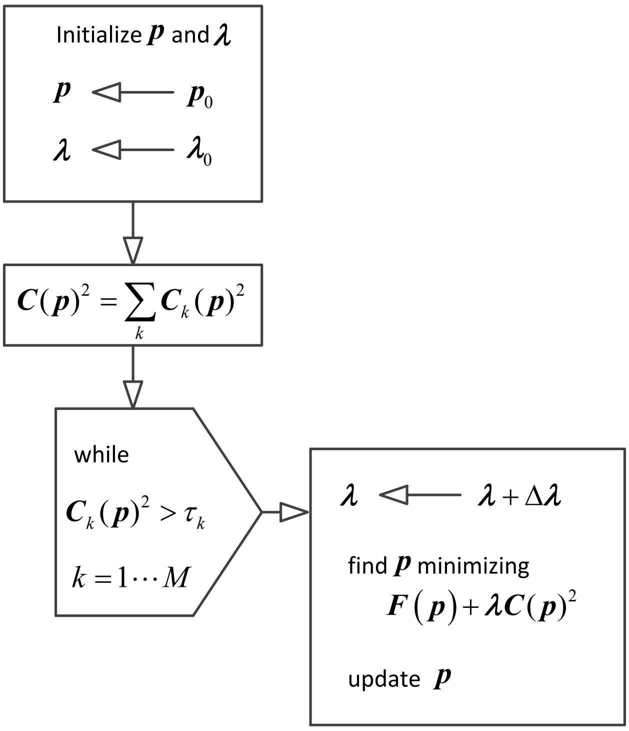

Initialize p and λ
$p \Leftarrow p_0$
$\lambda \Leftarrow \lambda_0$
$C(p)^2 = \sum_k C_k(p)^2$
while
$C_k(p)^2 > \tau_k$
$k = 1 \cdots M$
$\lambda \Leftarrow \lambda + \Delta\lambda$
find p minimizing
$F(p) + \lambda C(p)^2$
update p

Figure 4-6. Constraint optimization algorithm. (Werghi et al., 1999)

Even though the constraint functions $C_k(p)$ are not necessarily convex, the Hessian matrix of the squared constraint function is positive and definite, making the squared constraint function a convex function as well as the whole optimization function $E(p)$ in (4.6). Under the Khun-Tucker conditions (Fletcher, 2013), if the objective function and the constraint functions are continuously differentiable and the gradients of the constraint functions are linearly minimum, the solution of the system will be the optimal set $(p^T, \lambda_1, \lambda_2, .., \lambda_M)$ minimizing Eq. (4.6):

$$\frac{\partial F(p)}{\partial p} + \sum_{k=1}^{M} \lambda_k \frac{\partial C_k(p)^2}{\partial p} = 0 \tag{4.1}$$

The term $\sum_{k=1}^{M} C_k(p)^2$ serves as a penalty function that controls the constraints satisfaction. Shown in Figure 4-6, starting with a low value of λ_k the energy as defined in Eq. (6) is minimized with Levenberg Marquadt algorithm (Fletcher, 2013). Then λ_k is set to a higher value to give a higher weight to the constraint energy and $E(p)$ is again minimized. The process of incrementing λ_k and minimizing the redefined $E(p)$ continues until it reaches a very high value. At this point, solution p is the solution that minimizes the objective function $F(p)$ and meets the constraints C. For details of the optimization process, please refer to (Werghi et al., 1999).

4.2.2 Mathematic representation of the constraints

Table 4-2. Mathematical Representation of Constraints.

Constraint	Mathematical Representation	Parametric Matrices
Horizontal	$N_i \cdot N_p = 1$	$B_{(i+2)} = 1; B_{(m)} = 0, else$ $A = 0, C = \text{-}1$
Vertical	$N_i \cdot N_p = 0$	$B_{(i+2)} = 1; B_{(m)} = 0, else$ $A = 0, C = 0$
Orthogonal	$N_i \cdot N_j = 0$	$A_{(i,j)} = A_{(i+1,j+1)} = A_{(i+2,j+2)} = 1$ $A_{(m,?n} = 0, else$ $B = 0, C = 0$
Equal slopes	$\left(N_i \cdot N_p\right)^2 - \left(N_j \cdot N_p\right)^2 = 0$	$A_{(i+2,i+2)} = 1, A_{(j+2,j+2)} = -1$ $A_{(m,?n} = 0, else$ $B = 0, C = 0$
Horizontal intersection	$(N_i \times N_j) \cdot N_p = 0$	$A_{(i,j+1)} = 1, A_{(i+1,j)} = -1$ $A_{(m,n)} = 0, else$ $B = 0, C = 0$

In the CLSF, constraints need to be continuously differentiable, which is satisfied by all the constraints. The following paragraphs list their Jacobi matrix (first-order derivative) and Hessian matrix (second-order derivative). Here the direction of the plumb line is denoted as $\mathrm{N_p}$, $(0,0,1,0)$, which is in homogeneous coordinates. The parameter for each roof plane is written as p. With the exception of concurrent constraint, all other constraints can be represented by expressions containing cross-product terms:

$$C\left(p\right) = p^T Ap + B^T p + C \tag{4.2}$$

Where A and B are, respectively, a square matrix and a vector, with the same dimensions as the parameter vector p, C is a scalar. Its Jacobi and Hessian matrix is simple:

$$J\left(C\left(p\right)^2\right) = 2\left((A + A^T)p + B\right)\left(p^T Ap + B^T p + C\right) \tag{4.3}$$

$$H\left(C\left(p\right)^2\right) = 2(A + A^T)\left(p^T Ap + B^T p + C\right) + 4\left((A + A^T)p + B\right)\left((A + A^T)p + B\right)^T \tag{4.4}$$

Table 4-2 lists a mathematical representation of constraints and their feature matrices.

The concurrent constraint differs from other constraints in the form of constraint function. The concurrent constraint means several planes (ai, bi, ci, di) go through a single intersection point (X, Y, Z). Each plane satisfies the equation:

$$(a_i, b_i, c_i, d_i)(X, Y, Z, 1)^T = 0 \tag{4.5}$$

By introducing intersection points (X, Y, Z) as unknowns in the CSLF optimization, each concurrent constraint is grouped by several coplanar constraints shown in Eq.(4.5) By concatenating all parameter vectors $\mathrm{p_i}$ and intersection point vectors (X, Y, Z) into a single vector $\mathrm{p} = \left[\mathrm{p}_1^\mathrm{T}, \mathrm{p}_2^\mathrm{T}, \ldots, \mathrm{p}_\mathrm{n}^\mathrm{T}, \mathrm{X}, \mathrm{Y}, \mathrm{Z}\right]^\mathrm{T}$, constraint function can be written in the form of Eq.(4.2). For each plane (ai, bi, ci, di), we get the equation:

$$\begin{bmatrix} a_i \\ b_i \\ c_i \\ d_i \\ X \\ Y \\ Z \end{bmatrix}^T \begin{bmatrix} 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} a_i \\ b_i \\ c_i \\ d_i \\ X \\ Y \\ Z \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \\ 0 \\ 0 \\ 0 \end{bmatrix}^T \begin{bmatrix} a_i \\ b_i \\ c_i \\ d_i \\ X \\ Y \\ Z \end{bmatrix} = 0 \tag{4.6}$$

4.3 Roof polygons

The roof polygons are obtained by sequentially ordering all inner ridges, inner corners, outer boundaries and outer corners for each roof plane. The inner ridges and inner corners are already obtained by intersecting the roof planes. Outer corners are computed by assuming one auxiliary plane, which may be horizontal or vertical, and intersecting it with the two adjacency roof planes. Meanwhile, their intersection lines are determined. The new intersection lines and ridge lines still only provide parts of the polygon boundaries of roof faces. The remaining boundaries are hypothesized by inference based on the available boundaries. The hypothesized boundaries are constrained to be parallel and/or perpendicular to context boundaries. Figure 4-7 shows the workflow of enclosing roof polygons. The roof models are complemented with wall surfaces by projecting the roof outline onto the ground when a DTM is available, or onto a ground plane with a given height.

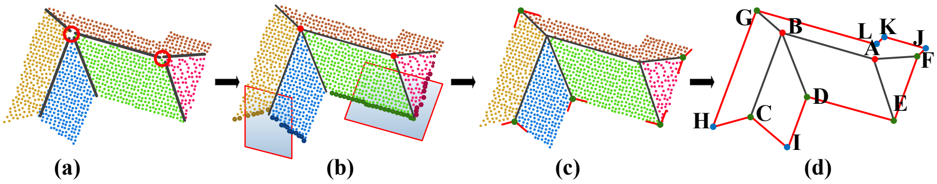


Figure 4-7. Enclosing Roof Polygons. (a) Fixed inner corners. (b) Estimate a vertical or horizontal plane. (c) Determine outer corners and boundary lines. (d) Enclose the roof polygon by connecting all boundaries.

4.2.3 Detection of outer corners and boundaries

There are two types of line segments of the roof polygon: inner boundary lines and outer boundary lines. An inner boundary line is the intersection line formed by two adjacent roof segments. The inner boundary lines are detected when the roof topology graph is constructed. Vertices of the roof polygon are defined as inner corners if they are formed by three or more roof planes. The other vertices are defined as outer corners. Inner corners, which are found by searching for building primitives in the topology graph, are calculated by intersecting the concurrent planes (step 1 in Figure 4-7).

All corners in 3D space are formed by three or more concurrent surfaces. Therefore, outer corners are computed by assuming one auxiliary plane, which may be horizontal or vertical, and intersecting it with the two adjacent roof planes. The auxiliary plane is estimated by RANSAC detection from the points on the boundaries of the two supporting roof faces. When the third plane is estimated, the outer corner can be computed by intersecting the three planes. Sampath and Shan (2010) also find this knowledge, and introduced two vertical planes that pass through the building edges to join the two roof planes, thus providing the outer corners. However, because the two vertical planes are estimated independently, this method cannot ensure that the four planes are concurrent, and the resulting corners are less precise than when all boundary points are used.

With the exception of the outer corners, the estimated plane can provide outer boundary lines by intersecting with the supporting roof planes (step b and c in Figure 4-7). In step d of Figure 4-7, vertexes A and B are inner corners; vertexes C-L are outer corners. Line segment AB is an inner boundary, and all other line segments are outer boundaries. However, the outer corners and outer boundaries detected by hypothesizing horizontal or vertical planes only provide parts of the outer corners and boundaries. For example, in step d of Figure 4-7, outer corners H and I, and outer boundaries JK and KL could not be detected. The remaining corners and boundaries are hypothesized by inferring the available boundaries, and are discussed in the following subsection.

4.2.4 Roof face outlining

After the detection of outer corners and boundaries, the next step is to hypothesize boundaries to link all available boundary lines to obtain enclosed polygons. The hypothesized boundaries are constrained to be parallel and/or perpendicular to context boundaries, if the constraints could be met. The context boundaries for one boundary are all other the available boundary lines on the same roof face. Line pairs whereby one end node is not attached to another line are searched for in order to snap them together. The distance between a pair of lines is used to evaluate the possibility that two lines can be snapped together. The distance of a line pair is the distance between the two lines if they are parallel, otherwise it is the average distance from the intersection point to the nearest endpoint of both lines. If one line pair is found, four ways are introduced here for snapping them together according to different situations. The information required for the four different operations is angle and distance between the outer boundaries. The four operations are demonstrated in Figure 4-8 and corresponding examples are showed in Figure 4-9.

1) **Intersect.** Two outer boundaries intersect at one point if they are not parallel. The intersection point is set as the final endpoint of both boundaries. In our implementation, the criterion for checking whether the two boundary lines are parallel is set to be 15 degrees.
2) **Merge.** If two outer boundaries are parallel and close to each other, they should be on a single boundary line. The merge operation is implemented by simply linking the two initial endpoints. The distance threshold is set as 0.3 meter.
3) **Expand.** Two parallel outer boundaries are considered to be two different boundaries if they are not near to one another. They both expand following their ray direction until the farthest lidar point on the roof segment. A third line segment is created by linking the far endpoint of the two boundaries.
4) **Rectify.** The situation is very similar to the one of the expanding operation, the only difference being an inverse ray direction. An edge gradient, based on the ratio of the number of points of one roof segment on both sides of the edge, is used to determine the stability of the edge. The higher the edge gradient becomes, the more likely an edge is present. The most unstable boundary line is rectified and constrained to be perpendicular to the other one. Then the two boundaries can be treated as the situation of an intersection.

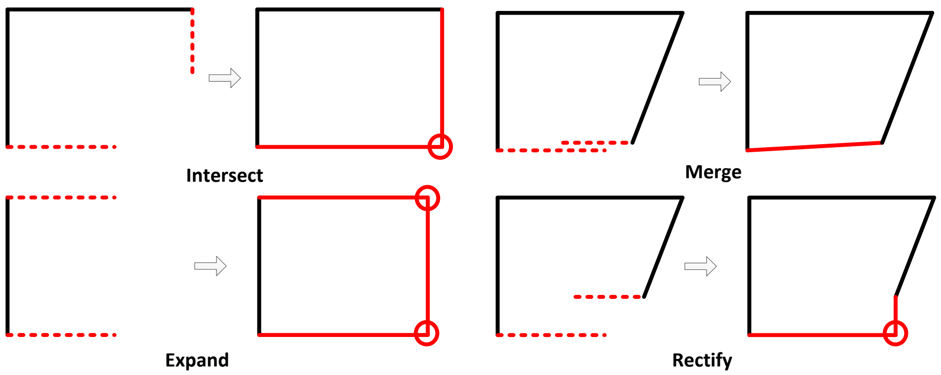


Figure 4-8. Four ways for enclosing a roof polygon. Dark lines are stable, red dashed lines need to be enclosed and red lines are after enclosing.

A special situation that should be considered is a building with one roof segment. This kind of building has no intersection line, so it cannot be reconstructed as a building with multiple roof segments. In this thesis, this type of building was reconstructed by boundary approximation and regularization. The α-shape algorithm (Bernardini and Bajaj, 1997) is applied to detect points on the roof boundary, and then key points are identified by picking points with a large curvature. The discrete curvature is computed for each vertex of the boundaries. A line segment is constructed using the two neighbouring vertices of a vertex, and the curvature $\boldsymbol{\kappa}$ is defined as $\boldsymbol{\kappa} = \boldsymbol{d}/\boldsymbol{l}$, where $\boldsymbol{d}$ is the distance of the vertex to the line segment, and $\boldsymbol{l}$ is the length of the line segment. All key points of a roof boundary are connected sequentially to result in a closed polygon. Then the polygon is refined according to the building principles direction and line constraints, such as parallelism and perpendicularity (Matei et al., 2008; Sampath and Shan, 2010). This boundary is already closed so does not need to be processed by the four enclosing operations.

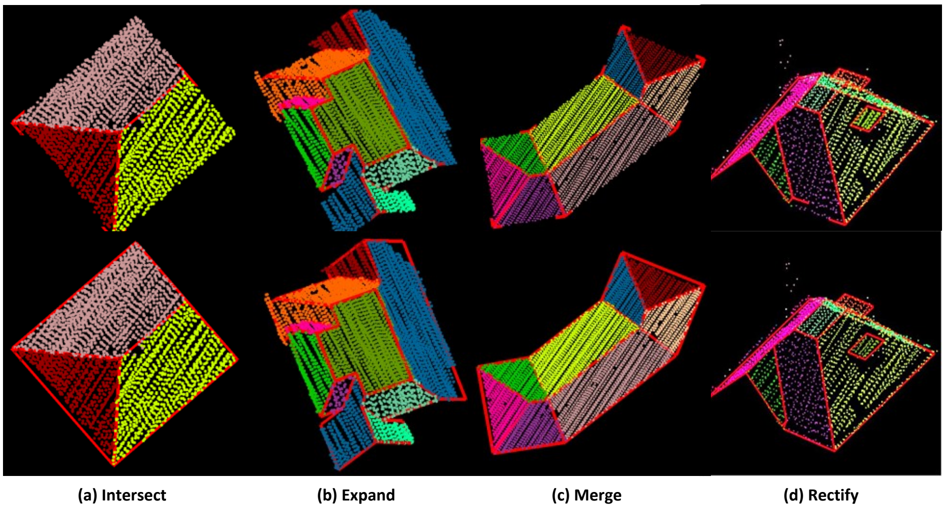


Figure 4-9. Estimated outer boundary lines and enclosing results.

4.4 Manual correction of roof topology graphs

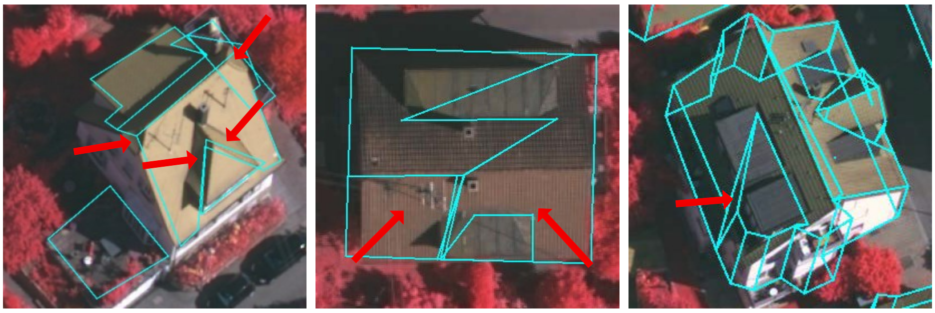


Figure 4-10. Incorrect models reconstructed from erroneous roof topology graph. From left to right: small roofs, wrong segmentations, incorrect combination of planes. (Rottensteiner et al., 2014)

The roof topology graph is essential in the proposed building reconstruction algorithm based on the new library of building primitives. However, errors in the topology graph are inevitable and will affect building reconstruction negatively. Because of outliers and occlusion, as well as inevitable defects in the pre-processing steps, such as classification, segmentation and intersection detection, the topology graph will always have errors, which limits correct construction. A tool has been designed to edit graphs interactively. For the four elementary errors, operators simply have to locate the errors and select the related features as well as the error type. The remaining work is carried out automatically.

Errors in the roof topology graph are inevitable and typically lead to incorrectly reconstructed building models (Rottensteiner et al., 2014). Figure

4-10 shows examples of incorrect models reconstructed from erroneous roof topology graphs. This section discusses the basic error types in the graphs, reasons for those errors and a tool developed to interactively repair roof topology graphs.

4.4.1 Four error types

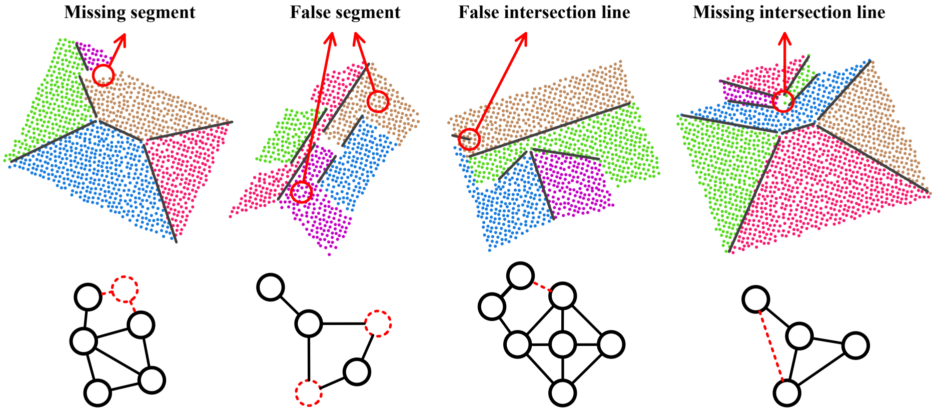


Figure 4-11. Errors in the roof topology graphs. Up: the roof segments of roof plane. Bottom: the roof topology graph. The edges and nodes in red dashed lines are the incorrect ones.

