Grammar-Guided Reconstruction of Semantic 3D Building Models From Airborne LiDAR Data Using Half-Space Modeling

vorgelegt von
M.Sc.
Andreas Wichmann
geb. in Al-Khobar (Saudi-Arabien)

von der Fakultät VI – Planen Bauen Umwelt
der Technischen Universität Berlin
zur Erlangung des akademischen Grades

Doktor der Ingenieurwissenschaften
- Dr.-Ing. -

genehmigte Dissertation

Promotionsausschuss:
Vorsitzender: Prof. Dr. Dr. h.c. Harald Schuh, Technische Universität Berlin
Gutachter: Prof. Dr.-Ing. Martin Kada, Technische Universität Berlin
Gutachter: Prof. Dr. Gunho Sohn, York University, Kanada
Gutachter: Prof. dr.ir. M.G. George Vosselman, University of Twente, Niederlande
Gutachter: Prof. Dr.-Ing. Frank Neitzel, Technische Universität Berlin

Tag der wissenschaftlichen Aussprache: 19. Januar 2018

Berlin 2018

Abstract

For several years, digital 3D city models have taken a central role in a number of different tasks. These models are used in urban and regional planning, surveying, and navigation and telecommunication technology. They also enable in the environmental field precise analyses and simulations of pollutant, flood, and noise propagation. For many applications, realistic building models are an essential component of a 3D city model. Their manual reconstruction provides good results but is associated with a very high time expenditure. In order to meet the high demand for building models and the increased demands on their level of detail, research has been recently conducted to develop various (semi-)automatic reconstruction methods for the time-efficient and cost-effective generation of 3D building models. The goal of designing an automatic building reconstruction method, which fully meets the high demands of the present day, has, however, not yet been achieved. The reasons for this are manifold; a major reason is, for example, the large number of different and complex roof shapes.

In this thesis, a new approach to the fully automatic reconstruction of semantic 3D building models based on airborne LiDAR data is presented. The approach is characterized by a strong integration of building knowledge, which is automatically derived during the reconstruction. The derivation of building knowledge is carried out by employing a grammar whose production rules are applied in several bottom-up and top-down phases. In the selection process of the applicable production rules, methods of reinforcement learning from the field of machine learning are utilized. Thereby, it is taken into account that in complex roof structures and neighboring buildings similar roof elements or roof structures often occur multiple times. Through the application of the grammar, knowledge about the buildings is derived. This knowledge is modeled for each building in a separate multi-scale knowledge graph. The use of the grammar, together with the multi-scale knowledge graphs, in alternating bottom-up and top-down phases provides a reliable and robust derivation of further building knowledge.

The grammar-guided method for deriving building knowledge can generally be integrated also into already existing data-driven reconstruction methods to improve their performance. This procedure is demonstrated exemplarily for a data-driven reconstruction based on binary space partitioning.

In order to reduce the search space for the automatic derivation of building knowledge, a method for the automatic segmentation of roof surfaces is presented. It uses the building points of the underlying data set and employs the principles of surface growing. However, not only the measured height values of the point cloud are incorporated in this procedure, but additionally also virtual points, which are automatically generated during segmentation. As a result, segments can also grow below other roof surfaces, whereby the derivation of building knowledge and the construction of 3D building models are considerably simplified.

Abstract

For the construction of 3D building models, a method based on half-space modeling is presented. It utilizes the building knowledge in the multi-scale knowledge graphs to define and combine half-spaces, whose hyperplanes result from the segments of the point cloud. In this way, buildings of arbitrary complexity are described in canonical form. The resulting building models have the property that they are always both geometrically and topologically correct. The building geometry of the reconstructed 3D building models can be enriched by the semantic information of the multi-scale knowledge graphs. Thereby, also the need for semantic 3D building models is satisfied.

In general, the requirements for reconstructed building models depend on the individual application. To take this into account, two methods are presented which can optionally be executed during the reconstruction. The first method is concerned with the reconstruction of small roof elements (e.g., dormers or chimneys) in order to increase the level of detail of insufficiently detailed building models. It does not reconstruct multiple occurrences of roof elements of similar shape individually from one another, but jointly by means of an ICP (iterative closest point) based method. As a result, also those roof elements are recognized and reconstructed which taken individually would not have been reconstructed. The second method recognizes the frequently occurring regularities in buildings and renders them more strongly in the reconstructed models. For this, divisive clustering methods are applied, both in a local and a global context, to the hyperplanes of the half-spaces that are used to define the half-space models in canonical form. The resulting well-shaped building models are particularly suitable for visualization purposes.

Finally, the presented reconstruction method together with the resulting building models is evaluated using a benchmark data set and compared with other current state-of-the-art building reconstruction approaches. The results show that the presented method is robust and transferable to different building shapes and regions, and that the resulting 3D building models have a high quality with regard to their completeness and correctness.

Zusammenfassung

Seit mehreren Jahren nehmen digitale 3D-Stadtmodelle in verschiedenen Aufgabenstellungen eine zentrale Rolle ein. Sie werden unter anderem in der Stadt- und Raumplanung, im Vermessungswesen und in der Navigations- und Telekommunikationstechnik eingesetzt. Zudem ermöglichen sie im Umweltbereich präzise Analysen und Simulationen zur Ausbreitung von Hochwasser, Lärm und Schadstoffen. Ein für viele Anwendungszwecke wesentlicher Bestandteil von 3D-Stadtmodellen sind realitätsgetreue Gebäudemodelle. Deren manuelle Rekonstruktion bietet gute Ergebnisse, ist aber mit einem sehr hohen Zeitaufwand verbunden. Um der hohen Nachfrage nach Gebäudemodellen und den gestiegenen Anforderungen an deren Detailierungsgrad nachzukommen, hat man sich in den letzten Jahren mit ihrer zeiteffizienten und kostengünstigen Erzeugung beschäftigt und einige (semi-) automatische Rekonstruktionsmethoden entwickelt. Das Ziel ein Verfahren zur automatischen Gebäuderekonstruktion zu entwerfen, das den hohen Ansprüchen der heutigen Zeit gerecht wird, wurde bislang jedoch noch nicht vollständig erreicht. Die Ursachen hierfür sind vielfältig; z.B. ist ein wesentlicher Grund die große Anzahl unterschiedlicher und komplexer Dachformen.