The errors in the roof topology graph can be classified into four basic types: false node, missing node, false edge, and missing edge (see to Figure 4-11). False and missing nodes correspond to incorrect and failed detection of shape primitives, which are planar roof segments in our work. No matter how accurate the detection of primitive shapes is, some of them will still be wrong or missed. The graph edges, which present the adjacency relationship of the shape primitives, are searched according to the predefined criteria. These criteria cannot be formulated such that all relationships are correctly detected. Furthermore, incorrect shape primitives will also result in incorrect adjacency relationships. The cause of these errors is discussed in more detail below.

a) ***Missing node.*** Small roof faces are often not well captured when the point density is low or when scanning direction is parallel to an elongated roof face. Low point densities also occur on roofs with reflecting materials (glass, solar panels) and absorbing materials (pitch, slate) and on flat roofs covered by water. These roof faces are often not detected. Narrow roofs are sometimes also missed because points may already have been assigned to the segments of their adjacent roof faces.
b) ***False node.*** This error is mainly caused by the roof segment detection process. When strict thresholds are set, one roof face may

be detected as several ones. Besides, occlusion sometimes splits points on one roof face into disconnected parts, which will then be taken as several roof segments.

c) ***Missing edge.*** The adjacency relationship is determined by checking the relative position of roof segments and the length of their intersection line. The number of connecting lidar points is used as criteria as well as intersection length. If two roofs only have a few points connecting them, or their intersection line is too short, these two roofs are not considered to be adjacent. These criteria sometimes overlook small roof face edges.

d) ***False edge.*** A false edge will be determined if two roofs that should not be adjacent are detected to be connected. This is typically caused by the wrong detection of roof segment points. For example, the lidar points of a tree will be assigned to the building if it is near the building. Then the roof segment may incorrectly connect other roofs. Small objects on a roof will also result in incorrect intersection lines. False edges are the most frequent errors in the topology graph.

4.4.2 Manual editing tool

As discussed, it is difficult to correct automatically the errors in the roof topology graph. The practical way is to semi-automatically fix them. In this section, a group of tools is developed to interactively correct the four types of errors. The operators just need to point out the location of the errors in the roof topology graph and identify the type of error. Then the computer will correct the errors automatically. The correction sequence is unlimited. The operators could randomly choose one error to correct.

1) ***Remove an intersection.*** If an intersection line is checked by an operator to be wrong, then it will be simply removed.
2) ***Restore an intersection.*** An intersection line will be created where two roofs are checked by an operator to be adjacent. The intersection will also be evaluated by checking whether the two roof segments are nearby and not parallel.
3) ***Merge roof segments.*** Two selected segments will merge into one when an operator tells the computer that they belong to one roof. The roof plane will be recalculated according to all lidar points of the two segments. The intersections of the two segments with other segments will also be grouped together, and their geometries are adjusted according to the new roof plane.
4) ***Restore a roof segment.*** When an operator finds one roof segment is missing, then select its nearby roof segments to be support planes. The support planes give clues about the missing roof. The computer will use the support planes to restore a reasonable roof plane, and get intersections with its adjacent roof segments.

The missed roof plane is estimated by automatic inference from its supporting roof segments. Figure 4-12 illustrates the restoration of a lost roof segment. With interactive input, computer knows one roof segment is lost; its next work is to estimate this roof. The missing roof is estimated by testing the points on the boundaries of the supporting roofs (Step b in Figure 4-12). All possible planes are estimated by RANSAC detection from all the boundaries. Then only the most likely one is kept. In the example of Figure 4-12, one plane is perpendicular to the ridge (top one in Step c), and another one is parallel to the ridge (bottom one in Step c). The perpendicular one is vertical, therefore, should be a wall. And the parallel one, which is under the other two roofs, rarely occurs in reality. Both the two roof planes should not exist. Therefore, the last plane, which is oblique to the ridge, is chosen as the final estimated roof plane (Step d).

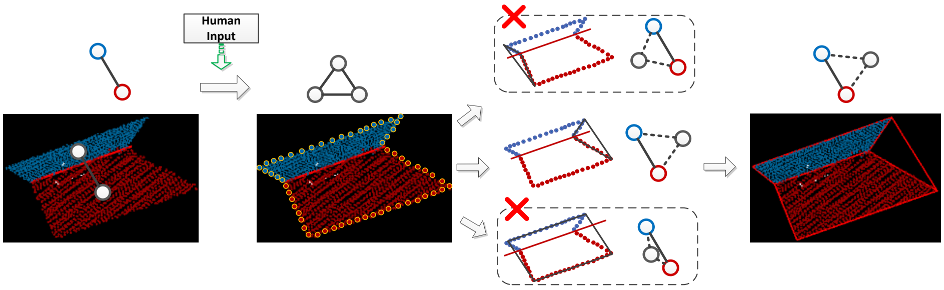


Figure 4-12: Restore a roof face. (a) Input roof patches, intersection line, and corresponding topology graph. (b) The goal graph is given by manual input and boundary points (high light points on the roof boundary) on the two supporting planes are automatically detected. (c) Three possible estimated planes. Only middle one is kept because the left two ones should not happen. (d) The final models and roof topology graph.

4.5 Experiments

4.5.1 Results of the new building reconstruction method

A dataset of Enschede is selected to fully test the new building reconstruction method. The city is characterized by residential buildings with diverse structures on the outskirts and a small number of commercial buildings in the city centre. The laser scanning data has been acquired by the FLI-MAP system of Fugro Aerial Mapping B.V at a flight height of 275 m and with a 30% strip overlap. The point density is on average about 20 points/m^2. 3 test sites with a total of about 500 buildings are selected for the performance test of the new building primitive library. The building reconstruction method designed by Oude Elberink and Vosselman (2009) is used to compare with the new method. The new building primitive library and a fully automatic method are used for reconstructing the building models, including the graph edit dictionary, to correct the erroneous roof topology graph.

In order to decide whether the quality of a building model is acceptable we need a single criterion for the complete building model. This measure should take into account that small deviations of a model from reality are allowed. These could be either small building parts, like chimneys or antennas that do not need to be modelled, or small geometric discrepancies between reconstructed and actual roof planes. Therefore a poorly reconstructed building model is defined as a model with an area of at least μ m^2 in which the lidar points deviate by more than ν m from the model. Reconstruction failures in areas smaller than μ m^2 are therefore ignored. It is assumed that these failures will not significantly impact the usefulness of the reconstructed model. In the experiments the values μ = 5 m^2 and ν = 0.3 m. After reconstruction, the distance to the model is calculated for every lidar point. A connected component analysis is applied to the points that deviated from the model by more than 0.3 m. Where the component sizes are below 5 m^2, the building model is considered acceptable. This quality criterion does not make use of reference data and hence may not capture all errors. It serves, however, as well as a self-diagnosis for the reconstruction method.

Table 4-3. Model improvements due to the new method.

Area	Number of buildings	Number of points	Method	Incorrect building models (%)	Average absolute residual
Area 1	61	182502	old	25	0.39
			new	5	0.04
Area 2	188	695446	old	35	0.22
			new	6	0.03
Area 3	224	731404	old	29	0.21
			new	3	0.07

Table 4-3 shows the difference in quality of models reconstructed using the old reconstruction method and using the new one. The old building primitive library is proposed by (Oude Elberink and Vosselman, 2009) and the new one is being introduced in this thesis. The percentage of poorly reconstructed building decreases from about 30% to about 5%. The average absolute point residual also shows improvements by evaluating the test site as a whole. E.g. in area 1, the average absolute residual decreases from 0.39 to 0.04. With the new method, we can see a correct reconstruction of about 95% of buildings and the models fit well to the input data. Figure 4-13 shows several examples of buildings that cannot be represented by the primitive library used in the old method but are well-reconstructed using our method. The most common cases relate to buildings with more than 4 faces. The reason for this is that the old library does not include such primitives. The building primitive library plays an important role in the building reconstruction. Only the buildings which are already defined in the library can be well reconstructed. Buildings which have not been predefined will be modelled by the most similar ones. Therefore the resulting models often appear to be well-reconstructed in visualizations, though in fact they are not correct. In theory, the library introduced can represent all buildings with plane surfaces. Therefore, most buildings will be modelled correctly.

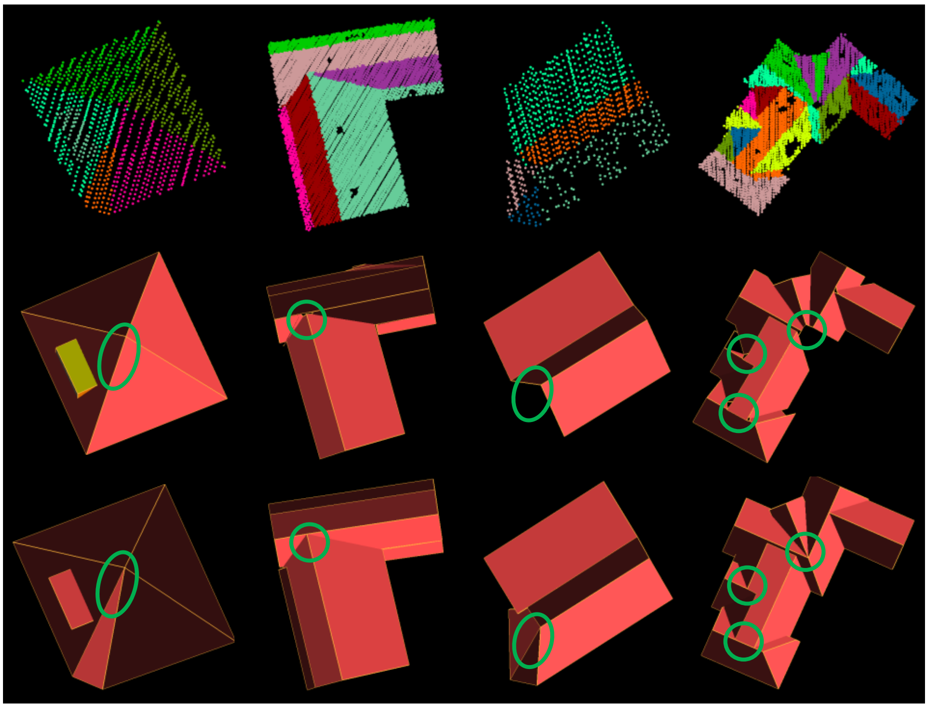


Figure 4-13. Model improvements due to the new library. First row: lidar points coloured by surface segments; Second row: models reconstructed using the old library; Third row: models reconstructed using the new library.

The improvement only resulting from the new library definition is not very large in the three test areas: only a few buildings could not be represented by the library defined by Oude Elberink and Vosselman (2009). A larger improvement lies in the fact that, with the old library, many roof faces are just not well outlined even though correct primitives have been found. In

Figure 4-14 (a) some ridges are incorrectly extended, leaving many points outside the roof polygons. In Figure 4-14 (b) the ridge line is given incorrect endpoints because of chimneys at both sides of the ridge line. In Figure 4-14 (c) several ridge lines are indicated on one roof plane, and their sequence is not correctly detected. This results in failure to create a closed polygon. In Figure 4-14 (d) and (e), one ridge line is incorrectly detected and one ridge line is not found. These problems are solved by the new roof face outlining algorithm and the roof topology graph correction algorithm.

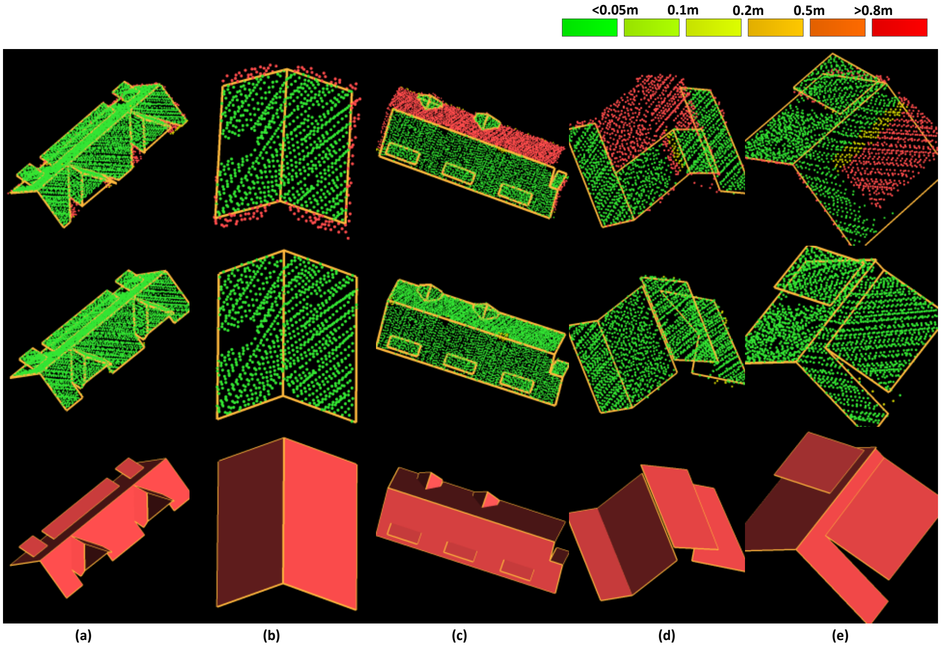


Figure 4-14. Model improvements by the new polygon-enclosing algorithm. First row: roof polygons reconstructed by old method and point distances to the models; second row: roof polygons reconstructed by the new method and point distances to the models; third row: the correct roof models. The lidar points are coloured by the distance to the model.

4.5.2 Geometric constraints

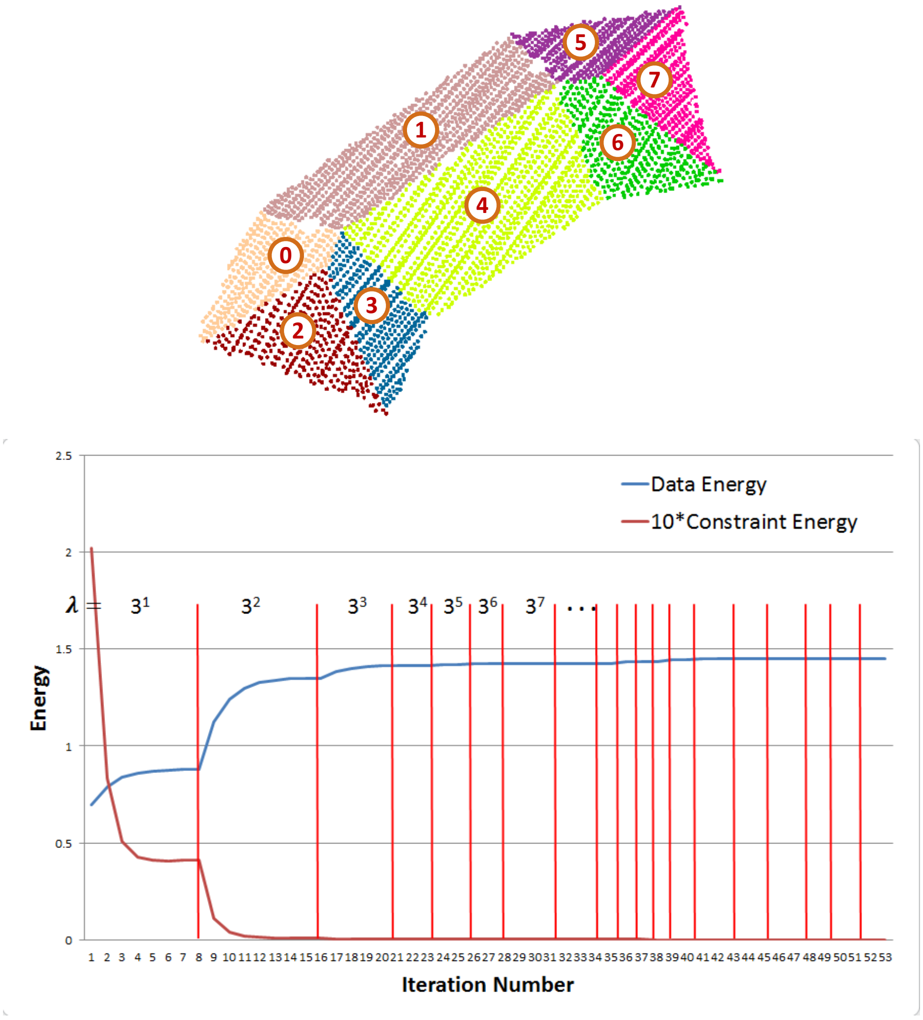


Figure 4-15. An example of constraint least-squares fitting. Left: the input roof segments; Right: the convergence of data energy and constraint energy with the iteration number.

This section tests the constrained least-squares fitting algorithm. Figure 4-15 shows the input roof segments and the energy convergence including data energy $\boldsymbol{F}(\boldsymbol{p})$ and constraint energy $\sum_{k=1}^{M} \boldsymbol{C}_k(\boldsymbol{p})^2$ with the iteration number. This building has eight roof segments and eight geometric constraints, including horizontal intersection constraints between surface pairs S_0-S_3, S_1-S_4 and S_5-S_6, equal slope constraints between surface pairs S_0-S_3, S_1-S_4, and S_5-S_6, and concurrent constraints between surface pairs S_0-S_1-S_4-S_3 and S_1-S_5-S_6-S_4. In the constrained least-squares fitting, the energy of geometric constraints iteratively decreases until it reaches acceptable criteria. After every increase of l, the total energy function (Eq. (6)) is minimized. Because of the increased weight of the geometric constraints, this leads to an increase in the data energy term. The X-axis represents the integration number of the all-gradient decent. The red bars are the locations where λ changes. The initial plane parameters are obtained by only considering the input points of roof segments. Therefore the data energy has small value at the start of optimization, and increase with the weight of constraint energy. Table 4-4 shows the roof plane geometries before and after optimization. In Table 4-4, slope ridge means the angle between the ridge line with the horizontal plane, slope difference means the slope difference between the two roof planes, concurrent distance means the average distance between intersection points to all planes. We can see that all constraints are met, e.g. the slope of ridge line Rid(S_0, S_3) is enforced from 2.75 degrees to 0 degrees, and the concurrent distance of intersection point (S_0, S_1, S_4, S_3) is enforced from 0.09m to 0m.

Table 4-4. Roof plane geometries before and after applying the geometric constraints.

Constraint	Slop of ridge (S_0, S_3)	Slop of ridge (S_1, S_4)	Slop of ridge (S_5, S_6)	Slop of ridge (S_0, S_3)	Slop difference (S_1, S_4)	Slop difference (S_5, S_6)	Concurrent Distance (S_0, S_1, S_4, S_3)	Concurrent Distance (S_1, S_4, S_6, S_5)
Without constraints	2.75	4.30	3.61	3.49	0.92	3.39	0.09	0.06
With constraints	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0

4.5.3 Interactive correction

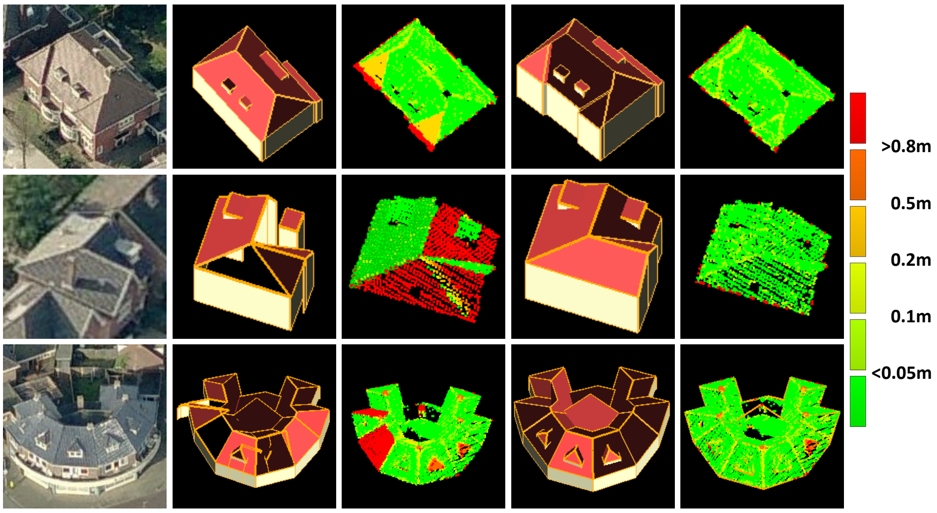


Figure 4-16. Model quality before and after manual input. Column 1: building image; column 2 and 4: building model before and after manual input; column 3 and 5: point clouds coloured by their distance to the corresponding models. The model quality is improved both in model structure and point distance to model.

In order to test the model improvements by the interactive correction, the quality of reconstructed models is evaluated. The point distances to the building models are used to evaluate the model quality. By visualizing the point distances to the building model, the model quality can be clearly observed, and the operator's attention is drawn to the poor models. The mean distance of a 10 points/m^2 density point cloud is typically below 0.2 m to a well reconstructed model, but above 0.2 m to poor models, and sometimes higher than 0.8 m for completely incorrect roofs (see Figure 4-16). The improvements obtained by graph editing are also directly shown in the reconstructed model. The models without manual input miss some roof faces, and the roof structures are wrong. These errors are rectified in the models with manual inputs. Both models show a high error value at the lidar points of outliers and the small objects that do not need to be modelled like chimneys, trees, and air conditioners on the roof. Some lidar points have large distance to the model because they are outside the building contour.

In addition, the operation frequency and time are used to evaluate the efficiency of interactive refinement (See Table 4-5). In the four test areas, about five percent of the buildings needed manual correction. The frequency of interactive correction depends on the data quality and the complexity of building structure. The most often used action is 'Remove Intersection'. In most situations, the operator needs to find where to remove unnecessary intersection lines. The action 'Restore a Segment' is the least used tool, usually less than ten times in one test area, because most roof faces are well detected. According to the complexity of test area, the refinement time changes from three minutes to ten minutes for about three hundred and fifty buildings. The time for the computer's automatic reconstruction is usually less than two minutes for each test area. The total consuming time is acceptable according to the building numbers.

Table 4-5. Operation Efficiency.

	Total building	Refined building	Remove intersection	Restore intersection	Merge segments	Restore a segment	Edit time (minutes)
Area 1	307	14	16	5	4	0	3
Area 2	401	12	72	2	23	0	10
Area 3	345	20	35	9	3	1	7
Area 4	318	16	17	6	8	2	3

4.5.4 Lidar point cloud vs photogrammetric point cloud

The building primitive library introduced in this thesis is based on building structures, and is therefore not restricted to input data types. In this experiment, we compare the influence of input data quality to the quality of the reconstructed model. The input data includes the airborne laser scanning point cloud and the photogrammetric point cloud from dense matching of stereo images.

The Vaihingen dataset is chosen for the evaluation. The Vaihingen dataset is part of a benchmark to compare 3D building reconstruction algorithms (Rottensteiner et al., 2014). It consists of 20 high-resolution DMC images and subsets of Airborne Laser Scanning data used in that test. The Airborne Laser Scanning data used in that test are acquired by a Leica ALS50 system with average 500 m flying height above ground and 45° field of view. The mean strip overlap is 30% and the mean point density is 5 points/m^2. There are three test sites: Area 1 is densely built and contains historic buildings with rather complex shapes. Area 2 contains a few high-rise residential buildings. Area 3 is a residential area with isolated houses. The reference LoD2 building models for building reconstruction have no roof overhangs or façade details, except detailed roof faces. The model accuracy is about 10 cm in planimetry and height. Further details are described in (Rottensteiner et al., 2014).

A dense matching algorithm (Jancosek and Pajdla, 2011) is used to produce photogrammetric point cloud from the calibrated and oriented stereo images.

The method uses visual-hull to reconstruct the difficult surfaces that are weakly supported by images. In order to handle large amount of data, the space is divided into boxes, each is fitted to memory and processed separately. The boxes are cut with overlaps for avoiding conflict between boxes. The method can reconstruct difficult surfaces while preserving noise and improving computation efficiency. The photogrammetric point cloud currently has a higher noise level than the lidar point cloud, and hence provides a less reliable roof topology graph for interpreting roof structures. Errors in the roof topology graph from photogrammetric point cloud are more complex and more diverse than ones from lidar point cloud. As a consequence, the roof topology graphs from the photogrammetric point cloud require more manual correction operations than those from the lidar point cloud.

Table 4-6 shows the model qualities from the two different point clouds. ITCX_G2 is a reconstruction method using lidar data and a graph edit dictionary. IMAGE is a reconstruction method using photogrammetric point cloud and manual correction. From Table 4-6, we can see that the image point cloud provides more complete building data, thus achieving higher completeness. The image pixels have a more regular distribution and the produced photogrammetric point cloud is denser than the lidar point cloud. The correctness is, however, lower because of occlusion and noise. The photogrammetric point clouds have a higher residual than lidar data, so the geometric accuracy of reconstructed models from photogrammetric data is lower than ones from LiDAR data, both in the X-Y plane and in the Z-direction. Although images are known to be more suitable for outlining, this is not evident from our results, because only point clouds from dense matching were used and no edges that could have been extracted from the images. In the photogrammetric point cloud, the walls and roofs are difficult to distinguish, especially when the point cloud is smoothed. Furthermore, points on adjacent trees sometimes also affect the classification and thereby decrease the precision on the X-Y plane.

Table 4-6. The qualities of models from lidar point cloud and photogrammetric point cloud in the Vaihigen dataset.

area	Abbrev.	Compl roof [%]	Corr roof [%]	Compl roof 10 [%]	Corr roof 10 [%]	Topo 1:M	Topo N:1	Topo N:M	RMS [m]	RMSZ [m]
1	ITCX_G2	89.2	96.4	93.2	97.7	5	39	6	0.81	0.23
	IMAGE	94.1	93.2	94.2	98.3	10	33	15	0.69	0.31
2	ITCX_G2	71.0	100.0	89.6	100.0	3	4	1	0.53	0.21
	IMAGE	88.4	95.2	98.8	99.2	13	4	1	0.97	0.54
3	ITCX_G2	88.1	88.2	96.8	95.8	3	50	2	0.69	0.12
	IMAGE	96.2	86.5	99.2	99.9	12	43	15	0.79	0.25

Figure 4-17 provides two examples of buildings showing the quality difference between lidar point cloud and photogrammetric point cloud. Photogrammetric point cloud provides more complete data, e.g. smaller structures in both examples can be seen in the photogrammetric point cloud, and the two flat buildings near to the main building in the bottom example are more complete. The two small hip roofs of the gable building in the bottom example have only several points in the lidar point cloud and could therefore not be detected. Though they could be seen in the photogrammetric point cloud, the noise is too high so that the correct roof structures could not be detected. As a result, an incorrect building model is reconstructed.

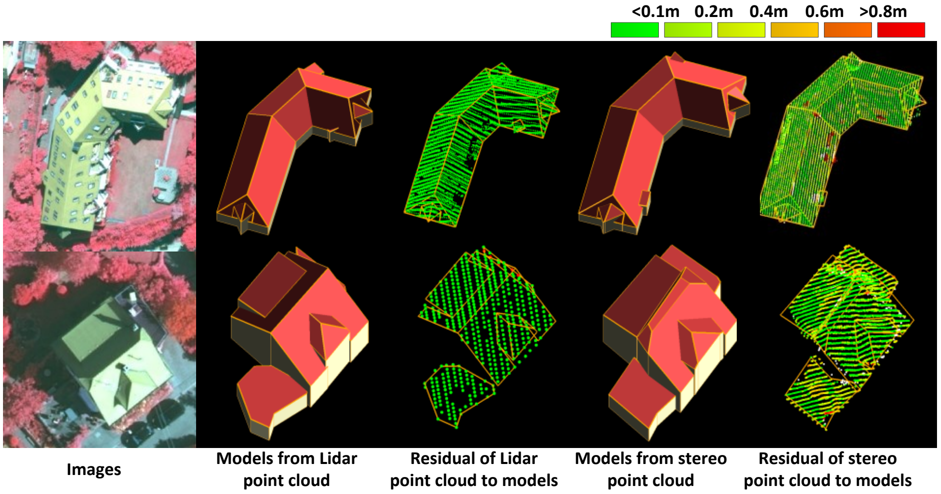

<0.1m
0.2m
0.4m
0.6m
>0.8m
Images
Models from Lidar point cloud
Residual of Lidar point cloud to models
Models from stereo point cloud
Residual of stereo point cloud to models

Figure 4-17. Quality difference of building models from Lidar point cloud and photogrammetric point cloud.

4.6 Conclusion

This chapter presents an algorithm for the reconstruction of 3D building models. The main innovation of this work is the definition of flexible building primitives which enables the representation of any complex building with planar surfaces. The problem of reconstructing complex roofs is reduced to the problem of detecting and reconstructing building primitives. We have demonstrated that a small but effective vocabulary of simple parametric shapes is capable of expressing extremely complex roof structures. The improvements in model quality are experimentally demonstrated by comparing our new building reconstruction method with an older method. This new method is capable of processing both lidar point clouds with good quality and photogrammetric point clouds with a higher level of noise. As there are many errors in the roof topology graphs automatically derived from the photogrammetric point clouds, lots of manual inputs are needed to correct the roof topology graphs.

This chapter have also developed a constraint least-squares fitting (CLSF) algorithm to include building shape constraints that can simultaneously adjust all roof parameters. The experiments show that shape constraints such as multi-plane concurrency and multiple horizontal roofs can be achieved while minimizing the average distance of lidar points to roof planes. Future improvements include adapting the current framework to include non-flat roof shapes such as domes, improving the robustness of our algorithm to decrease errors in the topology graph, and reconstructing entire buildings, including walls, by incorporating ground-based LIDAR.

The roof topology graph can be expanded to include top-graphs of faces of walls, as well as any primitive shapes, like cylinders, cones and squares. The basic elements in those top-graphs are still meaningful. Therefore, the work territory could be expanded to include the modelling of indoor scenes, as well as industrial products. As the roof topology graph is not restricted by data type, this process could also be used to reconstruct objects from stereo images, Kinect points, as well as Terrestrial Lidar points. The proposed method is dependent on the surface segment topology graphs, which are inevitably erroneous. Although an efficient interactive tool is designed to efficiently correct the erroneous topology graph, there is clearly still a need to automatically correct errors in graphs. In the following chapter, an automatic method will be introduced.

Chapter 5

A graph edit dictionary for correcting erroneous roof topology graphs

In the task of 3D building model reconstruction from point clouds we face the problem of recovering a roof topology graph in the presence of noise, small roof faces and low point densities. Errors in roof topology graphs will seriously affect the final modelling results. The aim of this research is to automatically correct these errors. The graph correction is defined as a graph-to-graph problem, similar to the spelling correction problem (also called the string-to-string problem). The graph correction is more complex than string correction, as the graphs are 2D while strings are only 1D. A strategy based on a dictionary of graph edit operations is designed to automatically identify and correct the errors in the input graph. For each type of error the graph edit dictionary stores a representative erroneous sub-graph as well as the corrected version. As an erroneous roof topology graph may contain several errors, a heuristic search is applied to find the optimum sequence of graph edits to correct the errors one by one. The graph edit dictionary can be expanded to include entries needed to cope with errors that were previously not encountered.

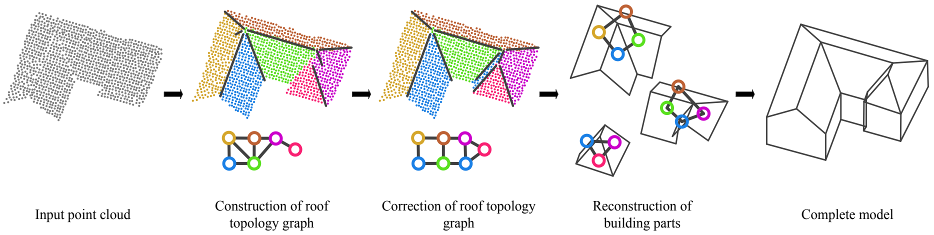

Input point cloud
Construction of roof topology graph
Correction of roof topology graph
Reconstruction of building parts
Complete model

Figure 5-1. Workflow of the building model reconstruction from a point cloud. The core part of the workflow is the automatic correction of roof topology graph. A graph edit dictionary which stores representative erroneous sub-graphs as well as the corrected versions is applied to search and correct the errors in the input roof topology graph.

Chapter 4 introduces a tool to interactively correct errors in roof topology graphs. Although the tool is efficient and easy to use, we still need to automatically correct errors in graphs. The tool has been used to construct the corrected sub-graphs for all entries in the dictionary and to construct ground truth data for evaluating the model qualities. Because dictionary entries typically consist of sub-graphs with multiple errors, making use of the dictionary for error correction is faster than manually correcting all errors. As illustrated in Figure 5-1, after automatically correcting the erroneous graphs, the roof topology graphs are then used to reconstruct building models, as discussed in Chapter 4.

The remainder of this chapter is organised as follows: Section 5.1 will present an overview of the proposed method. The innovative components of this method will be presented in more detail in Section 5.3, Section 5.4, and Section 5.5. Section 5.6 will discuss the results of the method as applied to a dataset of Vaihingen (Germany), and Enschede (the Netherlands). The chapter ends in Section 5.7 with concluding remarks, a discussion of open questions and suggestions for further work.

5.1 Overview

This section describes the main processing flow. The key components are further elaborated in the next sections. The procedure starts with the initial retrieval of roof topology graphs, similar to the methods presented by Verma et al. (2006) and Oude Elberink and Vosselman (2009). In contrast to these approaches, the most simple target graphs are used to obtain the largest possible flexibility in modelling complex buildings. The reconstructed roof topology graphs are the basis for the 3D geometric model of the buildings. After the initial model reconstruction by the method discussed in Chapter 4, it needs to be decided for every building model whether the quality is acceptable. The measures proposed by Rutzinger et al. (2009) are frequently used to describe the final building model quality. However, it is not straightforward to use one of the measures or a combination of measures to make the acceptance decision. Therefore this chapter introduces a simple new measure that will accept a building model, if there are no larger areas in the point cloud that are not close to the reconstructed model. The various quality measures are discussed in Section 5.3.

The main goal of this work is to analyse and improve the building models that did not pass the acceptance test. After analysing reconstruction results of over a thousand buildings, we found that various types of errors frequently reoccur in the reconstructed models. This led to the question whether it would be possible to recognise these types of errors and apply a standard correction to the roof topology graph. From this idea the concept of a graph edit dictionary was developed. The entries of this dictionary are pairs of erroneous sub-graphs of roof topology graphs and the corresponding corrected sub-graphs. By looking up an erroneous sub-graph in the dictionary the translation to the correct sub-graph can be found. To realise such a graph correction strategy various components have been developed: a method to characterise the errors such that the erroneous part of a roof topology graph can be localised, and a method to decide on the best correction option.

An error in a roof topology graph is typically not a single missing or false node or edge. If e.g. a roof segment is not detected in the point cloud, the node of this roof segment is missing in the roof topology graph as well as the edges representing the intersections with adjacent roof faces. Similarly, if a false edge is present in the graph, this will also impact the quality of the nodes on both sides. Consequently, the entries of the graph edit dictionary do not consist of the elementary graph errors only, but of compositions of various elementary errors. In addition to the nodes and edges that are incorrect the dictionary entries also contain a few adjacent correctly reconstructed nodes and edges. These latter nodes and edges provide some further context in which the error happened and are valuable for the identification of the right error type and selection of the right graph edit rule. In order to describe the erroneous sub-graphs in terms of well and poorly reconstructed nodes and edges, we need to be able to evaluate the quality of the individual nodes and edges. For this purpose quality measures were defined (Section 5.3). After the detailed quality analysis of a graph a quality label is assigned to every node and edge. The roof topology graph thereby becomes an attributed graph. The entries of the graph edit dictionary consist of pairs of the attributed erroneous sub-graphs and manually corrected sub-graphs. Section 5.4 further discusses the construction of the dictionary.

Once the graph edit dictionary has been composed it can be applied to the target graphs of building models that did not pass the overall acceptance test. First, the quality of all nodes and edges is evaluated leading to an attributed roof topology graph with one or more erroneous sub-graphs. Sub-graph isomorphism are then determined between the roof topology graph of a rejected model and the erroneous sub-graph entries of the graph edit dictionary. Typically, multiple graph isomorphism will be found as the sub-graphs in the dictionary are relatively simple. To select the best match a distance between the sub-graphs is defined on the basis of the quality attributes of the nodes and edges. Section 5.5 defines this distance measure and further explain how to search for the best graph edit rule.

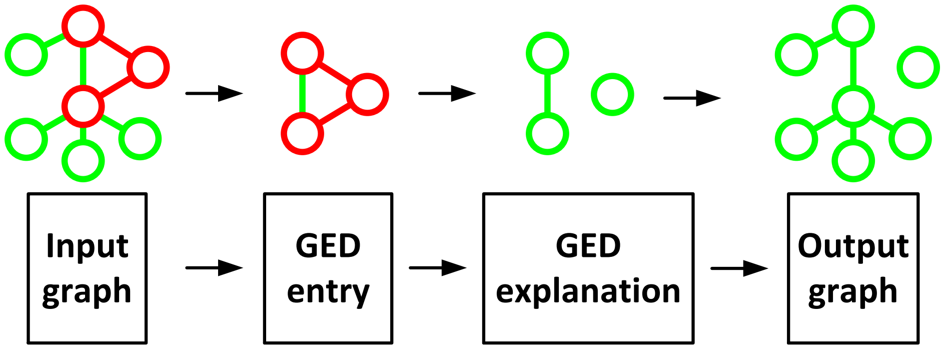


Figure 5-2. Graph correction workflow. The colour of nodes and edges presents their qualities. Red present erroneous nodes and edges, while green means good ones.

Figure 5-2 visualises the use of the graph edit dictionary. A roof topology graph with some errors is matched against the entries of the dictionary. The best sub-graph matching is determined and the corrected sub-graph of the best matching erroneous sub-graph is used for the correction of the input roof topology graph.

Once the best edit rule has been applied to a roof topology graph a new 3D building model is reconstructed. The new nodes and edges are created based on the found edit rules. Their geometries (plane and line equations) are calculated based on the neighbouring surfaces and the local lidar points. The quality of this new model is again evaluated and the editing is accepted if the model quality improved. As many roof topology graphs will contain errors in several sub-graphs, the building model may not be accepted after the first application of an edit rule. In this case the graph correction procedure iterates by again searching for the best match with sub-graphs in the dictionary. This procedure continues until the building model is accepted or until no further edit rules can be found that improve the building model.

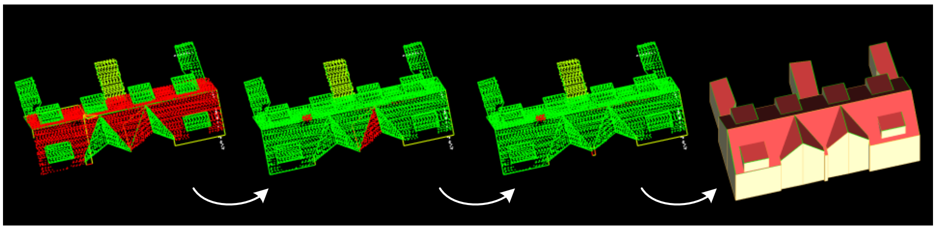


Figure 5-3. An example of iterative improvement of a building model.

Figure 5-3 gives an example of an iteratively improved reconstruction. The first reconstruction contained multiple errors. Various larger segments of points do not match well with the reconstructed model and are shown in red.

After the application of the best editing rule, the model is still not accepted as one side of a dormer window was not modelled well. The use of another edit rule corrects this error. The point cloud shows no larger patches of points that deviate from the reconstructed final building model.

5.2 Overall quality assessment of building models

5.2.1 Currently used evaluation measures

After the reconstruction we need to check the model quality and decide whether the model can be accepted. Measures to evaluate the model quality have been proposed by Rutzinger et al. (2009). These measures were also used in the ISPRS Benchmark on Urban Object Classification and 3D Building Reconstruction (Rottensteiner et al., 2014). The evaluation focuses on the segmentation quality and the geometrical errors of the reconstructed models. The per-roof-plane completeness and correctness of the extracted roof planes are reported on a basis (C_{comp}, C_{corr}). C_{comp10} / C_{corr10} report the completeness and correctness for planes covering an area of at least 10 m^2. The correspondence analysis ($N_{1:M}$ / $N_{N:1}$ / $N_{N:M}$) provides the numbers of instances where 1:M, N:1, and N:M relations between roof planes in the reference and planes in the reconstruction results occur. $N_{1:M}$ is an indicator for over-segmentation, $N_{N:1}$ for under-segmentation and $N_{N:M}$ for clusters of planes that are both over- and under-segmented. The RMS errors of the planimetric distances are derived by comparing the reconstructed boundary points of roof planes to their nearest neighbours on the corresponding reference boundaries. The RMS errors of the height differences (RMSZ) are based on the separation in height between reference planes and the corresponding extracted planes.

5.2.2 A criterion for accepting a reconstructed building model

As elaborated above the measures proposed by Rutzinger et al. (2009) primarily focus on the correctness and completeness of individual roof planes. To decide whether the quality of a building model is acceptable we need a single criterion for the complete building model. This measure should take into account that small deviations of a model from the reality are allowed. These can be either small building parts like chimneys or antennas that do not need to be modelled, or small geometric discrepancies between reconstructed and actual roof planes. Therefore define a poorly reconstructed building model is defined as a model with an area of at least μ m^2 in which the lidar points deviate by more than ν m from the model. Reconstruction failures in areas smaller than μ m^2 are therefore ignored. It is assumed that these failures will not significantly impact the usefulness of the reconstructed model. In the experiments the values are set that $\mu = 3\ m^2$ and $\nu = 0.3$ m.