In der vorliegenden Arbeit wird ein neuer Ansatz zur vollautomatischen Rekonstruktion von semantischen 3D-Gebäudemodellen auf der Basis von luftgestützten LiDAR-Daten vorgestellt. Der Ansatz zeichnet sich durch eine starke Integration von Gebäudewissen aus, welches während der Rekonstruktion automatisch abgeleitet wird. Die Ableitung von Gebäudewissen erfolgt über eine Grammatik, deren Produktionsregeln in mehreren Bottom-Up- und Top-Down-Phasen angewendet werden. Bei der Selektierung der anzuwendenden Produktionsregeln werden Methoden des bestärkenden Lernens aus dem Bereich des maschinellen Lernens eingesetzt. Dadurch wird berücksichtigt, dass in komplexen Dachstrukturen und benachbarten Gebäuden oftmals ähnliche Dachelemente oder Dachstrukturen mehrfach vorkommen. Das durch die Anwendung der Grammatik abgeleitete Gebäudewissen wird für jedes Gebäude in jeweils einem multiskaligen Wissensgraphen modelliert. Der Einsatz der Grammatik zusammen mit den Wissensgraphen in sich abwechselnden Bottom-Up- und Top-Down-Phasen ermöglicht eine zuverlässige und robuste Ableitung von weiterem Gebäudewissen.

Das durch eine Grammatik geleitete Verfahren zur Ableitung von Gebäudewissen lässt sich generell auch in bereits existierende datengetriebene Rekonstruktionsverfahren zu deren Vorteil integrieren. Demonstriert wird dies exemplarisch anhand einer datengetriebenen Rekonstruktion, die auf einer binären Raumpartitionierung (engl. binary space partitioning) basiert.

Um den Suchraum für die automatische Ableitung von Gebäudewissen zu verringern, wird eine Methode zur automatischen Segmentierung von Dachflächen vorgestellt. Sie verwendet

die Gebäudepunkte des zugrundeliegenden Datensatzes und die Prinzipien des Oberflächenwachstums (engl. surface growing). Es werden jedoch nicht nur die gemessenen Höhenwerte der Punktwolke in diesem Prozess berücksichtigt, sondern zusätzlich auch virtuelle Punkte, die während der Segmentierung automatisch erzeugt werden. Dies ermöglicht das Wachstum von Segmenten auch unterhalb anderer Dachflächen, wodurch die Ableitung von Gebäudewissen und die Konstruktion der 3D-Gebäudemodelle wesentlich vereinfacht werden.

Für die Konstruktion der 3D-Gebäudemodelle wird ein Verfahren basierend auf der Halbraummodellierung vorgestellt. Es nutzt das Gebäudewissen in den multiskaligen Wissensgraphen für die Definition und Kombination von Halbräumen, deren Hyperebenen sich aus den Segmenten der Punktwolke ergeben. Auf diese Weise werden Gebäude beliebiger Komplexität in kanonischer Form beschrieben. Die daraus abgeleiteten Gebäudemodelle haben die Eigenschaft, dass sie sowohl geometrisch als auch topologisch immer korrekt sind. Die Gebäudegeometrie der rekonstruierten 3D-Gebäudemodelle lässt sich durch die semantischen Informationen der multiskaligen Wissensgraphen anreichern. Dadurch wird auch der Bedarf an semantischen 3D-Gebäudemodellen befriedigt.

Generell hängen die Anforderungen an rekonstruierten Gebäudemodellen von der individuellen Anwendung ab. Um dies zu berücksichtigen werden zwei Verfahren vorgestellt, die optional während der Rekonstruktion ausgeführt werden können. Das erste Verfahren beschäftigt sich mit der Rekonstruktion von kleinen Dachelementen (z.B. Gauben oder Schornsteinen), um den Detailierungsgrad von nicht ausreichend detaillierten Gebäudemodellen zu erhöhen. Es rekonstruiert mehrfach und in gleicher Form vorkommende Dachelemente nicht unabhängig voneinander, sondern gemeinsam mittels einer ICP (engl. iterative closest point) basierten Methode. Dadurch werden auch solche Dachelemente erkannt und rekonstruiert, die einzeln für sich genommen nicht rekonstruiert worden wären. Das zweite Verfahren erkennt die in Gebäuden häufig auftretenden Regularitäten und gibt sie in den rekonstruierten Modellen verstärkt wieder. Dazu werden divisive Clusterverfahren sowohl im lokalen als auch im globalen Kontext auf die Hyperebenen der Halbräume der in der kanonischen Form definierten Halbraummodelle angewandt. Die daraus resultierenden formschönen Gebäudemodelle eignen sich insbesondere für Visualisierungszwecke.

Schließlich wird das vorgestellte Rekonstruktionsverfahren mit den daraus resultierenden Gebäudemodellen anhand eines Benchmark-Datensatzes evaluiert und mit anderen aktuellen Gebäuderekonstruktionsverfahren verglichen. Die Ergebnisse zeigen, dass das vorgestellte Verfahren sowohl robust als auch auf unterschiedliche Gebäudeformen und Regionen übertragbar ist, und dass die daraus resultierenden 3D-Gebäudemodelle eine hohe Qualität in Bezug auf deren Vollständigkeit und Korrektheit aufweisen.