After a reconstruction the distance to the model is calculated for every lidar point. A connected component analysis is applied to the points that deviate from the model by more than 0.3 m. When the component sizes are below 3 m^2, the building model is considered acceptable. This quality criterion does not make use of reference data and hence may not capture all errors. It serves, however, well as a self-diagnosis for the reconstruction method. Building models that do not pass the quality test will be further analysed and improved with the graph edit dictionary. In the quality analysis of building reconstruction in larger datasets presented in section 5, the measure e_b is proposed to denote the percentage of building models that have been rejected by the above criterion. Next to this measure the evaluation measures developed by Rutzinger et al. (2009) is also presented.

5.3 Local quality assessment in roof topology graphs

In order to locate the elementary errors in the roof topology graph, we need to evaluate the quality of the nodes and edges. These qualities are also used as attributes in the roof topology graph for the detection of erroneous sub-graphs. In our work, the supports by nearby points are the only hints for the quality of nodes and edges. Therefore, two measures based on the distributions of these points are defined as node and edge attributes.

5.3.1 Node Attributes

The node attributes should reflect the probability that a roof polygon is well reconstructed. In case of a good reconstruction the distances of the roof segment points to the roof polygon will show a normal distribution. We use G-tests, a maximum likelihood statistical significance test, to test whether the remaining point distances show the expected normal distribution.

$$G = 2 \cdot \sum_i O_i \cdot \ln(O_i / E_i) \qquad (8.1)$$

Here O_i is the observed frequency in a bin of the point distance histogram, E_i is the expected frequency under the null hypothesis of a correct reconstruction. The sum is taken over all non-empty bins.

The distance of a point to the roof plane is defined as the orthogonal distance to roof polygons, see Figure 5-4. If the projection of a point is located inside the polygon, the orthogonal distance is the orthogonal distance to the polygon plane. Otherwise the orthogonal distance is the orthogonal distance to the nearest boundary line of the polygon. Similarly, the nearest distance to the line segment is the orthogonal distance to the line if the projection lies inside the line segment. Otherwise, it is the distance to the nearest end point. Incorrect roof polygons caused by errors in the roof topology graph typically have many points that are far away from the polygon and will therefore receive a low value of the quality attribute.

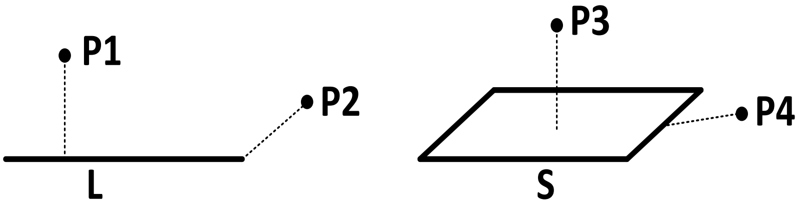


Figure 5-4. Orthogonal distance of a point to the line segment and polygons. Left: orthogonal distance to a line segment. Right: orthogonal distance to a polygon.

The G-test is a good alternative to the well-known χ^2-test when the expected value E_i is small. The distribution of G is approximately χ^2, with the same number of degrees of freedom as in the corresponding χ^2-test. We choose G-tests because the expected frequency is very small for residuals larger than 0.3 m when the standard deviation of point distance is small, typically 0.03m. The quality attribute of a node in a roof topology graph is defined as p-value for the computed G-test value and degree-of-freedom. The p-value is the probability of observing a test statistic at least as extreme in a χ^2 distribution. It expresses the goodness-of-fit of the observed frequencies to the expected normal distribution. The expected mean is always 0, and the standard deviation is computed for each roof segment, because the roof material and with it the standard deviation may vary. As we know the orthogonal distance should be less than 0.3 m, only the points within 0.3 m are used to compute the standard deviation. The node quality χ^2 is defined by

$$\mu = p(G,\lambda) = 1 - chi\_cdf(G,\lambda) \tag{8.2}$$

$$chi\_cdf(G,\lambda) = \int_0^G \frac{t^{(\lambda-2)/2} e^{-t/2}}{2^{\lambda/2}\Gamma(\lambda/2)} dt \tag{8.3}$$

Where p is the p-value, chi_cdf is the cumulative distribution function of χ^2 distribution, λ is the degree of freedom, and $\Gamma(\cdot)$ is the Gamma function.

The degree of freedom is essential for the χ^2 test. The degree of freedom is the number of independent frequencies reduced by the number of parameters of the fitted distribution. In our implementation the histograms of point distances always have 50 bins, and the number of parameters of the fitted distribution is 2 (mean and standard deviation of the Gaussian). Then the degree of freedom is 47 (50-1-2). For a 95% confidence level this results in a critical value of 64.0. In experiments it is, however, noted that most correct roof polygons had G values below 5 and that many incorrect roof polygons had G values below the critical value. It is experimentally determined that a degree of freedom of 2 led to a much better characterisation of the quality of the reconstructed nodes. The thus defined statistical measure is added as a quality attribute to all points.

5.3.2 Edge Attributes

The edges of the roof topology graphs have two types of attributes: line type and edge gradient. For the line type, we only consider horizontal and sloped lines.

$$L \in \{Horizontal, Sloped\} \tag{8.4}$$

More line types, like convex, concave attributes, could also be applied. However, the two types are enough for the application of the graph edit dictionary in our tests.

The edge gradient evaluates the point distribution along the ridge lines. The edge gradient is similar to image line gradient, but is adjusted to discrete point set. Unlike image pixels, the discrete points do not distribute uniformly. Consequently, the edge gradient is calculated based on the equally sized cells of the buffer zones along the line segment. The point distribution is described more precisely with these cells. In Figure 5-5(a), the points distribute uniformly, therefore splitting the edges into bins does not have an effect on the edge gradient. However, when the point distribute irregularly, the difference will be large.

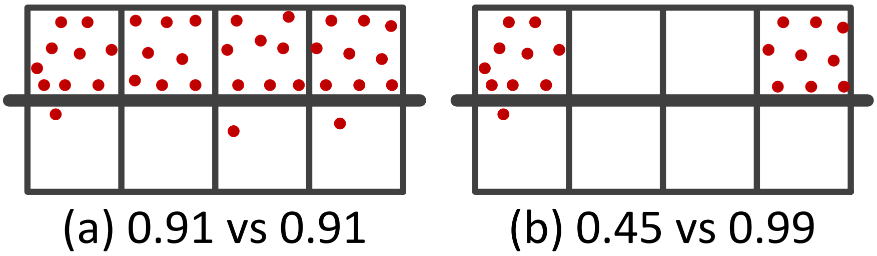


Figure 5-5. Edge gradient difference of splitting the edge into bins or not. First number in each case is the edge gradient with splitting, and second number is the one without splitting.

The edge gradient is based on the ratio of the number of points of one segment on both sides of the edge (Eqs. (8.5)-(8.7)). Figure 5-5 illustrates how the edge gradient is calculated and the gradient change when an edge line transforms along the orthogonal direction of the edge line on the roof plane. The gradient is calculated on each of the two supporting roof segments (G_l, G_r), and averaged. On each segment, the buffer zone of the edge is split into two sides and N bins. A bin pair is a set of two bins, one on either side of the intersection line. Each bin pair has a point ratio, and all point ratios are averaged together, subtracting the mean ratio from 1 gives the edge gradient. The gradient of each bin pair is the ratio of larger number of points to the smaller number of points in the two bins. Therefore the ratio is in the range of 0 to 1. If both bins of one bine pair are empty, the point ratio is defined to be 1.

$$R = \frac{Min(Bin_1, Bin_2)}{Max(Bin_1, Bin_2)} \tag{8.5}$$

$$\mathrm{G} = 1 - \frac{1}{\mathrm{N}} \sum_{i=1}^{N} R_i \tag{8.6}$$

$$G = \left(G_l + G_r\right) / 2 \tag{8.7}$$

The edge gradient, with interval [0, 1], presents the confidence of an edge. The higher the edge gradient becomes, the more likely an edge is present. An ideal edge has the edge gradient 1, because it has many points on one side and no points on the other side, when only considering one supporting roof segment. When a model line is located at the real boundary, it reaches the maximum edge gradient. Figure 5-6 shows the edge gradient of all model lines. Good lines are shown as green or grass yellow, while incorrect lines are tagged as red or black. The edge gradient can be used to evaluate the model quality and guide the reconstruction process.

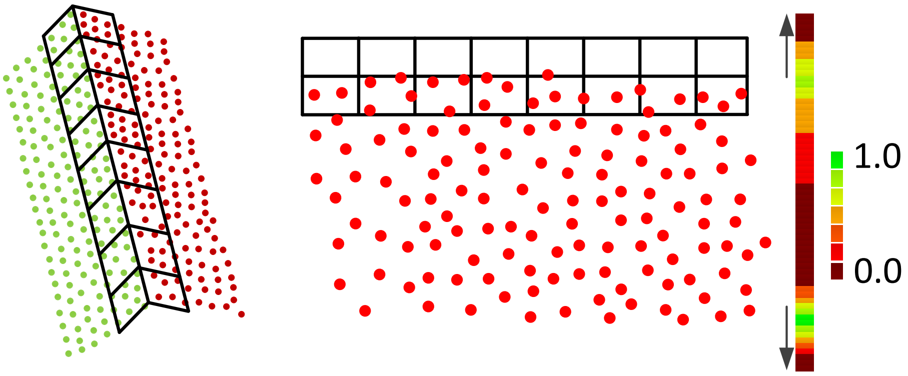


Figure 5-6. Example of edge gradient. Left: the gradient of one intersection line is computed on both roof segments. Right: the gradient on one segment, and the gradient changes if the line position shifts along the orthogonal direction of the intersection line.

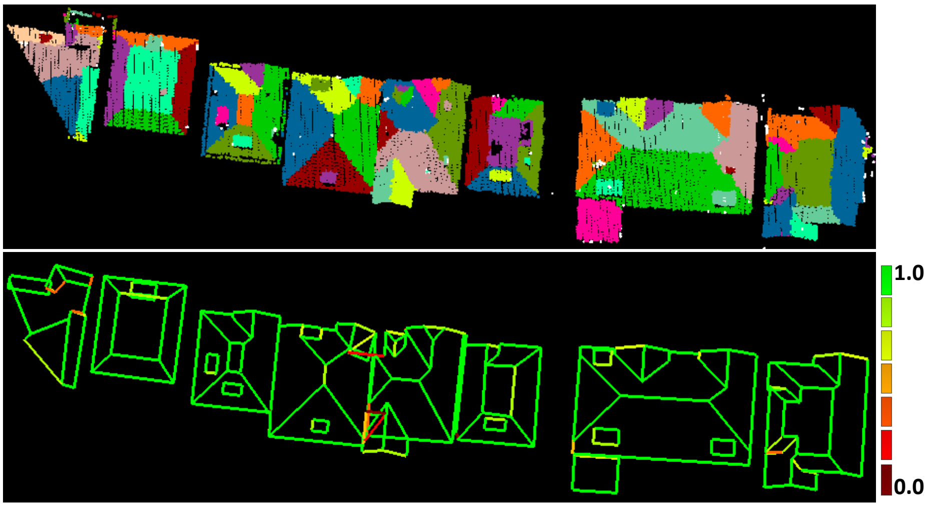


Figure 5-7. Edge gradients of model lines. Up: input lidar points. Bottom: line gradients.

5.4 Graph edit dictionary

The principle of spelling correction is used to correct errors in the topology graph. There are, however, a few differences between the spelling and graph corrections. Words can be considered as one-dimensional graphs, whereas roof topology graphs are two-dimensional. The creation of a full dictionary of building topology graph, as a dictionary of English words, will be infeasible.

Unlike spelling errors, errors in roof topology graphs usually occur in groups. A group of errors can be represented by a single erroneous sub-graph and be used as entry of a graph edit dictionary. In analogy to the 80-20 law we aim to correct a large percentage of the errors by creating editing rules for a small number of most common erroneous sub-graphs (Figure 5-8). It is accepted to leave the rare errors for interactive processing by a human operator. The dictionary can expand to include new error types when further training data becomes available. Therefore, it is adaptable to specific situations (e.g. architectural styles). The construction and expansion of graph edit dictionary (GED) will be discussed in the next sections.

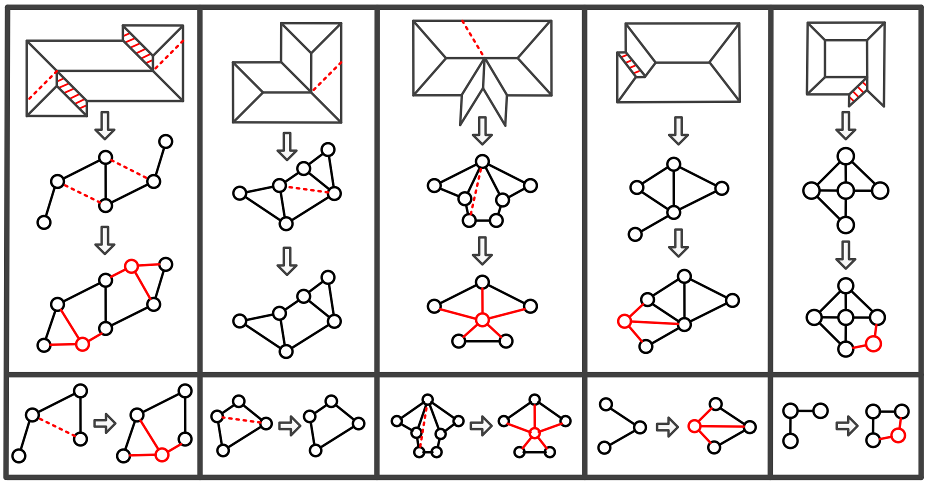


Figure 5-8. Typical error correction. Top: typical errors and correction methods. Bottom: entries in the graph edit dictionary. The red dashed lines are false intersection lines in the building model and false edges in the top-graph. The red shadowed polygons in the building models are undetected roof faces. The red circles and lines represent new generated nodes and edges.

5.4.1 GED construction

The structure and attributes of the erroneous and corrected sub-graphs are automatically learned from a training data set. The training strategy is aiming at automatically constructing and updating the graph edit dictionary according to provided training data set. The dictionary can be updated to better identify stored erroneous graphs or to correct new error types. The users only need to provide a new training data set. Maximum overlaps of training graphs are identified by graph isomorphism, which is discussed later. The greatest common sub-graph of all training graphs for one error type is then used as the entry graph. The attributes of nodes and edges are averaged over all training graphs. The entry graph can be seen as the cluster centre of the set of error graphs of the same type. The distance between two graphs is evaluated by the graph edit distance (discussed in section 5.5) and depends on the number of nodes and edges, as well as their attributes. The distance between two clusters determines the classification sensitivity. As larger graphs are easier to classify we use the greatest common divisor of training graphs as the entry graph. In this way the entry graphs not only contains nodes and edges with a low quality, but also the surrounding nodes and edges with a high quality. The latter nodes and edges serve to understand the context of the error and improve the error classification.

The interpretation graphs for each entry graph are the possible correction ways (corrected sub-graphs). In the training data, an interpretation graph is manually assigned to an erroneous graph. Then in the training period, the interpretation graphs are learned in the same way as the entry graphs. As one erroneous block may have several correction ways, the entry graph may also have several interpretation graphs. But in most cases, one entry graph only has one interpretation graph in our tests.

5.4.2 GED expansion

While working with the graph edit dictionary, the dictionary can be improved in two ways. When an entry of an existing error type is correctly applied to new data, the averaged node and edge attributes of the erroneous sub-graph can be updated. In the case of encountered new error types, the operator can manually correct the roof topology graph and insert the pair of erroneous and corrected graph as a new entry into the dictionary. In this way dictionary gradually grows and can be trained to specific application areas.

5.5 Searching the best graph edits

5.5.1 Graph matching

The graph matching is implemented by recursively traversing a tree, which enumerates all possible transformations between graphs X and Y (Fig. 11). The costs of each transformation are determined by an edit distance measure, which will be defined further on. The transformation with the minimum edit distance is chosen as the best match. More advanced tree search methods are not required as both the graphs in the dictionary and the data graphs are usually small, less than one hundred nodes. The graph matching is used for both the training and use of graph edit dictionary. In the training phase the graph matching is applied to find error blocks of the same type. In the correction phase the graph matching is used to search for potential sub-graphs in the edit dictionary.

The similarity of two graphs is determined by a graph edit distance, which measures the editing effort to transform one graph into the other. Classically, the graph edit distance is defined as the number of required operations for deleting and inserting nodes and edges. For looking up erroneous graphs in the dictionary, we require a one-to-one correspondence between the nodes, but not for the edges. Hence costs only incur for missing or extra edges. In addition to these costs, we also include costs for differences in the attributes of the nodes and edges defined in section 5.3. The edit distance between graphs X and Y is defined as:

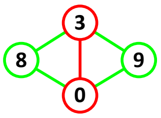
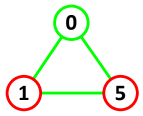
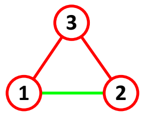
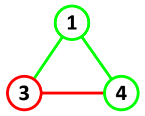
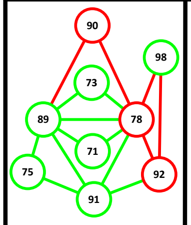
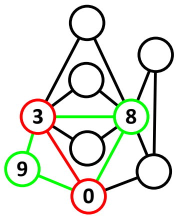
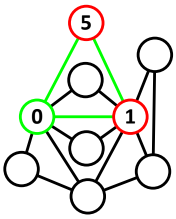
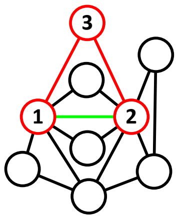
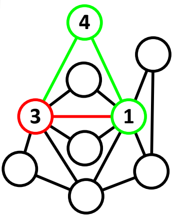
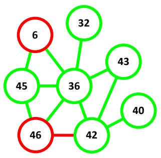
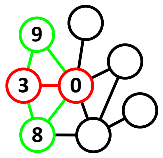
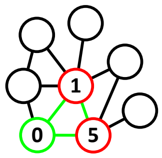
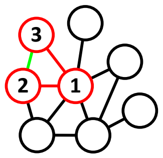
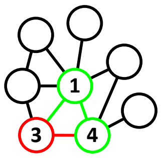
$$W(X,Y)=\sum_{i}\left|\mu_{X_i}-\mu_{Y_i}\right|+\sum_{ij}\left|G_{X_{ij}}-G_{Y_{ij}}\right|+\kappa_1\delta_1\left(L_{X_{ij}},L_{Y_{ij}}\right)+\kappa_2\delta_2\left(L_{X_{ij}},L_{Y_{ij}}\right) \quad (8.8)$$

Where the first sum captures the differences in the quality of the nodes and the remainder measures the differences in the edges. The three parts of the latter sum describe the differences in the edge gradient values G, costs for different edge labels where δ_1 evaluates to 1 if the labels are different and 0 if the labels are the same, and costs for missing edges where $\delta_2$evaluates to 1 if an edge is missing on one of the two graphs and 0 when corresponding edges are present. In experiments weights $k_1 = 2$ for differences in edge slope and a high $k_2 = 10$ for missing edges. These values were determined experimentally.

5.5.2 Looking up in the GED

By comparing all entries in the GED, the entry best matching the input graph as defined by the graph edit distance will be determined. The nodes of sub-graph are exactly corresponding to the nodes of entry graph. In this way, the error sub-graphs in the input graph are recognised. In Table 5-1 two input graph are looked up in a graph edit dictionary with four entries. The first input graph best matches with the third entry. Nodes 90, 78 and 89 in first input graph are matched with nodes 3, 1 and 2 in third entry graph.

Table 5-1. Looking up in the graph edit dictionary. The colours present the attributes of nodes and edges. The number in the circle is the node number, and the number below the graph is the edit distance between the sub-graphs of the input graph and the entry graphs. The edit distance in green is the minimum edit distance for matching all entries.

Entries Match & Distance Inputs				
	 0.873	 0.913	 0.446	 0.573
	 1.293	 0.341	 0.431	 0.010

5.5.3 Heuristic searching of shortest path

The problem of correcting an erroneous graph is a graph-to-graph problem. By analysing thousands of buildings, we found repetitive erroneous sub-graphs and their corrections, which are taken as the basic corrections. Hence, a roof topology graph can be corrected by sequentially applying various basic corrections. Each correction can be formulated as a transition rule:

$$\mathrm{G}_i = \mathrm{t}_k\left(G_{i-1}\right) \tag{8.9}$$

Where G_{i-1} is the previous graph, G_i is the new and improved graph, and t_k is a basic transition taken from the graph edit dictionary.