Contents

Contents

List of Tables

List of Tables

List of Figures

List of Abbreviations

ALS	Airborne Laser Scanning
AOI	Area of Interest
BSP	Binary Space Partitioning
B-rep	Boundary Representation
CCA	Connected Component Analysis
CGA	Computer Generated Architecture
CLF	Compass Line Filter
CSG	Constructive Solid Geometry
DGPF	German Society for Photogrammetry, Remote Sensing and Geoinformation
DMC	Digital Mapping Camera
DPO	Double Pushout
DSM	Digital Surface Model
DTM	Digital Terrain Model
FEM	Finite Element Method
FL-system	Functional Lindenmayer-System
FN	False Negative
FP	False Positive
G^2	Generalized Grammar
GG	Graph Grammar
GIS	Geographic Information System
GRR	Generalized Rewrite Rule
GSD	Ground Sampling Distance
HAT	Hypothesize and Test
ICP	Iterative Closest Point
ILP	Inductive Logic Programming
ISPRS	International Society of Photogrammetry and Remote Sensing
LHS	Left-Hand Side
LiDAR	Light Detection and Ranging
LoD	Level of Detail
L-system	Lindenmayer System

MDL	Minimum Description Length
MSE	Mean Squared Error
NARF	Normal Aligned Radial Feature
NRCA	National Roofing Contractors Association
PCA	Principal Component Analysis
PGA	Parallel Generation of Architecture
RANSAC	Random Sample Consensus
RHS	Right-Hand Side
RJMCMC	Reversible Jump Markov Chain Monte Carlo
RMS	Root Mean Square
RMSE	Root Mean Square Error
RTG	Roof Topology Graph
SIFT	Scale-Invariant Feature Transform
SPO	Single Pushout
SVM	Support Vector Machine
TIN	Triangular Irregular Network
TP	True Positive

1. Introduction

During the last decades, several approaches for the reconstruction of 3D building models have been developed. Starting in the 1980s with manual and semi-automatic reconstruction methods of 3D building models from aerial images, the degree of automation has increased in recent years so that they became applicable to various areas. Some typical applications and examples are shown in section 1.1. Especially since the 1990s, when airborne light detection and ranging (LiDAR) technology became widely available, approaches for (semi-)automatic building reconstruction of large urban areas turned out to be of particular interest. Only in recent years, some large cities have built detailed 3D city models. Although much effort has been put into the development of a fully automatic reconstruction strategy in order to overcome the high costs of semi-automatic reconstructions, no solution proposed so far meets all requirements (e.g., in terms of completeness, correctness, and accuracy). The reasons for this are manifold as discussed in section 1.2. Some of them are manageable, for example, either by using modern sensors which provide denser and more accurate point clouds than before or by incorporating additional data sources such as high-resolution images. However, there is quite a big demand for 3D building models in areas where such modern sensors or additional data sources are not available. Therefore, in this thesis a new fully automatic reconstruction approach of semantic 3D building models for low- and high-density airborne laser scanning (ALS) data of large urban areas is presented and discussed. Additionally, it is shown how automatically derived building knowledge can be used to enhance existing building reconstruction approaches. The specific research objectives are outlined in section 1.3. It includes an overview of the proposed reconstruction workflows and the contribution of this thesis. In order to have lean workflows with good performance, some general assumptions on the buildings to be reconstructed are imposed and explained in section 1.4. The introduction ends with an outline of this thesis in section 1.5.

1.1 Motivation

3D building models are one of the major components of 3D city models[1] (Geibel and Stilla, 2000; Rau and Lin, 2011). They already play for quite some time a central role in urban and regional planning (Danahy, 1999; Wolff and Asche, 2008; Chen, 2011; Czyńska and Rubinowicz, 2014), telecommunication (Siebe and Büning, 1997; Wagen and Rizk, 2003), positioning and navigation (Bradbury et al., 2007; Cappelle et al., 2012, Peyraud et al., 2013; Hsu et al., 2015), tourism (Coors et al., 2000; Glander and Döllner, 2009), and many other domains. Furthermore, they enable in the environmental field precise analyses and simulations of, for example, noise transmission (Czerwinski et al., 2007; Stoter et al., 2008; Ranjbar

[1] 3D city models are according to (Stadler and Kolbe, 2007) digital representations of the Earth's surface and related objects belonging to urban areas. They usually consist of a digital terrain model (DTM), 3D building models, street furniture (e.g., street signs, benches, statues, etc.), vegetation (e.g., trees, bushes, etc.), and other constructions above and below ground (e.g., bridges, tunnels, supply lines, streets, etc.).

et al., 2012), solar potential for large scale implementation of photovoltaics (Strzalka et al., 2012; Eicker et al., 2013), shadow detection and sunlight distribution (Robinson, 2006; Yu et al., 2009), wind flows (Ayhan and Sağlam, 2012; Toja-Silva et al., 2016), safety and security in public spaces (Yaagoubi et al., 2015), and pollutant and flood propagation (Leszek, 2015).

According to (Shiode, 2001), applications of 3D city models can be grouped generally into four categories:

- planning and design,
- infrastructure and facility services,
- commercial sector and marketing,
- promotion and learning of information on cities.

Further possible categorizations and applications of 3D city models are described in (Singh et al., 2013). A comprehensive state-of-the-art review of 3D city model applications is given in (Biljecki et al., 2015). An overview of 3D GIS (Geographic Information System) analyses for 3D city models is presented in (Moser et al., 2010). As can be seen, the application areas are quite diverse and lead at the present time to a rising demand for 3D building models. Particularly in the context of smart cities, the modeling and reconstruction of buildings became a trending topic in recent years.

Several cities such as New York (City of New York, 2017), Toronto (City of Toronto, 2016), Berlin (Döllner et al., 2006; Kada, 2009; City of Berlin, 2017), Singapore, (National Research Foundation, 2017), and Stuttgart (Wolf, 1999) provide nowadays 3D city models that have been reconstructed in a (semi-)automatic way. However, urban areas are continuously changing due to construction, destruction or extension (Morgan and Habib, 2002). Additionally, the change rate of three-dimensional information is often higher than the change rate of two-dimensional data required for traditional maps (Brenner, 2004). Consequently, to provide reliable and up-to-date information, building models need to be updated frequently. For this task, currently only manual or semi-automatic reconstruction methods are reasonably applicable. However, this is, especially for large areas with thousands of buildings or for areas in which many changes occur, up till now very time-consuming and expensive. Thus, the demand for automatic approaches that reduce or even completely eliminate the manual effort is nowadays of particular interest.