One transition path is composed by a sequence of several basic transitions:

$$T_k = \left\{\mathrm{t}_{k_1}, \mathrm{t}_{k_2}, \cdots, \mathrm{t}_{k_n}\right\} \tag{8.10}$$

The graph correction problem can be represented as an energy optimization problem (Eq. (8.11)): the target topology graph is the one which minimizes the model energy. The erroneous graph is at a high-energy state. After releasing energy in each transition, the graph gradually arrives at a stable state with a minimum energy. Hence, we are seeking the sequence of edit operations Ti leading to the best graph:

$$G_{output} := \underset{T_i \in \{T\}}{\arg\min}\, E\left(T_i\left(G_{input}\right)\right) \tag{8.11}$$

Where $\mathrm{G}_{\mathrm{input}}$ and $\mathrm{G}_{\mathrm{output}}$ are the graph before and after correction and E is the energy defined as the sum of one minus the quality of each of the nodes and edges of the graph.

The transition may have many alternative status goals and many possible paths. The objective is to search the best correction path, which needs the fewest basic transactions, from the set of all possible transaction paths. The best-first search strategy is used to search the shortest path. In each iteration step, the correction with the maximum energy release is taken. This is done by searching the most similar error pattern stored in the dictionary. The correction stops when no erroneous sub-graph could be found. The advantage of this heuristic search is that it can find a good solution in reasonable time, though not always the best one. When correcting one sub-graph, some other nodes and edges also change their states. For example, the segment attribute is determined by all cycles neighbouring to the node. If one cycle changes, the attribute will also change. Each step corrects one error block, which is only influenced by its members, and then the next erroneous sub-graph will appear when the last one is corrected.

5.6 Experiments

In the experiments we want to test the ability of the dictionary to correct roof topology graphs, the robustness of entries with the training data set, and the expandability of dictionary. We use the evaluation methods described in (Rutzinger et al., 2009) and the measure based on the point-to-model distances as described in Section 5.2.2 to evaluate the model qualities. Two data sets were used for the experiments.

5.6.1 Vaihingen (Germany)

The Vaihingen dataset is part of a benchmark to compare 3D building reconstruction algorithms (Rottensteiner et al., 2014). The Airborne Lidar Scanning data used in that test are acquired by a Leica ALS50 system with average 500 m flying height above ground and 45° field of view. The mean strip overlap is 30% and the mean point density is 5 points/m^2. There are three test sites: Area 1 is densely built and contains historic buildings with rather complex shapes. Area 2 contains a few high-rise residential buildings. Area 3 is a residential area with isolated houses. The reference LoD2 building models for building reconstruction have no roof overhangs or façade details, except detailed roof faces. The model accuracy is about 10 cm in planimetry and height. Further details are described in (Rottensteiner et al., 2014).

Table 5-2. Evaluation of the building reconstruction results in Vaihingen.

Name	N_b	GED	e_b [%]	C_{comp} [%]	C_{corr} [%]	C_{comp10} [%]	C_{corr10} [%]	$N_{1:M}/N_{N:1}/N_{N:M}$	RMS [m]	RMSZ [m]
area1	38	No	18.4	84.7	96.2	87.8	99.2	13 / 41 / 16	0.46	0.24
		Yes	7.9	89.2	96.4	93.2	97.7	5 / 39 / 6	0.42	0.23
area2	15	No	20.0	88.9	99.1	91.7	100.0	15 / 3 / 1	0.72	0.36
		Yes	13.3	71.0	100.0	89.6	100.0	3 / 4 / 1	0.49	0.21
area3	57	No	10.5	86.0	84.4	93.5	95.0	5 / 48 / 2	0.32	0.15
		Yes	3.5	88.1	88.2	96.8	95.8	3 / 50 / 2	0.32	0.15

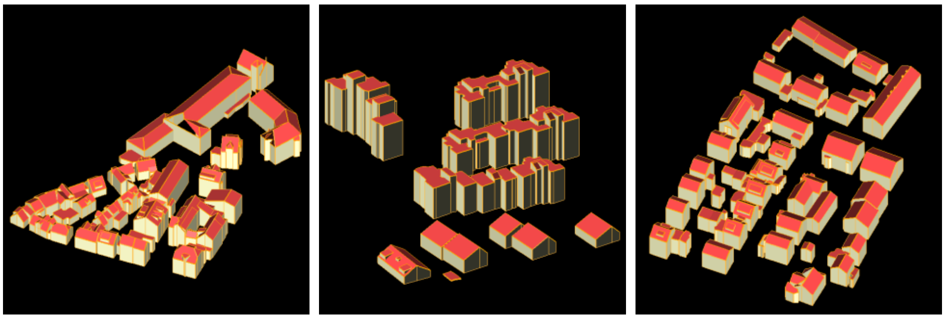


Figure 5-9. Reconstructed models of Vaihingen test areas. From left to right: area 1, area 2 and area 3.

The lidar data is classified and segmented into roof segments. As this reference dataset only has about 100 buildings, and only a few building models need to be corrected, there are not enough samples to test the influence of the number of entries in the graph edit dictionary and amount of the training samples. Therefore we only test the correction ability of the graph edit dictionary. Table 5-2 shows the quality of models before and after graph correction. Nb is the number of buildings in the area. The GED column shows whether or not the graph edit dictionary was applied. The measure e_b is the percentage of erroneous building models as defined in Section 5.2.2 with a maximum size of 3 m^2 of points further than 0.3 m away from the model. The measures C_{comp}, C_{corr}, C_{comp10}, C_{corr10}, $N_{1:M}$, $N_{N:1}$, $N_{N:M}$, RMS, and RMSZ were all introduced in Section 5.2.1. Most measures show the models are improved.

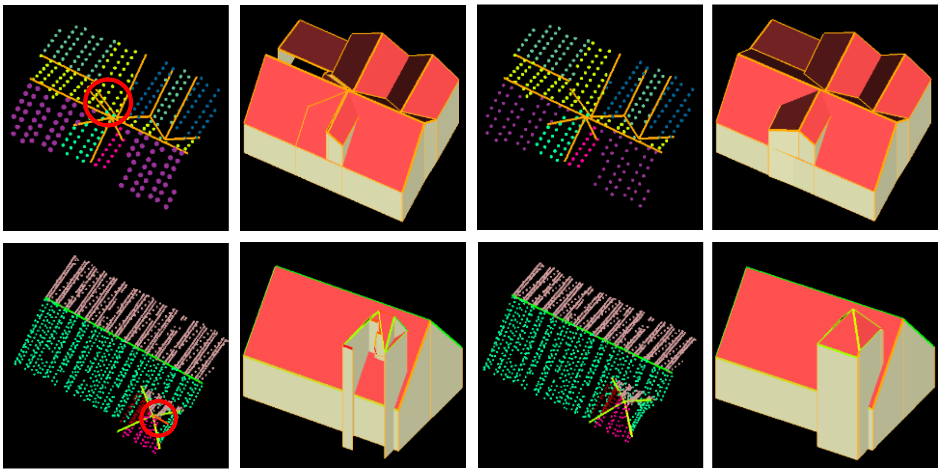


Figure 5-10. Two often wrong reconstructed buildings in the Vaihingen dataset. 1) Roof segments and ridges without correction;

2) Reconstructed models from 1); 3) Roof segments and ridges with correction; 4) Reconstructed models from 3).

Figure 5-9 shows the reconstructed models after the graph correction. Models in area 1 are improved most obviously because their complex shapes often result in topology graph errors. One building could not be corrected because the sparse point density results in a lack of data on a major part of the building. Models in area 2 are hardly improved because the large number of small point segments on high rising buildings strongly affect the roof topology graph and are hard to interpret. In this area the completeness for all roof faces and the correctness for roof faces with more 10 m^2 show the model quality was reduced. The correction algorithm sometimes incorrectly detects errors and then deteriorates the model. Figure 5-10 illustrates some model improvements in area 1. The two buildings are incorrectly reconstructed by most methodologies participating in the ISPRS test (Rottensteiner et al., 2014). The main reason for this is that the low point density (about 5 point/m^2) requires a low threshold to detect ridge lines. Hence, false ridge lines are likely to be created. With a higher threshold many ridge lines would be missed. The experiments show that the wrong roof topology graphs were correctly detected and corrected by the proposed GED, resulting in correct models.

5.6.2 Enschede (Netherlands)

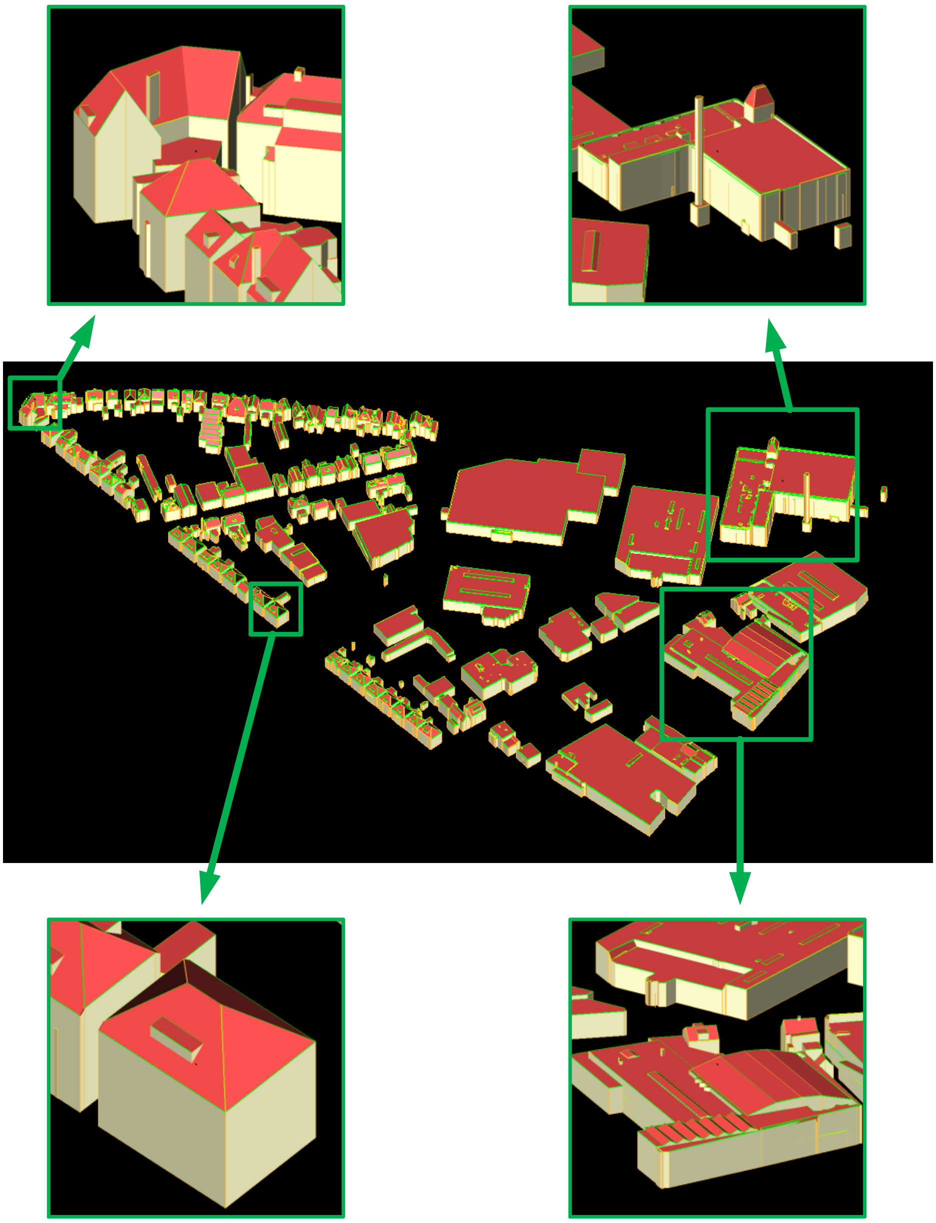


Figure 5-11. Reconstructed models of one area in Enschede.

A dataset of Enschede is selected to fully test the graph edit dictionary because the number of buildings in the ISPRS benchmark are not enough to evaluate the performance of the graph edit dictionary. The city is characterized by residential buildings with diverse structures in the outskirts and a small part of commercial buildings in downtown. The laser scanning data has been acquired by the FLI-MAP system of Fugro Aerial Mapping B.V at a flight height of 275 m and a 30% strip overlap. The point density is on average about 20 points/m^2. Figure 5-11 shows the reconstructed models for one area of the dataset.

Performance of dictionary correction

Table 5-3. Evaluation of the building reconstruction results in Enschede.

Area	N_b	GED	e_b [%]	RMS [m]	RMSZ[m]	$N_{1:M}$ /$N_{N:1}$ /$N_{N:M}$
De_Laares	751	No	5.3	0.16	0.13	14 / 3 / 9
		Yes	3.7	0.15	0.11	15 / 7/ 9
Gerfert_2	409	No	5.6	0.13	0.13	3 / 2 / 3
		Yes	3.2	0.11	0.05	2 / 1/ 0
Gerfert_3	233	No	9.4	0.40	0.24	11 / 5 / 6
		Yes	8.2	0.39	0.24	10 /5 / 5
Gerfert_4	418	No	9.1	0.32	0.29	15 / 1 / 20
		Yes	6.7	0.26	0.24	10 / 3 / 12
Gerfert_5	235	No	7.7	0.32	0.24	18 / 1 / 6
		Yes	6.8	0.32	0.24	16 / 7 / 6
Gerfert_w_1	427	No	7.5	0.24	0.14	18 / 9 / 9
		Yes	5.9	0.22	0.13	15 / 5 / 7
Gerfert_w_2	472	No	6.4	0.14	0.09	18 / 9 / 9
		Yes	5.1	0.13	0.09	15 / 5 / 7
Gerfert_w_3	682	No	9.8	0.18	0.15	35 / 4 9
		Yes	7.3	0.17	0.14	21 / 2 / 9
Hogeland_n_1	431	No	4.4	0.35	0.2	50 / 9 / 17
		Yes	3.5	0.33	0.11	46 / 9 / 12
Hogeland_n_2	433	No	6.5	0.22	0.13	78 / 10 / 42
		Yes	4.4	0.22	0.12	66 / 12 / 38

We selected 10 test sites with a total of about 4500 buildings for the performance test of graph edit dictionary because of their diverse building structures. For each area, we compare the model quality before and after the graph correction. The full dictionary has 15 entries. Table 5-3 shows that the initial reconstruction already had an acceptance rate of 93%. The graph edit dictionary corrected around 25% of the erroneous buildings, leaving only 5% of the buildings in total which are not accepted. The correction rates vary from 5% to 45% because the building types and complexities are different in the 10 test sites. The errors in some test sites are well learned in the graph edit dictionary, while less learned in others. The correction rates are also influenced by the permissible size of an area with a poor reconstruction result as defined in section 4.2.2. Figure 5-12 shows that the relative correction rates for all test sites increase with the area size threshold. This implies that our method performs better in correcting the larger errors. Small errors, e.g. caused by chimneys and other small constructions on roofs, are more difficult to correct automatically. In the next experiment, we test the performance of dictionary with different entries.

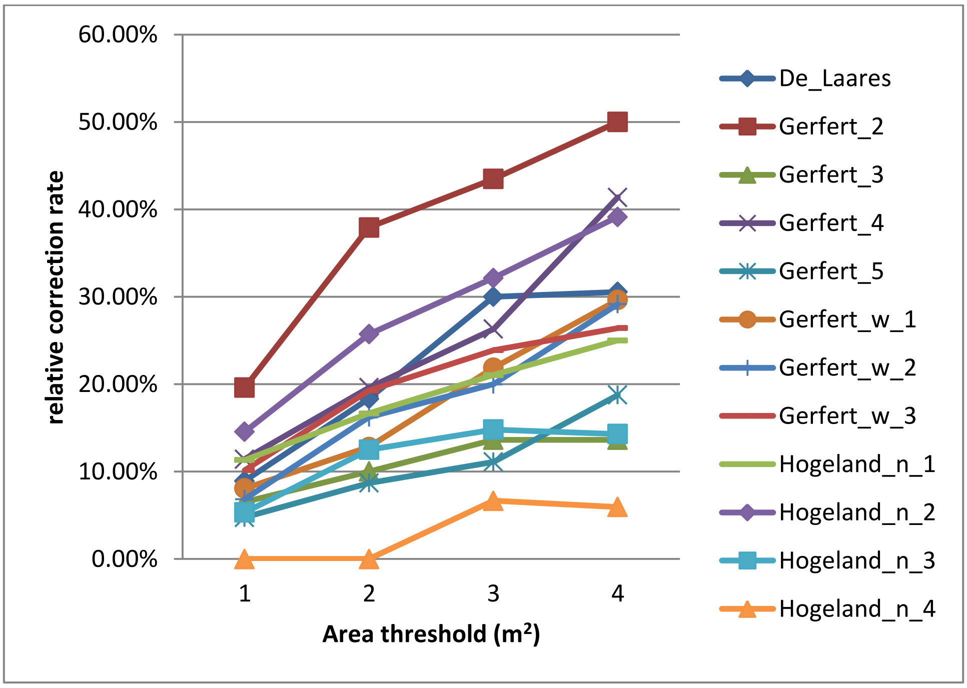


Figure 5-12. Correction rate as a function of the acceptable size of an area with a poor reconstruction.

Performance of expanded dictionary

In this experiment, we test the performance of graph edit dictionaries with different numbers of entries. We assume the dictionary could correct more building error types by expanding it. Four areas, Gerfert_2, Gerfert_3, Gerfert_4, and Gerfert_5, are selected for the test. In each area, we compared the reconstruction results after correction by the graph edit dictionary with different entry sets. The full dictionary has 15 entries. We choose the most frequent one entry and two entries of the full dictionary for the two new dictionaries. The dictionary with zero entry means the models are not corrected. Results are shown in Figure 5-13. As expected the model quality improves with the dictionary size. Therefore, when adding more suitable entries, the correction ability of GED increases although the first few entries will have the largest impact.

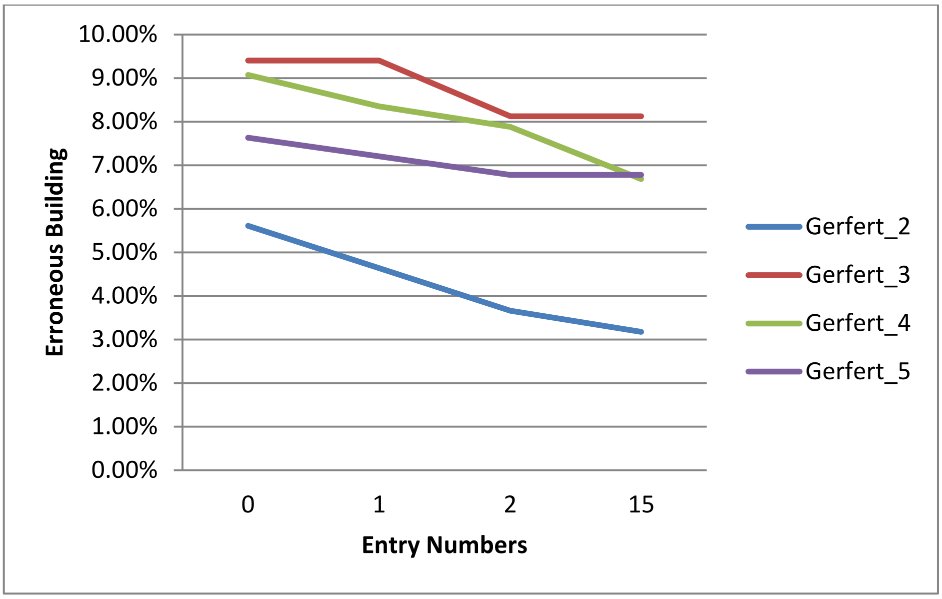


Figure 5-13. Effects of entry numbers in the GED. Gerfert_2, Gerfert_3, Gerfert_4, and Gerfert_5 are the test areas.

Effects of training samples

In this experiment, we test the performance of the dictionary entry with different training data. As the attributes of the entry graphs are the average attributes of graphs in the training dataset, the representativeness of training dataset determines the ability of entry to match with a graph containing the same error. We select one of the most frequent errors in the Enschede test area, and manually collect the lidar points of buildings with this error. The test area has 24 buildings of this type in total. In each group, a part of them are randomly chosen and used as training data, and the rest of them as test data. E.g., when 1 sample is taken as training data, the remaining 23 buildings are used as test data. One building with this error is shown in Figure 5-14. For simplification, only the chimney and its supporting roof part are shown. In the Netherlands, many resident buildings have chimneys of less than 0.5 meters in height. The laser segment of the chimney is often incorrectly detected to be intersecting with the supporting roof. Therefore, two wrong ridges will be detected. In the corrected roof topology graph the corresponding two wrong edges, as well as node of the chimney roof are deleted. Figure 5-15 shows the detection rates as a function of the number of training samples. The detection rate clearly increases with the number of training samples. With 7 or more training samples, the algorithm detects all buildings in the test dataset.