In addition to that, new possibilities to reconstruct more detailed building models have arisen in recent years due to the developments of new sensors that provide point clouds with higher accuracy and resolution than before. For this, new methods need to be developed that are able to deal with the increased amount of data and that can process them in a reasonable computational time. Moreover, these methods should be capable of incorporating more comprehensive information in the reconstruction process so that more accurate and detailed building models can be generated.

3D building models reconstructed by an automatic approach with current state-of-the-art methods are, however, not able meet the sophisticated demands of the aforementioned applications. With recent technologies, only standard and relatively simple buildings can be automatically and reliably reconstructed from ALS data. Consequently, many buildings in large 3D city models are represented as block models without any detailed roof structures. Block models provide, however, for many applications only insufficient information. The automatic reconstruction of distinctive roof structures and superstructures on the other hand is still coupled with so far unsolved issues. For example, many approaches are limited to certain roof types or are still struggling with difficulties to provide consistent and topologically correct building models whose roof planes do not unintentionally intersect each other or contain unwanted gaps. The reasons for this are manifold and described in section 1.2. In order to address the unsolved issues, some hybrid reconstruction approaches have been recently proposed that incorporate and combine different types of data sources (e.g., ALS and aerial images). However, additional data sets are not always available and their use increases usually the costs and the complexity of the hybrid approach. Therefore, a fully automatic reconstruction approach needs to be developed that only utilizes ALS data without being too dependent on sensor specifications and that provides 3D building models with detailed roof structures.

Besides the geometric information of buildings, also semantic building information is in demand nowadays from several users such as public authorities (Gröger and Plümer, 2012). In this context, CityGML (Gröger et al., 2008) has been developed to provide an open standardized data model and exchange format for 3D city models that incorporates not only geometric but also semantic information. Note, there exist diverse implementations of CityGML so that there are multiple valid variants of models within each level of detail (LoD) leading to different results in spatial analyses (Biljecki et al., 2016b). For further information about CityGML and its applications see, for example, (Kolbe et al., 2005) and (Kolbe, 2009). Current building reconstruction approaches are, however, essentially only concerned with geometric information. The need for an automatic workflow to derive building models with semantic information is therefore great. The availability of semantic information is of interest not only for many applications but also for improving, for example, the geometry of building models produced by existing building reconstruction processes. Consequently, a robust method to automatically extract semantic building information is needed that is flexible enough to be integrated into existing reconstruction approaches.

1.2 Challenges in 3D Building Reconstruction

The reconstruction of 3D building models is a complex task that requires a workflow comprising of several processing steps such as classification, outline extraction, segmentation, feature recognition, hypothesis generation and verification, geometric modeling and construction, and adjustment and refinement. To increase the productivity of each step, a number of methods have been developed for the automatic processing of ALS point clouds. Despite the acquired knowledge, there is still a significant number of unsolved problems. The primary causes for these problems are outlined in the following paragraphs.

An issue that makes the automatic reconstruction task challenging results from data gaps which are frequently present in ALS point clouds. They are difficult to handle since they can be caused by various reasons. Common reasons are, for example, unwanted reflections of laser pulses on shiny surfaces, absorption of laser pulses by water, or occlusions. The latter one usually occurs close to high building parts or other objects such as high vegetation. A possible effect of data gaps in point clouds is that some structures become unrecognizable so that for the modeling of buildings only insufficient information can be extracted. In the worst case, this can actually cause the whole reconstruction process to fail. But even if sufficient information can be extracted, special care has always to be taken to properly close such data gaps during the reconstruction process in order to avoid gaps in the final model. This task is already challenging in 2D, for example, in the case of precise outline reconstructions of buildings that are surrounded and partly covered by high trees. In such a situation, many assumptions have to be made to avoid jagged shapes and to guarantee topological correctness. In 3D, however, this task becomes even more complex.

Furthermore, building models that best fit the given point cloud in terms of geometry are not necessarily the most accurate models or the best representations of the real buildings because ALS data contain noise and outliers. Although the standard deviation of LiDAR data is nowadays usually less than 0.3 m (Xiong et al., 2014a), noise and outliers must be always considered during the reconstruction process.

Additionally, the resolution of point clouds is always limited, even with modern sensors that can achieve more than 100 points/m². Consequently, sharp features such as break lines cannot be directly extracted from point clouds but must be approximated. Particularly in low-density point clouds, a strong focus on the point coordinates would often lead to a low model accuracy and the missing regularities in the building models would be visually obvious. In contrast, a reconstruction framework with a strong focus on geometric regularities would lose the capability to deal with the high variability of complex building shapes. Finding the right balance between the confidence in data and the confidence in geometric regularities is quite demanding and complicates the reconstruction because it varies from situation to situation (Oude Elberink, 2008).

Another challenge arises from the large amount of different building types in urban areas and the high variability of combining simple building roofs to more complex rooftops. Generally, there are no specific rules to cover the complexity of the structure and shape of buildings so that it is impossible to define a universal reconstruction framework based only on predefined object models. This becomes particularly apparent during the reconstruction of large urban areas that possess a vast number of different building shapes.

For the reconstruction of detailed building models from low-density point clouds, particularly the detection and construction of small roof parts has been repeatedly proven to be very difficult due to the low information content of the data. Therefore, many reconstruction approaches treat points of small roof parts as outliers and discard them from the whole reconstruction process. But by ignoring these outliers they are only capable of providing

building models with few details. There is thus still a great demand for automatic reconstruction workflows that are able to provide detailed building models from low-density point clouds.

The combination of all aforementioned issues makes the fully automatic reconstruction of 3D buildings from ALS point clouds extremely challenging. As a consequence, this topic still continues to be an active area of research.

1.3 Research Objectives

The main research objective of this thesis is the development of an automatic reconstruction approach that provides semantic 3D building models from classified low- and high-density ALS point clouds of large urban areas. Specifically, the objectives of the proposed workflow are defined as follows:

Accuracy: The resulting building models should represent the actual buildings in terms of their geometry as accurately as possible. For this purpose, the process should consider that a strict alignment of the model to the input data is not always the best solution to reflect reality.

Automation: The elimination of mandatory user interaction is especially of great interest for the reconstruction of large urban areas. Therefore, a workflow should not require any user interactions during the reconstruction process.

Complexity: The resulting building models should be adequately represented by a set of points that is significantly smaller than the original point cloud. Otherwise, the advantage for subsequent applications would be low.

Efficiency: The reconstruction approach should be capable of processing large urban areas within a reasonable amount of computational time and with modest hardware resources (e.g., less than one second per building with current computer hardware).

Quality: The reconstruction approach should provide building models of high quality. For this, the completeness and the correctness of the reconstructed building models should be both at least 85 %.

Resolution: The resulting building models should generally be as detailed as possible. This means that for ALS point clouds as input data the proposed method should strive for building models with a level of detail that corresponds to LoD 2, as defined in the CityGML standard (Gröger et al., 2012), with detailed roof structures. However, the workflow should also be capable of incorporating the fact that different applications require different levels of detail. For example, in noise analyses small roof superstructures have often only a small impact on the results. Thus, if such details can be omitted, the subsequent analyses can be performed in less time. For this, the workflow should be designed to allow users to select the LoD of the resulting building models.

Robustness: The general workflow should be applicable to any ALS point cloud regardless of its characteristics (e.g., point density, amount of noise, etc.). Particularly for sparse or noisy point clouds, it is essential that reasonable and conclusive models can still be achieved. For this, occluded areas in the input data need to be adequately closed in the final building model.

Transferability: The reconstruction workflow should be capable of processing data from any city without being limited to certain regions or specific building types.

These are the fundamental objectives of the proposed reconstruction workflow. To reduce the complexity of this task, some general assumptions regarding 3D building models are made in section 1.4.

1.3.1 General Framework

In this subsection, an overview of the developed grammar-guided approach for the automatic reconstruction of semantic 3D building models is presented. Its general workflow is illustrated in Figure 1.1 and described in the following paragraphs.

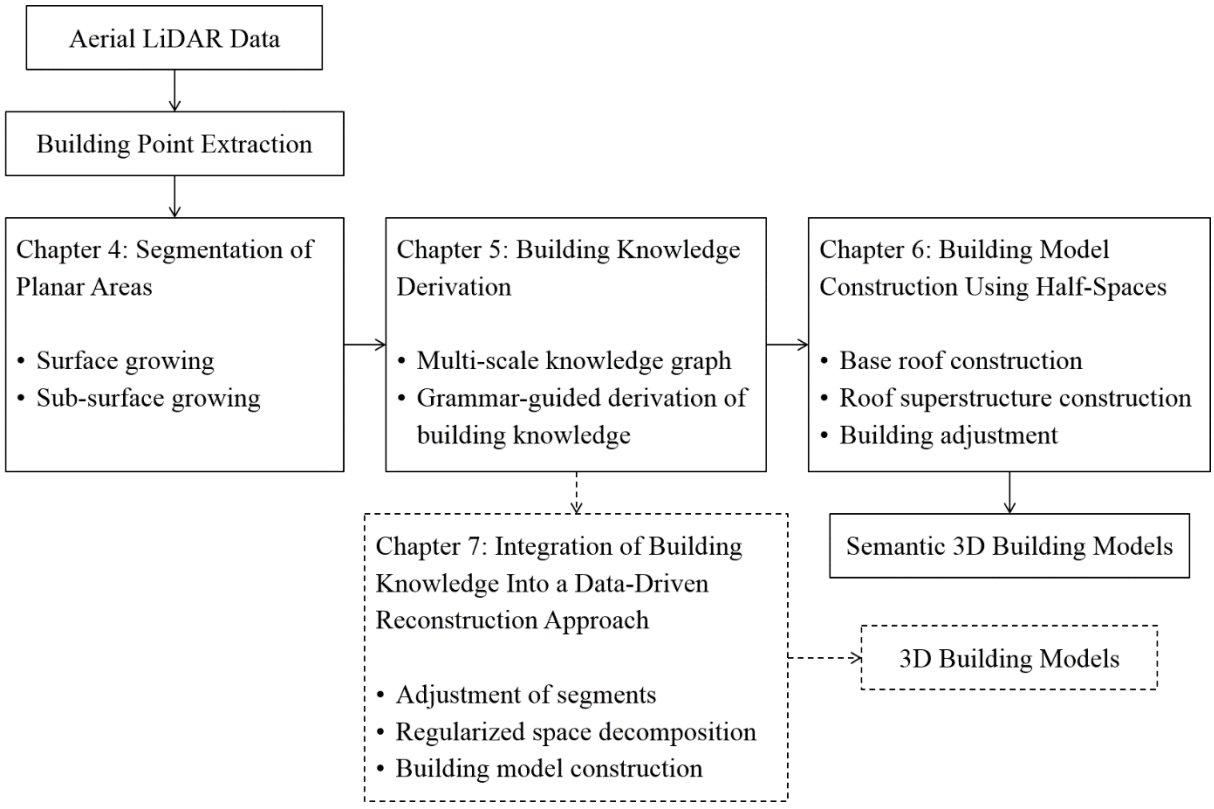


Figure 1.1. General workflow of both the grammar-guided reconstruction approach for semantic 3D building models and the integration of building knowledge into a data-driven reconstruction method to improve the resulting building geometry.

In order to automatically reconstruct semantic 3D building models in large urban areas, a framework has been developed that only requires building points of an aerial LiDAR data set. In literature, the distinction between building points and non-building points is usually

considered and addressed separately from the automatic reconstruction of 3D building models. In this thesis only the automatic reconstruction of 3D building models is covered. For the evaluation of the proposed approach, the building points were determined by using manually digitized ground plans in combination with a ray casting algorithm in order to solve the point-in-polygon problem. Other classification methods are described, for example, in (Brenner, 2010) and (Rottensteiner, 2017).

The proposed grammar-guided building reconstruction approach is a hybrid reconstruction approach that keeps the benefits of data-driven (flexibility and completeness) and model-driven (robustness and visual attractiveness) reconstruction methods. For this, building knowledge is extracted and incorporated during the reconstruction process, on the one hand, to guide the recognition of further building features and, on the other hand, to improve shape regularization of building roof components. The general framework of the proposed approach consists of the following three steps: roof plane segmentation, building knowledge derivation, and building model construction.