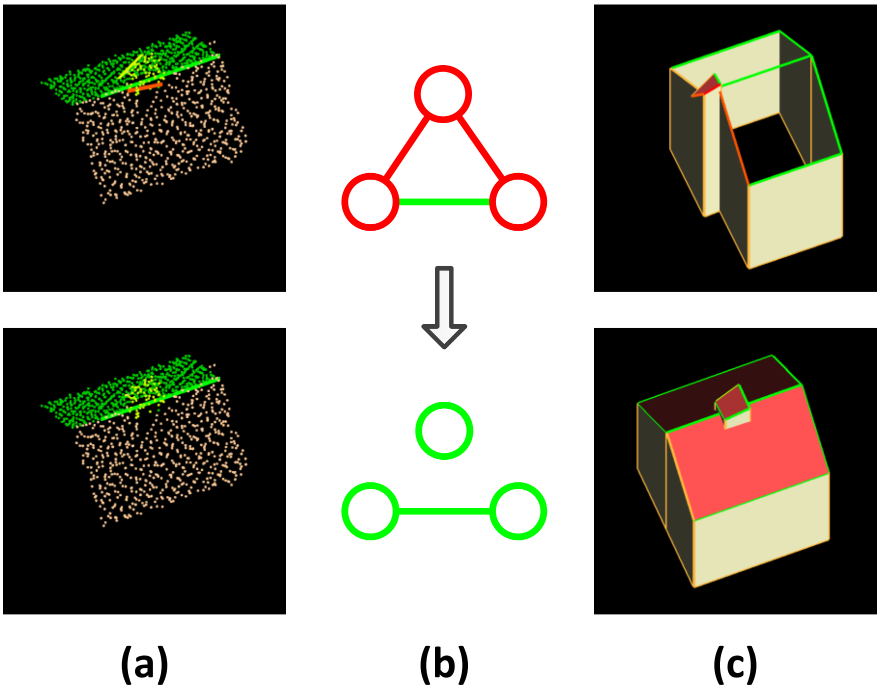


Figure 5-14. One frequently incorrectly constructed building. a) Roof segments and ridges; b) Roof topology graph; c) Reconstructed model. Up: without correction; Bottom: with correction.

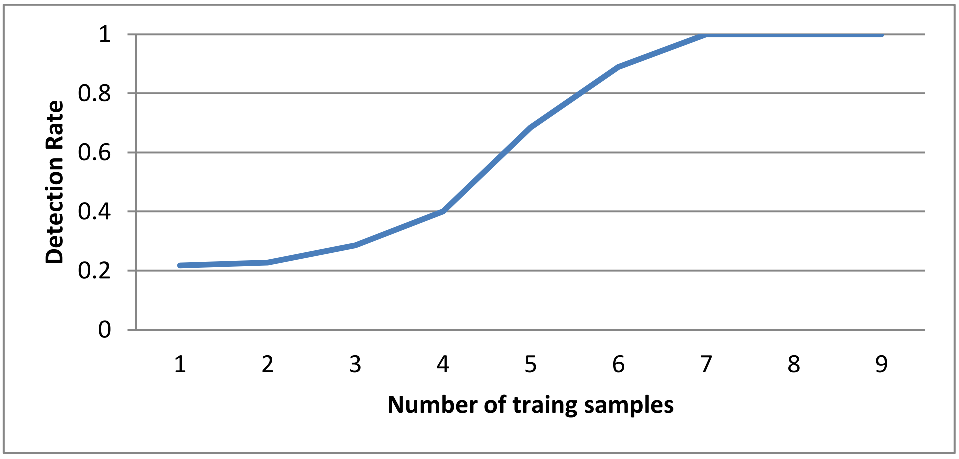


Figure 5-15. Detection rates as a function of the number of training samples. When the number of training samples reaches 7, all remaining 17 buildings are correctly detected.

5.6.3 Problems not solved with the graph edit dictionary

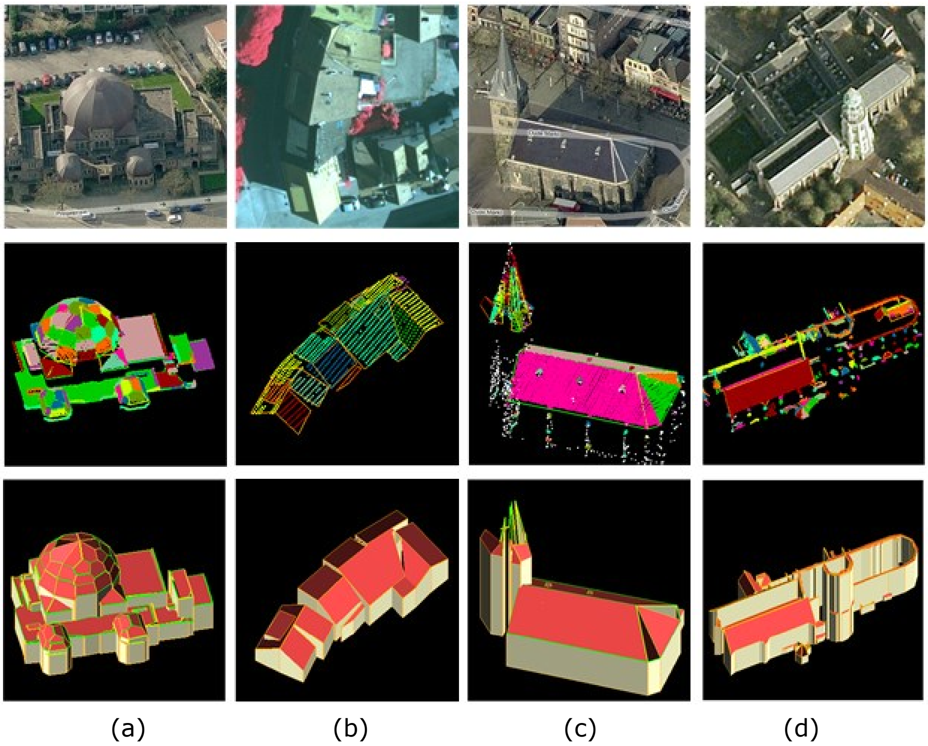


Figure 5-16. Buildings could not be well reconstructed now. First row: images; second row: lidar segments; third row: reconstructed models. (a) free form surfaces; (b) tiny roof pieces; (c) steep roofs; (d) light absorbing surfaces.

Figure 5-16 shows some buildings that could not be reconstructed well. The main problems are caused by the complex shape of the buildings and the lack of data. Our algorithm could approximate some smooth curved surfaces, like cylinders, spheres, and cones, by a group of planar surfaces. Figure 5-16 (a) shows that a building with ellipsoidal surfaces is approximately modelled, but many buildings with freeform surface could not be modelled by planar surfaces. Though in theory any surface can be represented by a set of planar surfaces, the limited point density restricts the detection of such small planar segments. Another problem is that our method can only recover topology graphs when the larger majority of a building is well captured. Figure 5-16 (b) shows a building with very small roof faces that are not well captured by the laser scanner. This resulting misdetection of these surfaces seriously affects the whole model as these surfaces play an important role in the whole roof topology graph, unlike a chimney which could be ignored. Some steep roofs, shown in Figure 5-16 (c), although captured, are not well extracted by surface growing algorithm. The church shown in column (d) of Figure 5-16 is not well recorded because of its light absorbing surfaces. This kind of incorrectly reconstructed building models constitutes around 5% of our dataset of reconstructed models.

5.7 Conclusion

This paper presents a graph edit dictionary to detect and correct errors in roof topology graphs. The graph edit dictionary with 15 entries on the average corrects 25% of the erroneous building models in our test areas rising to about 50% for some test areas. It increases the successful reconstruction rate to about 95% in the Enschede test dataset. The dictionary is expandable towards error types which have not yet been considered. The training data for each entry is as well crucial. About 5 training samples are usually enough to train an entry.

Still about 5% buildings could not be automatically reconstructed. The problem mainly comes from the seriously lack of data and complex freeform surfaces. What is more, the point clouds with poor quality, like photogrammetric point cloud, cannot provide good roof segments therefore the roof topology graph based method cannot be applied to reconstruct buildings. In contrast to only considering the individual building, further work can be taking into account the information that might be learned from the building's context. The geometric and topological relations between building parts could be learned to constrain the model reconstruction. Obvious further ways for improvement are the inclusion of maps and images.

Chapter 6
Conclusions and Recommendations

This thesis introduced new methods on 3D building reconstruction from airborne lidar data. Various novel methods have been discussed, including building regularities, definition of flexible building primitives and automatic correction of roof topology graphs. The conclusions for each step have been given in the previous chapters. This chapter concludes with an overview of the approaches in section 6.1 and a discussion of perspectives for further research in section 6.2.

6.1 Conclusions

This thesis has made two main contributions. Firstly, flexible building primitives have been proposed to solve the problem that many complex buildings cannot be represented by the state-of-the-art building primitives. Secondly, a graph edit dictionary has been introduced to automatically correct the errors in roof topology graphs. In order to solve the problems accompanying the two main contributions, several important contributions have been made, including shape regularities, efficient interactive tools, adaptive constraints, and a single quality criterion. By introducing the improvements inch by inch, the method presented in this thesis is able to automatically reconstruct 95% buildings now.

The work presented in this thesis still has some limitations. The first problem is that about 5% buildings still could not be automatically reconstructed, especially the buildings that are seriously lacking data. It is very difficult to design an algorithm which gives considerations for all buildings. Therefore it is better to manually reconstruct the remaining 5% buildings. The second problem is that most test areas are in the Netherlands. Further tests should be carried out for other kinds of buildings in a number of countries. The flexible building primitives are able to represent all buildings in theory. But the adaption ability of graph edit dictionary should be tested in area where building style differs from the Dutch buildings. Additionally, roofs in other areas, like China, are not as clean as buildings in the Netherlands. Many objects, like air conditionings, clothes hangers, vegetation, and so on, are located on the roofs and will affect the correct detection of roof segments. Only when all differences are considered and adapted, the workflow for 3D building reconstruction can be universally applied.

Another problem is that the proposed method is designed for lidar data, but not well suitable for the photogrammetric point clouds. The photogrammetric point clouds are similar to lidar point clouds, but are much cheaper and have texture by themselves. The photogrammetric point clouds produced by the state-of-the-art methods are less accurate than lidar point cloud. Therefore roof segments cannot be well detected. The methods proposed in this thesis require derived roof topology graphs of a reasonable quality, be it is not perfect. In order to use photogrammetric point clouds, the algorithm should be less relaying on the good roof topology graphs. More model-driven methods could be a choice because they are less sensitive to noise. However, the pure model-driven methods exhaustively fit models to input data and are therefore less accurate and much more time consuming. Roof topology graphs are the middle level information between input data (point clouds, lines and segments) and final building models. Roof topology graphs will still provide valuable hints for building modelling from noisy data. The methods to derive and use roof topology graphs should be well designed.

The building primitives defined in this thesis are more flexible than other building primitives. However, the flexibility is used at the cost of strong constraints. When the input data is of high quality, the flexibility enables us to reconstruct buildings accurately. However, when the input data is poor, the algorithm may result in some odd models. Therefore, the flexible primitives are better to be applied with high quality data.

6.2 Recommendations

This thesis only uses point cloud for the building reconstruction. But in most situations 2D topographic maps are already available. The users who are interested in 3D models usually want the 3D models to be consistent with the 2D maps. Therefore the 2D maps should be taken into account in the reconstruction. Besides, one important application of 3D models is visualization. The models should be mapped with textures for showing details. For large-scale applications we also need to consider update, management and storage.

This thesis uses airborne lidar point clouds to reconstruct LOD2 buildings. Two main research topics would be interesting to extend this research. The first one will be to still reconstruct LOD2 buildings, but using other input data or combination of other data, e.g. stereo images and SAR. The building regularities used in this thesis still hold for any type of data, but the way to use regularities should be adapted to input data. The roof topology graphs derived from data that are less accurate than lidar point cloud have more serious errors and the errors are rarely repeated. Therefore, the graph edit dictionary may be inefficient to correct errors in those roof topology graphs.

The second research topic should be to extend the reconstruction to other objects. All of the important objects that can be seen from airplanes, including trees, towers, lamps, roads, railway, and so on, would be taken into account. The application of lidar will dramatically transform field works of survey into indoor works in near future. In addition, because of the fast improvements of cloud computing and artificial intelligence, the automation will increase quickly, meaning the survey industry is changing and lots of new applications will boom. The 3D maps will be the infrastructure data for better understanding the relationship between societies, cities and nature, and further on for better serving human beings. From this view, the combination of 3D maps with other disciplines, including economics, hydrology, urban planning, and archaeology and so on, will be important research topics in near future.

Bibliography

Adan, A. and Huber, D., 2011. 3D Reconstruction of Interior Wall Surfaces under Occlusion and Clutter.2011 International Conference on 3D Imaging, Modeling, Processing, Visualization and Transmission (3DIMPVT), 275-281.

Agarwal, S., Furukawa, Y., Snavely, N., Curless, B., Seitz, S. and Szeliski, R., 2010. Reconstructing Rome. Computer, 43(6), 40-47.

Agarwal, S., Snavely, N., Simon, I., Seitz, S.M. and Szeliski, R., 2009. Building Rome in a day.IEEE 12th International Conference on Computer Vision, 72-79.

AlHalawani, S., Yang, Y., Liu, H. and Mitra, N.J., 2013. Interactive Facades Analysis and Synthesis of Semi-Regular Facades. Computer Graphics Forum, 32(2pt2), 215-224.

Arikan, M., Schwärzler, M., Flöry, S., Wimmer, M. and Maierhofer, S., 2013. O-snap: Optimization-based snapping for modeling architecture. ACM Transactions on Graphics (TOG), 32(1), 1-15.

Attene, M., Falcidieno, B., Rossignac, J. and Spagnuolo, M., 2003. Edge-sharpener: recovering sharp features in triangulations of non-adaptively re-meshed surfaces.Proceedings of the 2003 Eurographics/ACM SIGGRAPH symposium on Geometry processing, Aachen, Germany, pp. 62-69.

Becker, S. and Haala, N., 2007 Grammar supported facade reconstruction from mobile LIDAR mapping. Proc. CMRT. Int. Arch. Photogramm., Remote Sens. Spatial Inf. Sci, 38, 229–234.

Berger, F., Gritzmann, P. and Vries, S., 2004. Minimum Cycle Bases for Network Graphs. Algorithmica, 40(1), 51-62.

Bernardini, F. and Bajaj, C., 1997. Sampling and Reconstructing Manifolds Using Alpha-Shapes.9th Canadian Conference on Computational Geometry, 193-198.

Bernardini, F., Mittleman, J., Rushmeier, H., Silva, C. and Taubin, G., 1999. The ball-pivoting algorithm for surface reconstruction. IEEE Transactions on Visualization and Computer Graphics, 5(4), 349-359.

Biber, P., Andreasson, H., Duckett, T. and Schilling, A., 2004. 3D modeling of indoor environments by a mobile robot with a laser scanner and panoramic camera.Intelligent Robots and Systems, 2004. (IROS 2004). Proceedings. 2004 IEEE/RSJ International Conference on, 4(3430-3435 vol.4.

Bolter, R., 2000. Reconstruction of man-made objects from high resolution SAR images.2000 IEEE Aerospace Conference Proceedings, 287-292.

Brédif, M., 2010. 3D Building Modeling: Topology-Aware Kinetic Fitting of Polyhedral Roofs and Automatic Roof Superstructure Reconstruction, Télécom ParisTech, France.

Brenner, C., 2000. Towards fully automatic generation of city models. International Archives of Photogrammetry and Remote Sensing, 33(B3/1; PART 3), 84-92.

Brenner, C., 2005. Constraints for modelling complex objects.the CMRT'05, object extraction for 3D city models, road databases, and trafficmonitoring – Concepts, algorithms, and evaluation, IAPRS, Vienna., pp.

Brenner, C. and Haala, N., 1998. Fast production of virtual reality city models. International Archives of Photogrammetry and Remote Sensing, 32(part 4), 77-84.

Brenner, C. and Von Goesseln, G., 2004. Tools and Workflow for the Rapid Acquisition of 3-D City Models.Proceedings of 24th Urban Data Management Symposium, Chioggia, Italy, pp. 11 p.

Brunn, A. and Weidner, U., 1997. Extracting buildings from digital surface models. International Archives of Photogrammetry and Remote Sensing, 32(Part3/4W2), 27-34.

Brunner, D., Lemoine, G. and Bruzzone, L., 2009. Estimation of building heights from detected dual-aspect VHR SAR imagery using an iterative simulation and matching procedure in combination with functional analysis.2009 IEEE Radar Conference, 1-6.

Brunner, D., Lemoine, G., Bruzzone, L. and Greidanus, H., 2010. Building Height Retrieval From VHR SAR Imagery Based on an Iterative Simulation and Matching Technique. IEEE Transactions on Geoscience and Remote Sensing, 48(3), 1487-1504.

Budroni, A. and Boehm, J., 2010. Automated 3D Reconstruction of Interiors from Point Clouds. International Journal of Architectural Computing, 8(1), 55-73.

Cabral, M., Lefebvre, S., Dachsbacher, C. and Drettakis, G., 2009. Structure Preserving Reshape for Textured Architectural Scenes.Computer Graphics Forum 28, 469-480.

Ceylan, D., Mitra, N.J., Li, H., Weise, T. and Pauly, M., 2012. Factored Facade Acquisition using Symmetric Line Arrangements. Computer Graphics Forum, 31(2pt3), 671-680.

Chauve, A.L., Labatut, P. and Pons, J.P., 2010. Robust piecewise-planar 3D reconstruction and completion from large-scale unstructured point data.2010 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 1261-1268.

Chen, G., Esch, G., Wonka, P., Müller, P. and Zhang, E., 2008. Interactive procedural street modeling.SIGGRAPH '08: Proceedings of the 35th Annual Conference on Computer Graphics and Interactive Techniques, New York, NY, USA, pp. 1-10.

Chojnacki, S., 2012. Scoring functions for automatic arrangement of business interiors.SIGGRAPH Asia 2012 Technical Briefs, Singapore, Singapore, pp. 1-4.

Cohen-Steiner, D., Alliez, P. and Desbrun, M., 2004. Variational shape approximation. ACM Transactions on Graphics (TOG), 23(3), 905-914.

Coughlan, J.M. and Yuille, A.L., 1999. Manhattan World: compass direction from a single image by Bayesian inference.The Proceedings of the Seventh IEEE International Conference on Computer Vision, 941-947 vol.2.

Coughlan, J.M. and Yuille, A.L., 2000. The Manhattan world assumption: Regularities in scene statistics which enable Bayesian inference.Advances in Neural Information Processing Systems 13, 845-851.

Dai, D., Prasad, M., Schmitt, G. and Gool, L.V., 2012. Learning domain knowledge for facade labelling.Proceedings of the 12th European conference on Computer Vision - Volume Part I, Florence, Italy, pp. 710-723.

Debevec, P.E., Taylor, C.J. and Malik, J., 1996. Modeling and rendering architecture from photographs: a hybrid geometry- and image-based approach.Proceedings of the 23rd annual conference on Computer graphics and interactive techniques, 11-20.

Demantke, J., Vallet, B. and Paparoditis, N., 2013. Facade Reconstruction with Generalized 2.5d Grids. ISPRS Ann. Photogramm. Remote Sens. Spatial Inf. Sci., II-5/W2, 67-72.

Dick, A., Torr, P. and Cipolla, R., 2004. Modelling and interpretation of architecture from several images. International journal of computer vision, 60(2), 111-134.

Diego, G.-A. and Javier, G.-L., 2008. From 2D TO 3D Through Modelling Based On A Single Image. The Photogrammetric Record, 23(122), 208-227.

Digne, J., Cohen-Steiner, D., Alliez, P., de Goes, F. and Desbrun, M., 2013. Feature-Preserving Surface Reconstruction and Simplification from Defect-Laden Point Sets. Journal of Mathematical Imaging and Vision, 1-14.

Dorninger, P. and 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.

dudviscomm, Shinjuku District of Tokyo, Japan, http://dudviscomm.exteen.com/20100208/yann-arthus-bertrand-earth-from-above, accessed in 2014.

Edelsbrunner, H. and Mücke, E.P., 1994. Three-dimensional alpha shapes. ACM Transactions on Graphics (TOG), 13(1), 43-72.