In the segmentation step, building points are grouped to a segment if they belong to the same roof surface. The resulting set of segments is crucial for the reconstruction process because as, inter alia, pointed out in (Perera, 2015), the success rate of building reconstruction approaches is in many cases dependent on the segmentation result. Therefore, a new segmentation method has been developed that is specially designed to support the modeling and reconstruction process of buildings. It starts with a surface growing algorithm to find an initial set of planar segments and continues with a sub-surface growing procedure that generates virtual points below real measured surface points. In this way, holes in a segment that are caused by building superstructures or links are automatically closed. Furthermore, small building details are better represented in the final set of segments.

Afterward, modeling cues in form of building features are recognized during the grammar-guided building knowledge derivation step. In order to reduce the search space, the recognition process is initially performed on the segmentation result rather than on the raw point cloud. All building features detected in the set of segments are then organized in a multi-scale knowledge graph introduced in this thesis. Based on already recognized building features, further information about the building is iteratively derived and inserted into a multi-scale knowledge graph. The process of recognizing building features is guided by a graph grammar so that it is possible to support regularities and to avoid unnatural structures in the final building models. Furthermore, the combination of the multi-scale knowledge graph with the graph grammar enables the explicit search for building features that are difficult to identify. Thus, it serves as a good basis for the reconstruction of detailed building models.

In the subsequent building model construction step, the recognized building features of the previous building knowledge derivation step are utilized together with the principles of half-space modeling to construct 3D building models. For the different building features, half-spaces are introduced so that the whole building can be described as a combination of half-spaces. Since small roof superstructures are, particularly in low-density point clouds,

represented only by a small number of points, the recognition of their building features is quite challenging. In contrast to the reconstruction of base roofs, special care has therefore to be taken in the reconstruction process of such small superstructures. In order to meet this challenge, an Iterative Closest Point (ICP) based method is applied in which instances of the same roof superstructure are reconstructed not independently from each other but simultaneously. Thereby, the point density can be locally increased and information of a roof superstructure is automatically incorporated in the construction of another instance. Finally, divisive clustering techniques are applied in a local and global context to the hyperplanes associated with the half-spaces so that frequently occurring regularities in the building models are better supported. The resulting building models are expressed in canonical form and then directly converted into polyhedral models for visualization purposes. If needed, the polyhedral building models can be enriched with semantic information acquired during the building knowledge derivation step.

To show the capabilities of the proposed building knowledge derivation procedure for existing data-driven building reconstruction approaches, a further workflow is presented. It supports natural regularities in buildings and shows how semantic information can be incorporated in an existing data-driven reconstruction method to improve the shapes of the building models without significantly lowering its reconstruction capabilities. In order to demonstrate this, the data-driven reconstruction approach presented in (Sohn et al., 2008), which is based on space partitioning, has been exemplarily selected. In this case, as indicated in Figure 1.1, first the derived building knowledge is used for the adjustment of segments. Afterward, these adjusted segments are used to obtain a regularized decomposition of the space. Finally, polyhedral building models are directly extracted from the regularized space decomposition. Using the building knowledge once again, the number of partitions during the decomposition of space as well as the merging ambiguity of adjacent partitions during the construction of the building model is reduced.

1.3.2 Contributions

The major contributions of the presented research can be summarized as follows:

- A feature-driven reconstruction approach which has to a large extent both the flexibility of data-driven methods and the robustness of model-driven methods. This approach is able to provide not only geometric but also semantic information of the reconstructed building models.
- A segmentation method that is especially designed for the segmentation of roof surfaces. It keeps segment patches of a common roof surface together, preserves small roof details that otherwise would be missed, and automatically closes holes in segments that are caused by roof superstructures or links. In order to accomplish this, virtual points that are located below roof surfaces are introduced and used together with the measured points in the segmentation process.
- A graph grammar that extracts building features automatically and formalizes the description of real rooftops. It is shown how the grammar can be used in combination

with the proposed multi-scale knowledge graph to guide the feature recognition process and to control the shape of the resulting building models.

- A modeling and segmentation method that utilizes half-space modeling techniques to obtain valid building models. It is able to define complex roof shapes by a rather small number of planar half-spaces and guarantees that the resulting models are always closed and without any unintentional gaps.
- An ICP based reconstruction method for the detailed reconstruction of complex rooftops. It is especially designed to detect, model, and construct small repetitive roof superstructures in low-density point clouds and incorporates regularities in a straight-forward way.
- Local and global regularization rules that improve both the shape of the resulting building models and their accuracy. The rules are defined for half-spaces and describe how their hyperplanes are to be adjusted. In order to adjust more than one hyperplane at a time, a divisive clustering algorithm is applied.
- Integration of building knowledge into a data-driven reconstruction approach that improves the regularities of the resulting models without lessening the flexibility to generate any roof shapes occurring in the real world. Additionally, for the data-driven reconstruction approach exemplarily selected in this thesis, hyperpolylines are introduced to reduce the number of partitions during the binary partitioning of the space.

1.4 Restrictions on 3D Building Models

All reconstructed building models in this thesis are 2.5D models since only ALS point cloud data are utilized, which mainly represent 2.5D point information. This means, only a single elevation value of the building roof is accepted for any given x-y location. This elevation value corresponds to the highest elevation of the object at the given x-y location. The only exceptions where more than one elevation value for a given x-y location is accepted are those locations that belong to the vertical face of a step edge. Thus, objects located under a roof are not considered in the reconstruction process and facades are represented as vertical walls without any details.

Furthermore, it is assumed that surfaces of a building can be described with sufficient accuracy by planar faces. This is for most buildings, particularly in residential areas, the case and useful because arbitrary shapes usually require computationally intensive surface fitting procedures (Wang, 2013).

Both restrictions are common for most automatic reconstruction methods using data acquired by traditional airborne sensors; see, for example, (Dorninger and Pfeifer, 2008), (Sampath and Shan, 2010), (Zhang et al., 2011), (Sohn et al., 2013), (Perera and Maas, 2014), and (Xiong et al., 2014b).

1.5 Outline of the Thesis

This thesis is organized in nine chapters that are briefly described in this section. After this introductory chapter 1, it is structured as follows:

In chapter 2, literature related to the research topic of this thesis is reviewed. It provides an overview of different approaches and techniques concerned with the automatic reconstruction of 3D building models from aerial LiDAR data, the reconstruction of small roof superstructures, and the procedural modeling of buildings.

In chapter 3, relevant fundamentals are summarized. This includes a description of commonly applied segmentation methods, the principles of grammars, different possibilities to model 3D objects, data clustering techniques, and point cloud registration techniques.

In chapter 4, details of the developed procedure to determine planar roof areas from a given set of building points and its benefits are described. It is based on a surface growing algorithm which is modified so that segments are able to grow also below other segments. This modification improves the procedure significantly in many aspects.

In chapter 5, a multi-scale knowledge graph is introduced to organize and manage meaningful building features. Furthermore, it is explained how building features can be automatically extracted by applying the proposed grammar-guided building knowledge derivation procedure.

In chapter 6, a half-space modeling method is utilized for the construction of 3D building models and diverse clustering methods are applied for their adjustment. With these proposed methods, it is guaranteed that the resulting building models can be converted into watertight boundary models. Furthermore, it is explained how small roof superstructures can be reconstructed in low-density point clouds.

In chapter 7, an automatic workflow is presented that integrates automatically derived building knowledge into an existing purely data-driven reconstruction approach to improve the regularities of the resulting building models.

In chapter 8, experimental results of the proposed grammar-guided building reconstruction approach are presented and discussed. For this, different data sets from different locations are used to prove the applicability of the proposed approach.

In chapter 9, the main conclusions of this thesis are drawn and recommendations for future research are given.

Note that over the course of research, parts of this thesis have been presented to renowned international scientists and have already been published. The publications of the author that have been partly included in this thesis or that are related to the following chapters are: (Kada

and Wichmann, 2012), (Kada and Wichmann, 2013), (Wichmann and Kada 2014), (Wichmann et al., 2015), (Jung et al., 2016), and (Wichmann and Kada, 2016).

1. Introduction

2. Related Work

Since several years 3D city models assume a central role in a vast number of applications. Their varied application areas lead at the present time to an increased demand. The challenge in the reconstruction of 3D city models is in particular the large number of different and complex building forms. Numerous building reconstruction approaches have therefore been proposed in the last two decades but the subject still remains as a very active area of research in various scientific disciplines. In this chapter, an overview of existing building reconstruction approaches is given. Reviews of further reconstruction approaches are presented in (Brenner, 2005), (Haala and Kada, 2010), (Brenner, 2010) and (Wang, 2013).

In general, there are several possible ways to categorize existing 3D building reconstruction approaches. Typically, building reconstruction approaches are classified according to the criteria stated, for example, in (Brenner, 2010):

- amount of human interaction (manually, semi-automatic, fully automatic)
- data sources (point clouds, images, multi-sources)
- reconstructed models (geometric, topologic, semantic description)
- control (model-driven (top-down), data-driven (bottom-up), hybrid methods)

With respect to the control criteria, some model- and data-driven building reconstruction approaches are summarized in section 2.1 and 2.2, respectively. For a direct comparison between model- and data-driven reconstruction approaches see also, for example, (Tarsha-Kurdi et al., 2007b).

Lately, data-driven and model-driven reconstruction approaches have been merged towards hybrid reconstruction approaches that try to exploit the advantages of both worlds: the shape flexibility of data-driven approaches with the shape regularization capabilities of model-driven approaches. In (Satari et al., 2012), for example, a data-driven method is used for the reconstruction of the main roof and a model-driven approach is utilized in order to add further details such as dormers. The developed reconstruction approach in this thesis can also be classified as a hybrid approach because automatically derived building knowledge is extensively incorporated during the bottom-up reconstruction.

A major challenge for most reconstruction approaches is the automatic reconstruction of small roof superstructures such as dormers and chimneys. During the last couple of years, different strategies have been therefore developed to solve this challenge. An overview of building reconstruction approaches with a particular focus on this task is given in section 2.3.

In addition to the reconstruction of existing buildings, the automatic generation of virtual urban areas has become important as well. The virtual areas are used, for example, in movies

and games, on the one hand, to keep the production and development costs low and, on the other hand, for safety reasons. For this purpose, several approaches have been developed that utilize different procedural modeling techniques. These techniques are also for reconstruction approaches steadily gaining in importance. Therefore, an overview of so far developed procedural building modeling approaches and the utilization of their principles in reconstruction approaches are finally given in section 2.4.

2.1 Model-Driven Building Reconstruction Approaches

In model-driven approaches, building templates are chosen from a predefined catalog and then adapted by their parameters to best fit their roof shapes to the given data. This is possible due to the fact that a great number of buildings in urban areas are rather simple and feature uniform shapes. In order to construct more complex roof structures and to keep the number of building templates in the catalog low, building templates are combined. In this way, building shapes can be constructed that are not explicitly defined in the template catalog. As stated, for example, in (Dorninger and Pfeifer, 2008), an advantage of model-driven approaches is that they always generate topologically correct building models if the catalog and the combination of its templates are well-defined. Furthermore, the parameterization of the building templates ensures that the inherently strong regularization of shape templates is implicitly passed on to the reconstructed building models so that they are always well-shaped. Constraints such as parallelism and orthogonality in the resulting building geometry are thus already provided by the parametric models so that further adjustments become obsolete. The robustness of model-driven approaches is of particular advantage if the information content of the input data is low (e.g., in low-density point clouds). The resulting building models are well-suited for visualization purposes that do not require a high LoD. A major limitation of model-driven approaches is, however, that the resulting building shapes are always limited to a predefined catalog because only those building shapes can be constructed that are either included in the catalog or that can be composed of its templates. Consequently, some buildings can be only crudely approximated by model-driven approaches. Furthermore, catalogs in model-driven approaches are at risk to become quickly extensive if a large urban area with a large number of different building shapes needs to be reconstructed. Some model-driven reconstruction approaches are summarized in the following paragraphs.