El-Hakim, S., Whiting, E., Gonzo, L. and Girardi, S., 2005. 3D reconstruction of complex architectures from multiple data. Proceedings of the ISPRS WG V/4 Workshop 3D-ARCH, Mestre-Venice, Italy, 22–24 August, 2005. International Archives of Photogrammetry, Remote Sending and Spatial Information Sciences 36, 5/W17 (Venice 2005).

Esteban, C.H. and Schmitt, F., 2003. Silhouette and stereo fusion for 3D object modeling.3-D Digital Imaging and Modeling, 2003. 3DIM 2003. Proceedings. Fourth International Conference on, 46-53.

Ferro, A., Brunner, D. and Bruzzone, L., 2013. Automatic Detection and Reconstruction of Building Radar Footprints From Single VHR SAR Images. IEEE Transactions on Geoscience and Remote Sensing, 51(2), 935-952.

Fischer, A., Kolbe, T. and Lang, F., 1997a. Integration of 2D and 3D reasoning for building reconstruction using a generic hierarchical model.Proceedings of the Workshop on Semantic Modeling for the Acquisition of Topographic Information from Images and Maps SMATI'97, 159-180.

Fischer, A., Kolbe, T. and Lang, F., 1997b. Integration of 2D and 3D reasoning for building reconstruction using a generic hierarchical model. Semantic Modeling for the Acquisition of Topographic Information from Images and Maps, Proceedings of SMATI, 97, 159-180.

Fleishman, S., Cohen-Or, D. and Silva, C.T., 2005. Robust moving least-squares fitting with sharp features. ACM Transactions on Graphics (TOG), 24(3), 544-552.

Fletcher, R., 2013. Practical methods of optimization. John Wiley & Sons.

Friedman, S. and Stamos, I., 2011. Real Time Detection of Repeated Structures in Point Clouds of Urban Scenes.3DIMPVT, 220-227.

Fröhlich, B., Rodner, E. and Denzler, J., 2013. Semantic Segmentation with Millions of Features: Integrating Multiple Cues in a Combined Random Forest Approach. In: K. Lee, Y. Matsushita, J. Rehg and Z. Hu (Editors), Computer Vision – ACCV 2012. Lecture Notes in Computer Science. Springer Berlin Heidelberg, pp. 218-231.

Frueh, C., Jain, S. and Zakhor, A., 2005. Data Processing Algorithms for Generating Textured 3D Building Facade Meshes from Laser Scans and Camera Images. International Journal of Computer Vision, 61(2), 159-184.

Furukawa, Y., Curless, B., Seitz, S.M. and Szeliski, R., 2009a. Manhattan-world stereo.IEEE Conference on Computer Vision and Pattern Recognition, 2009. CVPR 2009., 1422-1429.

Furukawa, Y., Curless, B., Seitz, S.M. and Szeliski, R., 2009b. Reconstructing building interiors from images.2009 IEEE 12th International Conference on Computer Vision, 80-87.

Garland, M. and Heckbert, P.S., 1997. Surface simplification using quadric error metrics.Proceedings of the 24th annual conference on Computer graphics and interactive techniques, 209-216.

Giraudot, S., Cohen-Steiner, D. and Alliez, P., 2013. Noise-Adaptive Shape Reconstruction from Raw Point Sets. Computer Graphics Forum, 32(5), 229-238.

Greuter, S., Parker, J., Stewart, N. and Leach, G., 2003. Real-time procedural generation of ‘pseudo infinite’cities.GRAPHITE '03: Proceedings of the 1st International Conference on Computer Graphics and Interactive Techniques in Australasia and South East Asia, New York, NY, USA. ,87-94.

Gruen, A. and Wang, X., 1998. CC-Modeler: a topology generator for 3-D city models. ISPRS Journal of photogrammetry and remote sensing, 53(5), 286-295.

Guibas, L., Mitra, N.J. and Nguyen, A., 2004. Estimating Surface Normals in Noisy Point Cloud Data. International Journal of Computational Geometry & Applications, 14(4--5), 261-276.

Guillou, E., Meneveaux, D., Maisel, E. and Bouatouch, K., 2000. Using vanishing points for camera calibration and coarse 3D reconstruction from a single image. The Visual Computer, 16(7), 396-410.

Gülch, E., Muller, H. and Labe, T., 1999. Integration of automatic processes into semi-automatic building extraction. International Archives of Photogrammetry and Remote Sensing, 32(3-2W5), 177-186.

Gutmann, J.S., Fukuchi, M. and Fujita, M., 2005. A Floor and Obstacle Height Map for 3D Navigation of a Humanoid Robot.Proceedings of the 2005 IEEE International Conference on Robotics and Automation, 2005. ICRA 2005. , 1066-1071.

Haala, N., Becker, S. and Kada, M., 2006. Cell decomposition for the generation of building models at multiple scales.Symposium Photogrammetric Computer Vision, 19-24.

Haala, N., Brenner, C. and Anders, K.H., 1998. 3D urban GIS from laser altimeter and 2D map data. International Archives of Photogrammetry and Remote Sensing, 32(Part 3/1), 339-346.

Haala, N. and Kada, M., 2010. An update on automatic 3D building reconstruction. ISPRS Journal of Photogrammetry and Remote Sensing, 65(6), 570-580.

Hähnel, D., Burgard, W. and Thrun, S., 2003. Learning compact 3D models of indoor and outdoor environments with a mobile robot. Robotics and Autonomous Systems, 44(1), 15-27.

Hammoudi, K. and Dornaika, F., 2010. A Featureless Approach to 3D Polyhedral Building Modeling from Aerial Images. Sensors, 11(1), 228-259.

Henn, A., Gröger, G., Stroh, V. and Plümer, L., 2013. Model driven reconstruction of roofs from sparse LIDAR point clouds. ISPRS Journal of photogrammetry and remote sensing, 76, 17-29.

Hirschmuller, H., 2008. Stereo Processing by Semiglobal Matching and Mutual Information. IEEE Transactions on Pattern Analysis and Machine Intelligence, 30(2), 328-341.

Hu, J., You, S. and Neumann, U., 2003. Approaches to large-scale urban modeling. IEEE Computer Graphics and Applications, 23(6), 62-69.

Hu, J., You, S., Neumann, U. and Park, K.K., 2004. Building modeling from LiDAR and aerial imagery.ASPRS'04, Denver, Colorado, USA,4(23-28.

Huang, H., Brenner, C. and Sester, M., 2011. 3D building roof reconstruction from point clouds via generative models.In Proceedings of the 19th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems, 16-24.

Huber, D., Akinci, B., Adan, A., Anil, E., Okorn, B. and Xiong, X., 2011. Methods for automatically modeling and representing as-built building information models.NSF Engineering Research and Innovation Conference, Atlanta.

Izadi, S., Kim, D., Hilliges, O., Molyneaux, D., Newcombe, R., Kohli, P., Shotton, J., Hodges, S., Freeman, D., Davison, A. and Fitzgibbon, A., 2011. KinectFusion: real-time 3D reconstruction and interaction using a moving depth camera.Proceedings of the 24th annual ACM symposium on User interface software and technology, Santa Barbara, California, USA, pp. 559-568.

Jancosek, M. and Pajdla, T., 2011. Multi-view reconstruction preserving weakly-supported surfaces.IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2011), 3121-3128.

Jia, Z., Gallagher, A., Saxena, A. and Chen, T., 2013. 3D-Based Reasoning with Blocks, Support, and Stability.Proceedings of the 2013 IEEE Conference on Computer Vision and Pattern Recognition, 1-8.

Jung, C. and Kim, C., 2012. Real-time estimation of 3D scene geometry from a single image. Pattern Recognition, 45(9), 3256-3269.

Kada, M., 2009. The 3D Berlin Project.Photogrammetric week, Fritsch, D. (Ed.), pp. 331-340.

Kada, M. and McKinley, L., 2009a. 3D building reconstruction from LiDAR based on a cell decomposition approach. International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences, 38(Part 3/W4), 47-52.

Kada, M. and McKinley, L., 2009b. 3D building reconstruction from LiDAR based on a cell decomposition approach. International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences, 38, 47-52.

Karantzalos, K. and Paragios, N., 2010. Large-scale building reconstruction through information fusion and 3-d priors. IEEE Transactions on Geoscience and Remote Sensing, 48(5), 2283-2296.

Kelly, G. and McCabe, H., 2006. A survey of procedural techniques for city generation. ITB Journal, 14, 87-130.

Keqi, Z., Jianhua, Y. and Shu-Ching, C., 2006. Automatic Construction of Building Footprints From Airborne LIDAR Data. IEEE Transactions on Geoscience and Remote Sensing, 44(9), 2523-2533.

Kerber, J., Bokeloh, M., Wand, M. and Seidel, H.P., 2013. Scalable Symmetry Detection for Urban Scenes. Computer Graphics Forum, 32(1), 3-15.

Kim, Y.M., Mitra, N.J., Yan, D.-M. and Guibas, L., 2012. Acquiring 3D indoor environments with variability and repetition. ACM Trans. Graph., 31(6), 1-11.

Kobbelt, L.P., Botsch, M., Schwanecke, U. and Seidel, H.-P., 2001. Feature sensitive surface extraction from volume data.Proceedings of the 28th annual conference on Computer graphics and interactive techniques, 57-66.

Kolbe, T., 2009. Representing and Exchanging 3D City Models with CityGML. In: J. Lee and S. Zlatanova (Editors), 3D Geo-Information Sciences. Lecture Notes in Geoinformation and Cartography. Springer Berlin Heidelberg, pp. 15-31.

Kolluri, R., Shewchuk, J.R. and O'Brien, J.F., 2004. Spectral surface reconstruction from noisy point clouds.Proceedings of the 2004 Eurographics/ACM SIGGRAPH symposium on Geometry processing, 11-21.

Koppula, H.S., Anand, A., Joachims, T. and Saxena, A., 2011. Semantic labeling of 3d point clouds for indoor scenes.Advances in Neural Information Processing Systems, 244-252.

Koutsourakis, P., Simon, L., Teboul, O., Tziritas, G. and Paragios, N., 2009. Single view reconstruction using shape grammars for urban environments.2009 IEEE 12th International Conference on Computer Vision, 1795-1802.

Labatut, P., Pons, J.P. and Keriven, R., 2009. Hierarchical shape-based surface reconstruction for dense multi-view stereo.Computer Vision Workshops (ICCV Workshops), 2009 IEEE 12th International Conference on Computer Vision, 1598-1605.

Lafarge, F., 2013. A Hybrid Multiview Stereo Algorithm for Modeling Urban Scenes. IEEE Transactions on Pattern Analysis and Machine Intelligence, 35(1), 5-17.

Lafarge, F. and Alliez, P., 2013. Surface reconstruction through point set structuring.Computer Graphics Forum, 225-234.

Lafarge, F., Descombes, X., Zerubia, J. and Pierrot-Deseilligny, M., 2008. Automatic building extraction from DEMs using an object approach and application to the 3D-city modeling. ISPRS Journal of Photogrammetry and Remote Sensing, 63(3), 365-381.

Lafarge, F., Descombes, X., Zerubia, J. and Pierrot-Deseilligny, M., 2010. Structural approach for building reconstruction from a single DSM. IEEE Transactions on Pattern Analysis and Machine Intelligence, 32(1), 135-147.

Lafarge, F. and Mallet, C., 2011. Building large urban environments from unstructured point data.Proceedings of the IEEE International Conference on Computer Vision (ICCV), Barcelona, Spain,1068-1075.

Larive, M. and Gaildrat, V., 2006. Wall grammar for building generation.GRAPHITE '06: Proceedings of the 4th International Conference on Computer Graphics and Interactive Techniques in Australasia and Southeast Asia, New York, NY, USA,437.

Leblanc, L., Houle, J. and Poulin, P., 2011. Component-based modeling of complete buildings.Proceedings of Graphics Interface 2011, St. John's, Newfoundland, Canada, pp. 87-94.

Lechner, T., Watson, B., Wilensky, U. and Felsen, M., 2003. Procedural city modeling.1st Midwestern Graphics Conference, St. Louis, MO, USA.

Ledoux, H., 2013. On the Validation of Solids Represented with the International Standards for Geographic Information. Computer-Aided Civil and Infrastructure Engineering, 28(9), 693-706.

Lee, D.C., Gupta, A., Hebert, M. and Kanade, T., 2010. Estimating Spatial Layout of Rooms using Volumetric Reasoning about Objects and Surfaces. Advances in Neural Information Processing Systems (NIPS), 24.

Lee, D.C., Hebert, M. and Kanade, T., 2009. Geometric reasoning for single image structure recovery.IEEE Conference on Computer Vision and Pattern Recognition, 2009. CVPR 2009., 2136-2143.

Li, Y., Wu, X., Chrysathou, Y., Sharf, A., Cohen-Or, D. and Mitra, N.J., 2011. Globfit: Consistently fitting primitives by discovering global relations.ACM Transactions on Graphics (TOG), 30(4), pp. 52.

Li, Z., Dugas-Phocion, G., Samson, J.S. and Seitz, S.M., 2001. Single view modeling of free-form scenes.Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2001. CVPR 2001, 990-997.

Lindenmayer, A., 1968. Mathematical models for cellular interactions in development I. Filaments with one-sided inputs. Journal of theoretical biology, 18(3), 280-299.

Liu, T., Carlberg, M., Chen, G., Chen, J., Kua, J. and Zakhor, A., 2010. Indoor localization and visualization using a human-operated backpack system.2010 International Conference on Indoor Positioning and Indoor Navigation (IPIN), 1-10.

Lowe, D., 2004. Distinctive image features from scale-invariant keypoints. International journal of computer vision, 60(2), 91-110.

Maas, H.G. and Vosselman, G., 1999. Two algorithms for extracting building models from raw laser altimetry data. ISPRS Journal of photogrammetry and remote sensing, 54(2), 153-163.

Martinovi, A., Mathias, M., Weissenberg, J. and Gool, L.V., 2012. A three-layered approach to facade parsing.Proceedings of the 12th European conference on Computer Vision - Volume Part VII, Florence, Italy, pp. 416-429.

Marton, Z.C., Rusu, R.B. and Beetz, M., 2009. On fast surface reconstruction methods for large and noisy point clouds.IEEE International Conference on Robotics and Automation, 2009. ICRA '09, 3218-3223.

Matei, B.C., Sawhney, H.S., Samarasekera, S., Kim, J. and Kumar, R., 2008. Building segmentation for densely built urban regions using aerial lidar data.IEEE Conference on Computer Vision and Pattern Recognition, 2008. CVPR 2008., 1-8.

Merrell, P., 2007. Example-based model synthesis.Proceedings of the 2007 symposium on Interactive 3D graphics and games, 105-112.

Merrell, P., Schkufza, E. and Koltun, V., 2010. Computer-generated residential building layouts. ACM Transactions on Graphics (TOG), 29(6), 1-12.

Micusik, B. and Kosecka, J., 2009. Piecewise planar city 3D modeling from street view panoramic sequences.IEEE Conference on Computer Vision and Pattern Recognition, 2009. CVPR 2009, 2906-2912.

Mičušík, B. and Košecká, J., 2010. Multi-view Superpixel Stereo in Urban Environments. International Journal of Computer Vision, 89(1), 106-119.

Milde, J. and Brenner, C., 2009. Graph-based modeling of building roofs.Proceedings of the 12th AGILE Conference on GIScience, Hannover, Germany,(on CD-ROM).

Mozos, O.M., Mizutani, H., Kurazume, R. and Hasegawa, T., 2012. Categorization of Indoor Places Using the Kinect Sensor. Sensors, 12(5), 6695-6711.

Müller, P., Wonka, P., Haegler, S., Ulmer, A. and Gool, L., 2006. Procedural modeling of buildings. ACM Transactions on Graphics, 25(3), 614-623.

Müller, P., Zeng, G., Wonka, P. and Van Gool, L., 2007a. Image-based procedural modeling of facades. ACM Trans. Graph., 26(3), 85.

Müller, P., Zeng, G., Wonka, P. and Van Gool, L., 2007b. Image-based procedural modeling of facades. ACM Transactions on Graphics, 26(3), 85.

Musialski, P., Wonka, P., Aliaga, D.G., Wimmer, M., van Gool, L. and Purgathofer, W., 2013. A Survey of Urban Reconstruction. Computer Graphics Forum, 32(6), 146-177.

Nan, L., Sharf, A., Zhang, H., Cohen-Or, D. and Chen, B., 2010. SmartBoxes for interactive urban reconstruction.ACM SIGGRAPH 2010 papers, Los Angeles, California, pp. 1-10.

Nan, L., Xie, K. and Sharf, A., 2012. A search-classify approach for cluttered indoor scene understanding. ACM Transactions on Graphics (TOG), 31(6), 137.

Neverova, N., Muselet, D. and Trémeau, A., 2013. $2^{1/2}$D Scene Reconstruction of Indoor Scenes from Single RGB-D Images. In: S. Tominaga, R. Schettini and A. Trémeau (Editors), Computational Color Imaging. Lecture Notes in Computer Science. Springer Berlin Heidelberg, pp. 281-295.

Nie, M., Zollh, M., Izadi, S. and Stamminger, M., 2013. Real-time 3D reconstruction at scale using voxel hashing. ACM Transactions on Graphics (TOG), 32(6), 1-11.

Ochmann, S., Vock, R., Wessel, R., Tamke, M. and Klein, R., 2014. Automatic Generation of Structural Building Descriptions from 3D Point Cloud Scans.GRAPP 2014 - International Conference on Computer Graphics Theory and Applications, Lisbon, Portugal.

Oesau, S., Lafarge, F. and Alliez, P., 2013. Indoor Scene Reconstruction using Primitive-driven Space Partitioning and Graph-cut.Eurographics Workshop on Urban Data Modelling and Visualisation.

Ohori, K.A., Ledoux, H. and Meijers, M., 2012. Validation and automatic repair of planar partitions using a constrained triangulation. Photogrammetrie-Fernerkundung-Geoinformation, 2012(5), 613-630.

Ohtake, Y., Belyaev, A., Alexa, M., Turk, G. and Seidel, H.-P., 2003. Multi-level partition of unity implicits. ACM Transactions on Graphics (TOG), 22(3), 463-470.

Ok, A.O., Wegner, J.D., Heipke, C., Rottensteiner, F., Soergel, U. and Toprak, V., 2012. Matching of straight line segments from aerial stereo images of urban areas. ISPRS Journal of photogrammetry and remote sensing, 74(0), 133-152.

Okorn, B., Xiong, X., Akinci, B. and Huber, D., 2010. Toward automated modeling of floor plans.Proceedings of the Symposium on 3D Data Processing, Visualization and Transmission.

Oßwald, S., Gutmann, J.-S., Hornung, A. and Bennewitz, M., 2011. From 3D point clouds to climbing stairs: A comparison of plane segmentation approaches for humanoids.2011 11th IEEE-RAS International Conference on Humanoid Robots (Humanoids), 93-98.

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

Parish, Y.I. and Müller, P., 2001. Procedural modeling of cities.Proceedings of the 28th annual conference on Computer graphics and interactive techniques, 301-308.

Pauly, M., Keiser, R. and Gross, M., 2003a. Multi-scale Feature Extraction on Point-Sampled Surfaces. Computer Graphics Forum, 22(3), 281-289.

Pauly, M., Keiser, R., Kobbelt, L.P. and Gross, M., 2003b. Shape modeling with point-sampled geometry. ACM Transactions on Graphics (TOG), 22(3), 641-650.

Pauly, M., Mitra, N., Wallner, J., Pottmann, H. and Guibas, L., 2008. Discovering structural regularity in 3D geometry. ACM Transactions on Graphics (TOG), 27(3), 43-43.

Peng, Z., Lei, Y., Honghui, Z. and Long, Q., 2012. Per-pixel translational symmetry detection, optimization, and segmentation.Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on, 526-533.

Perlin, K., 1985. An image synthesizer. ACM Siggraph Computer Graphics, 19(3), 296.

Pollefeys, M., Gool, L.V., Vergauwen, M., Verbiest, F., Cornelis, K., Tops, J. and Koch, R., 2004. Visual Modeling with a Hand-Held Camera. Int. J. Comput. Vision, 59(3), 207-232.

Prusinkiewicz, P., Lindenmayer, A. and Hanan, J., 1988. Development models of herbaceous plants for computer imagery purposes. ACM SIGGRAPH Computer Graphics, 22(4), 141-150.

Prusinkiewicz, P., Lindenmayer, A., Hanan, J.S., Fracchia, F.D., Fowler, D.R., de Boer, M.J. and Mercer, L., 1990. The algorithmic beauty of plants, 2. Springer-Verlag New York.

Pu, S. and Vosselman, G., 2009. Knowledge based reconstruction of building models from terrestrial laser scanning data. ISPRS Journal of photogrammetry and remote sensing, 64(6), 575-584.

Reeves, W., 1983. Particle systems—a technique for modeling a class of fuzzy objects. ACM Transactions on Graphics (TOG), 2(2), 108.

Reynolds, C., 1987. Flocks, herds and schools: A distributed behavioral model.ACM SIGGRAPH Computer Graphics, 25-34.

Riemenschneider, H., Krispel, U., Thaller, W., Donoser, M., Havemann, S., Fellner, D. and Bischof, H., 2012. Irregular lattices for complex shape grammar facade parsing.2012 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 1640-1647.

Ripperda, N. and Brenner, C., 2006. Reconstruction of Façade Structures Using a Formal Grammar and RjMCMC. In: K. Franke, K.-R. Müller, B. Nickolay and R. Schäfer (Editors), Pattern Recognition. Lecture Notes in Computer Science. Springer Berlin Heidelberg, pp. 750-759.

Rottensteiner, F., Sohn, G., Gerke, M., Wegner, J.D., Breitkopf, U. and Jung, J., 2014. Results of the ISPRS benchmark on urban object detection and 3D building reconstruction. ISPRS Journal of Photogrammetry and Remote Sensing, 93, 256-271.

Russell, B.C. and Torralba, A., 2009. Building a database of 3D scenes from user annotations.IEEE Conference on Computer Vision and Pattern Recognition, 2009. CVPR 2009, 2711-2718.

Rusu, R.B., Marton, Z.C., Blodow, N., Dolha, M. and Beetz, M., 2008. Towards 3D Point cloud based object maps for household environments. Robotics and Autonomous Systems, 56(11), 927-941.

Rusu, R.B., Marton, Z.C., Blodow, N., Holzbach, A. and Beetz, M., 2009. Model-based and learned semantic object labeling in 3D point cloud maps of kitchen environments.IEEE/RSJ

International Conference on Intelligent Robots and Systems, 2009. IROS 2009, 3601-3608.

Rutzinger, M., Rottensteiner, F. and Pfeifer, N., 2009. A comparison of evaluation techniques for building extraction from airborne laser scanning. IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing, 2(1), 11-20.

Sampath, A. and Shan, J., 2010. Segmentation and reconstruction of polyhedral building roofs from aerial lidar point clouds. IEEE Transactions on Geoscience and Remote Sensing, 48(3), 1554-1567.

Satoshi, I., Yoshihiro, K., Jun, M. and Yukio, F., 2012. Efficiently Modeling 3D Scenes from a Single Image. IEEE Computer Graphics and Applications, 32(6), 18-25.

Saxena, A., Chung, S. and Ng, A., 2008. 3-D Depth Reconstruction from a Single Still Image. International Journal of Computer Vision, 76(1), 53-69.

Saxena, A., Min, S. and Ng, A.Y., 2007a. 3-D Reconstruction from Sparse Views using Monocular Vision.IEEE 11th International Conference on Computer Vision, 2007. ICCV 2007, 1-8.

Saxena, A., Min, S. and Ng, A.Y., 2007b. Learning 3-D Scene Structure from a Single Still Image.IEEE 11th International Conference on Computer Vision, 2007. ICCV 2007, 1-8.

Saxena, A., Min, S. and Ng, A.Y., 2009. Make3D: Learning 3D Scene Structure from a Single Still Image. IEEE Transactions on Pattern Analysis and Machine Intelligence, 31(5), 824-840.

Schindler, K. and Bauer, J., 2003. A model-based method for building reconstruction.First IEEE International Workshop on Higher-Level Knowledge in 3D Modeling and Motion Analysis, 2003. HLK 2003, 74-82.

Schnabel, R., Wahl, R. and Klein, R., 2007. Efficient RANSAC for Point-Cloud Shape Detection. Computer Graphics Forum, 26(2), 214-226.

Shapiro, V. and Vossler, D.L., 1993. Separation for boundary to CSG conversion. ACM Transactions on Graphics (TOG), 12(1), 35-55.

Shen, C., Huang, S., Fu, H. and Hu, S., 2011. Adaptive partitioning of urban facades.Proceedings of the 2011 SIGGRAPH Asia Conference, Hong Kong, China, pp. 1-10.

Sinha, S.N., Ramnath, K. and Szeliski, R., 2012. Detecting and reconstructing 3d mirror symmetric objects.Proceedings of the 12th European conference on Computer Vision - Volume Part II, Florence, Italy, pp. 586-600.

Sinha, S.N., Steedly, D., Szeliski, R., Agrawala, M. and Pollefeys, M., 2008. Interactive 3D architectural modeling from unordered photo collections.ACM Transactions on Graphics (TOG), 27(159.

Smisek, J., Jancosek, M. and Pajdla, T., 2011. 3D with Kinect.2011 IEEE International Conference on Computer Vision Workshops (ICCV Workshops), 1154-1160.

Snavely, N., Seitz, S. and Szeliski, R., 2006. Photo tourism: exploring photo collections in 3D. ACM transactions on graphics (TOG), 25(3), 835-846.

Snyder, G.I., The 3D Elevation Program--Summary of Program Direction: U.S. Geological Survey Fact Sheet 2012-3089, http://pubs.usgs.gov/fs/2012/3089/, accessed in 2012.

Sohn, G. and Dowman, I., 2003. Building extraction using Lidar DEMs and Ikonos images. International Archives of Photogrammetry and Remote Sensing, 34, 3.

Sohn, G., Huang, X. and Tao, V., 2008. Using a binary space partitioning tree for reconstructing polyhedral building models from airborne lidar data. Photogrammetric Engineering and Remote Sensing, 74(11), 1425-1440.

Sohn, G., Jung, J., Jwa, Y. and Armenakis, C., 2013. Sequential Modelling Of Building Rooftops By Integrating Airborne LiDAR Data And Optical Imagery: Preliminary Results. ISPRS Ann. Photogramm. Remote Sens. Spatial Inf. Sci., II-3/W1, 27-33.

Sohn, G., Jwa, Y., Tao, V. and Cho, W., 2007. Geometric Regualrization of Irregular Building Polygons: A Comparative Study. Korean Journal of Geomatics, 25(6-1), 545-555.

Steinhage, V., Behley, J., Meisel, S. and Cremers, A., 2013. Reconstruction by components for automated updating of 3D city models. Applied Geomatics, 5(4), 285-298.

Sturm, P., 2000. A method for 3D reconstruction of piecewise planar objects from single panoramic images.IEEE Workshop on Omnidirectional Vision, 2000, 119-126.

Suveg, I. and Vosselman, G., 2004. Reconstruction of 3D building models from aerial images and maps. ISPRS Journal of photogrammetry and remote sensing, 58(3), 202-224.

Taillandier, F., 2005. Automatic building reconstruction from cadastral maps and aerial images. International Archives of Photogrammetry and Remote Sensing, 36(Part 3/W24), 105-110.

Taillandier, F. and Deriche, R., 2004. Automatic buildings reconstruction from aerial images: a generic bayesian framework.Proceedings of the XXth ISPRS Congress, Istanbul, Turkey.

Tang, P., Huber, D., Akinci, B., Lipman, R. and Lytle, A., 2010. Automatic reconstruction of as-built building information models from laser-scanned point clouds: A review of related techniques. Automation in Construction, 19(7), 829-843.

Teboul, O., Simon, L., Koutsourakis, P. and Paragios, N., 2010. Segmentation of building facades using procedural shape priors.Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR).

Thiele, A., Cadario, E., Schulz, K., Thoennessen, U. and Soergel, U., 2008. Reconstruction of residential buildings from multi-aspect InSAR data.Proceedings of ESA-EUSC Workshop, March.

Thiele, A., Cadario, E., Schulz, K., Thonnessen, U. and Soergel, U., 2007. Building Recognition From Multi-Aspect High-Resolution InSAR Data in Urban Areas. IEEE Transactions on Geoscience and Remote Sensing, 45(11), 3583-3593.

Thiele, A., Hinz, S., Cadario, E. and Lacoste, H., 2010. Fusion of InSAR and GIS data for 3D building reconstruction and change detection.Proceedings of 'Fringe 2009 Workshop', Frascati, Italy, ESA SP-677, http://earth.eo.esa.int/workshops/fringe09/.

Tian, Y., Gerke, M., Vosselman, G. and Zhu, Q., 2010. Knowledge-based building reconstruction from terrestrial video sequences. ISPRS Journal of photogrammetry and remote sensing, 65(4), 395-408.

Tianfan, X., Jianzhuang, L. and Xiaoou, T., 2011. Symmetric piecewise planar object reconstruction from a single image.2011 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2577-2584.

Turner, E. and Zakhor, A., 2012. Watertight As-Built Architectural Floor Plans Generated from Laser Range Data.2012 Second International Conference on 3D Imaging, Modeling, Processing, Visualization and Transmission (3DIMPVT), 316-323.

Turner, E. and Zakhor, A., 2013. Watertight Planar Surface Meshing of Indoor Point-Clouds with Voxel Carving.3DV-Conference, 2013 International Conference on, 41-48.

Turner, E. and Zakhor, A., 2014. Floor Plan Generation and Room Labeling of Indoor Environments from Laser Range Data.GRAPP 2014, Lisbon, Portugal, pp.

Tutenel, T., Smelik, R.M., Lopes, R., de Kraker, K.J. and Bidarra, R., 2011. Generating Consistent Buildings: A Semantic Approach for Integrating Procedural Techniques. IEEE Transactions on Computational Intelligence and AI in Games, 3(3), 274-288.

Tylecek, R. and Sara, R., 2011. Modeling symmetries for stochastic structural recognition.2011 IEEE International Conference on Computer Vision Workshops (ICCV Workshops), 632-639.

Tyleček, R. and Šára, R., 2013. Spatial Pattern Templates for Recognition of Objects with Regular Structure. In: J. Weickert, M. Hein and B. Schiele (Editors), Pattern Recognition. Lecture Notes in Computer Science. Springer Berlin Heidelberg, pp. 364-374.

Valero, E., Adan, A. and Cerrada, C., 2012a. Automatic Construction of 3D Basic-Semantic Models of Inhabited Interiors Using Laser Scanners and RFID Sensors. Sensors, 12(5), 5705-5724.

Valero, E., Adán, A. and Cerrada, C., 2012b. Automatic Method for Building Indoor Boundary Models from Dense Point Clouds Collected by Laser Scanners. Sensors, 12(12), 16099-16115.

Vanegas, C.A., Aliaga, D.G., Wonka, P., Müller, P., Waddell, P. and Watson, B., 2010. Modelling the Appearance and Behaviour of Urban Spaces. Computer Graphics Forum, 29(1), 25-42.

Verdie, Y., Lafarge, F. and Zerubia, J., 2011. Generating compact meshes under planar constraints: An automatic approach for modeling buildings from aerial LiDAR. 18th IEEE International Conference on Image Processing (ICIP), 877-880.

Verma, V., Kumar, R. and Hsu, S., 2006. 3D building detection and modeling from aerial lidar data.Proceedings IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06), Washington, DC, USA, pp. 2213-2220.

Vosselman, G., 2002. Fusion of laser scanning data, maps, and aerial photographs for building reconstruction.IEEE International Geoscience and Remote Sensing Symposium, IGARSS'02, Toronto, Canada, 24-28June,on CD-ROM, 4 p.

Vosselman, G., 2012. Automated planimetric quality control in high accuracy airborne laser scanning surveys. ISPRS Journal of photogrammetry and remote sensing, 74, 90-100.

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

Vosselman, G., Gorte, B.G.H., Sithole, G. and Rabbani, T., 2004. Recognising structure in laser scanner point clouds. International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences, 46(Part 8/W2), 33-38.

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

Wang, G., Tsui, H.-T., Hu, Z. and Wu, F., 2005. Camera calibration and 3D reconstruction from a single view based on scene constraints. Image and Vision Computing, 23(3), 311-323.

Wang, R., 2013. 3D building modeling using images and LiDAR: a review. International Journal of Image and Data Fusion, 4(4), 273-292.

Watson, B., Müller, P., Veryovka, O., Fuller, A., Wonka, P. and Sexton, C., 2008. Procedural urban modeling in practice. IEEE Computer Graphics and Applications, 28(3), 18-26.

Weber, C., Hahmann, S. and Hagen, H., 2010. Sharp Feature Detection in Point Clouds.Shape Modeling International Conference (SMI), 2010, 175-186.

Weidner, U. and Förstner, W., 1995. Towards automatic building extraction from high-resolution digital elevation models. ISPRS Journal of photogrammetry and remote sensing, 50(4), 38-49.

Weissenberg, J., Riemenschneider, H., Prasad, M. and Van Gool, L., 2013. Is There a Procedural Logic to Architecture?Computer Vision and Pattern Recognition (CVPR), 2013 IEEE Conference on, 185-192.

Wenzel, S. and Förstner, W., 2008. Semi-supervised incremental learning of hierarchical appearance models.Proceeding of the 21st Congress of the International Society for Photogrammetry and Remote Sensing (ISPRS), 3(399-405.

Werghi, N., Fisher, R., Robertson, C. and Ashbrook, A., 1999. Object reconstruction by incorporating geometric constraints in reverse engineering. Computer-Aided Design, 31(6), 363-399.

Werner, T. and Zisserman, A., 2002. New Techniques for Automated Architectural Reconstruction from Photographs.Proceedings of the 7th European Conference on Computer Vision-Part II, 541-555.

Whiting, E., Ochsendorf, J. and Durand, F., 2009. Procedural modeling of structurally-sound masonry buildings. ACM Transactions on Graphics (TOG), 28(5), 1-9.

Wilczkowiak, M., Boyer, E. and Sturm, P., 2001. Camera calibration and 3D reconstruction from single images using parallelepipeds.Eighth IEEE International Conference on Computer Vision, 2001. ICCV 2001, 142-148 vol.1.

Wonka, P., Wimmer, M., Sillion, F. and Ribarsky, W., 2003. Instant architecture. ACM Transactions on Graphics (TOG), 22, 669-677.

Wurm, K.M., Hornung, A., Bennewitz, M., Stachniss, C. and Burgard, W., 2010. OctoMap: A probabilistic, flexible, and compact 3D map representation for robotic systems.Proc. of the ICRA 2010 workshop on best practice in 3D perception and modeling for mobile manipulation.

Xiao, J., Fang, T., Tan, P., Zhao, P., Ofek, E. and Quan, L., 2008. Image-based façade modeling. ACM Trans. Graph., 27(5), 1-10.

Xiao, J., Fang, T., Zhao, P., Lhuillier, M. and Quan, L., 2009. Image-based street-side city modeling. ACM transactions on Graphics (TOG), 28(5), 1-12.

Xiao, J. and Furukawa, Y., 2012a. Reconstructing the World's Museums.Proceedings of the 12th European Conference on Computer Vision.

Xiao, J. and Furukawa, Y., 2012b. Reconstructing the World’s Museums. Computer Vision–ECCV 2012, 668-681.

Xiong, X., Adan, A., Akinci, B. and Huber, D., 2013. Automatic creation of semantically rich 3D building models from laser scanner data. Automation in Construction, 31(0), 325-337.

Yun, J., Koppula, H. and Saxena, A., 2013. Hallucinated Humans as the Hidden Context for Labeling 3D Scenes.2013 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2993-3000.

Zebedin, L., Bauer, J., Karner, K. and Bischof, H., 2008a. Fusion of Feature- and Area-Based Information for Urban Buildings Modeling from Aerial Imagery. In: D. Forsyth, P. Torr and A. Zisserman (Editors), Computer Vision – ECCV 2008. Lecture Notes in Computer Science. Springer Berlin Heidelberg, pp. 873-886.

Zebedin, L., Bauer, J., Karner, K. and Bischof, H., 2008b. Fusion of feature-and area-based information for urban buildings modeling from aerial imagery.Computer Vision–ECCV 2008, Springer Berlin Heidelberg, pp. 873-886.

Zhao, Y. and Zhu, S., 2013. Scene Parsing by Integrating Function, Geometry and Appearance Models.2013 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 3119-3126.

Zheng, Q., Sharf, A., Wan, G., Li, Y., Mitra, N.J., Cohen-Or, D. and Chen, B., 2010. Non-local scan consolidation for 3D urban scenes. ACM Trans. Graph., 29(4), 1-9.

Zheng, X., Zhang, X. and Guo, P., 2011. Building modeling from a single image applied in urban reconstruction.Proceedings of the 10th International Conference on Virtual Reality Continuum and Its Applications in Industry, Hong Kong, China, pp. 225-234.

Zhou, J. and Li, B., 2011. Rapid modeling of cones and cylinders from a single calibrated image using minimum 2D control points. Machine Vision and Applications, 22(2), 303-321.

Zhou, Q.Y. and Neumann, U., 2010. 2.5 d dual contouring: A robust approach to creating building models from aerial lidar point clouds.Proceedings European Conference on Computer Vision (ECCV 2010), Springer-Verlag Berlin, Heidelberg, pp. 115-128.

Zhou, Q.Y. and Neumann, U., 2011. 2.5 D building modeling with topology control.Proceedings IEEE Computer Society

Conference on Computer Vision and Pattern Recognition (CVPR'11), Colorado Springs, USA,2489-2496.

Zhou, Q.Y. and Neumann, U., 2012. 2.5 D building modeling by discovering global regularities.Computer Vision and Pattern Recognition (CVPR), Providence, RI, pp. 326-333.

Zhu, X. and Bamler, R., 2010. Very High Resolution Spaceborne SAR Tomography in Urban Environment. IEEE Transactions on Geoscience and Remote Sensing, 48(12), 4296-4308.

Zhu, X. and Shahzad, M., 2014. Facade Reconstruction Using Multiview Spaceborne TomoSAR Point Clouds. IEEE Transactions on Geoscience and Remote Sensing, 52(6), 3541-3552.

Ziegler, R., Matusik, W., Pfister, H. and McMillan, L., 2003. 3D reconstruction using labeled image regions.Proceedings of the 2003 Eurographics/ACM SIGGRAPH symposium on Geometry processing, 248-259.

Zou, C., Liu, J. and Liu, J., 2013. Precise 3D Reconstruction from a Single Image. In: K. Lee, Y. Matsushita, J. Rehg and Z. Hu (Editors), Computer Vision – ACCV 2012. Lecture Notes in Computer Science. Springer Berlin Heidelberg, pp. 271-282.

Publications

ISI Journal Paper:

Xiong, B., Jancosek, M., Oude Elberink, S. and Vosselman, G. 2014. Flexible building primitives for 3D building modeling ISPRS Journal of photogrammetry and remote sensing. (Under review after minor revision)

Xiong, B., Oude Elberink, S. and Vosselman, G. 2014. A graph edit dictionary for correcting errors in roof topology graphs reconstructed from point clouds. ISPRS Journal of photogrammetry and remote sensing, http://dx.doi.org/10.1016/j.isprsjprs.2014.01.007.

International Conference Paper:

Xiong, B., Oude Elberink, S., and Vosselman, G.: Building modeling from noisy photogrammetric point clouds, ISPRS Ann. Photogramm. Remote Sens. Spatial Inf. Sci., II-3, 197-204, doi:10.5194/isprsannals-II-3-197-2014, 2014.(peer reviewed)

Xiong, B., Oude Elberink, S. and Vosselman, G. Effects of Building Primitive Library and Automatic Correction of Roof Topology Graph. in ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences,Volume II-5/W2, 2013 ISPRS Workshop Laser Scanning 2013, 11 – 13 November 2013, Antalya, Turkey. (peer reviewed)

Curriculum Vitae

Biao Xiong was borne on October 1985 in Jiangxi, China. From 2003 to 2007, he studied in the School of Geodesy and Geomatics (SGG), Wuhan University, where he received his Bachelor's degree in survey engineering as an outstanding student. From 2007 to 2010, he enrolled in the Laboratory for Information Engineering in Survey, Mapping and Remote Sensing (LIESMARS), Wuhan University, with a specialization in Photogrammetry and Remote Sensing. He received a Master of Science degree with honour with the thesis 'Integrating space and spectral scale for interpreting high resolution image'. In 2010, he received Chinese Scholarship Council (CSC) grants to start PhD research in the Faculty of Geo-Information Science and Earth Observation, University of Twente. His research is on using shape knowledge to reconstruct 3D building models. His research interest is on lidar data processing, 3D modelling, (photogrammetric) computer vision, and computer graphics.

ITC Dissertation List

http://www.itc.nl/research/phd/phd_graduates.aspx