In (Verma et al., 2006), a model-driven reconstruction approach is proposed that consists of the following three major components: (i) extraction of building points, (ii) roof topology inference, and (iii) parametric shape fitting. During the extraction of building points, first all those points are discarded from a given point cloud that are not located on a locally flat and non-vertical surface. Then, the remaining points are grouped based on their spatial proximity to each other and the points of the largest connected component, which is expected to represent the ground, is discarded. Further methods are finally carried out to discard also all points of small connected components that represent, for example, cars, low vegetation, mailboxes, etc. For the subsequent roof topology inference, local planar patches are fitted to the identified building points using the ball-pivoting algorithm; see (Medeiros et al., 2003) for further details. Based on the set of planar patches, a roof topology graph (RTG) is constructed

in which each planar patch is represented by a vertex and the adjacency of two patches by an edge. Edges are labeled depending on the normal vectors of the planar patches: orthogonal in the x-y plane and pointing away from each other, orthogonal in the x-y plane and pointing towards each other, parallel in the x-y plane and pointing away from each other, and no constraint. Using a predefined set of roof topology graphs of which each represents a simple building component, an iterative subgraph matching procedure is performed to recognize their occurrences in the constructed RTG. In order to avoid ambiguities during the subgraph matching process, subgraphs are recognized in decreasing order of complexity and vertices are not allowed to be part of more than one matched subgraph. After all occurring building components are identified, a building geometry fitting procedure is carried out in which first initial shapes are estimated and afterward refined to minimize the square error between the roof model and the building points based on an energy formulation. Further approaches based on RTGs that incorporate an extended catalog of predefined subgraphs are, for example, presented in (Oude Elberink and Vosselman, 2009), (Perera and Maas, 2014), (Xiong et al., 2015), (Xu et al., 2015), and (Jarząbek-Rychard and Borkowski, 2016). In order to correct an RTG that has been derived from an erroneous set of planar segments, a graph edit dictionary for correcting errors in an RTG is presented in (Xiong et al., 2014b). Although, many RTG based approaches achieve good reconstruction results, they are always at risk to become time-consuming if a catalog with a large number of predefined subgraphs is required.

In (Kada and McKinley, 2009), building models are reconstructed based on a cell decomposition approach from ALS data and existing ground plans. For this, each ground plan is decomposed into non-overlapping cells of preferably rhomboid or trapezoid shape. In order to keep the number of cells low and to avoid small cells, all building outlines are generalized according to the method presented in (Kada, 2007) while the proposed cell decomposition procedure is applied. For the resulting cell decomposition of a footprint, parameterized roof shapes of all cells are determined. This is realized for each cell by examining the normal vectors of all points (estimated based on their local neighborhood) that are horizontally located inside the cell and by comparing their direction with a predefined set of roof shapes. The set of predefined shapes consists of three different types: basic, connecting and manual. In the basic class, basic roof shapes are defined such as flat, shed, gabled, hipped, and Berlin roof shape. In order to connect two basic shapes, further shapes that represent specific junctions are defined in the connecting class. For the efficient calculation of which shape best represents the points of a cell, all basic and connecting shapes are split into eight sections and each section is labeled according to its major orientation as follows: up, north, east, south, and west. Some examples are shown in Figure 2.1. The shape template for each cell is determined by distributing its points to the eight sections and by evaluating the number of corresponding directions. The shape template with the highest number of corresponding directions is finally selected for a cell. Those cells, however, whose number of corresponding directions is too low are assigned to the shape type manual so that they need to be afterward constructed in a manual way. The proposed method has been proven to be suitable for the reconstruction of large urban areas. However, it still requires manual post-processing because the number of combinations that feature different orientations in the eight sections is severely limited and building details are thus discarded in this approach. Further problems occur if a major

orientation in a section is not present in the input data. This is, for example, particularly often the case for buildings with many roof superstructures.

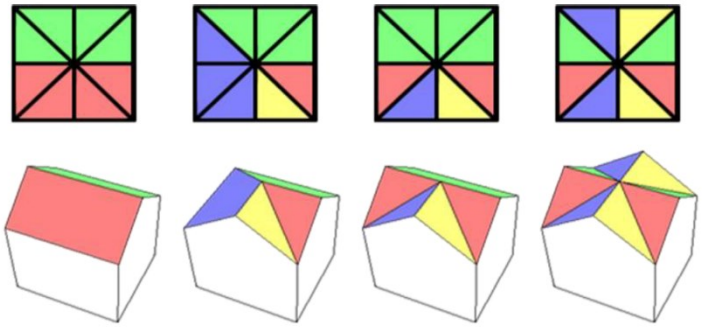


Figure 2.1. Predefined shape templates and the orientation of their sections (Kada and McKinley, 2009).

Similarly in (Lafarge et al., 2010), 2D building outlines are first decomposed into sets of connected, non-overlapping quadrilaterals with common edges. Then, a 3D building shape is placed in each quadrilateral from a set of predefined shape templates. In contrast to (Kada and McKinley, 2009), the selection of the appropriate shape template is not based on the major orientation of a further subdivision but it makes use of a Gibbs model to control the block assemblage and the fitting to data. In order to determine the optimal configuration of selected shape templates, a Bayesian framework using a Markov Chain Monte Carlo sampler is applied. The approach has been tested on satellite and aerial digital surface models (DSM) with different resolutions (0.7 m, and 0.1 m and 0.25 m, respectively). The altimetric evaluation of the reconstructed buildings has been improved for satellite and aerial DSM as well by an interactive decomposition of the building outlines from 2.3 m and 1.3 m to 1.1 m and 0.8 m, respectively. To achieve higher precision and to decrease the computational time, it is suggested in (Lafarge et al., 2010) to improve the optimization step by using, for example, belief propagation techniques for graphical models or Jump-Diffusion processes.

In (Kwak and Habib, 2014), first data-driven methods are applied to determine approximated building boundaries using the modified convex hull algorithm presented in (Sampath and Shan, 2007). These building outlines are then regularized and decomposed into rectangular shapes based on a recursive minimum bounding rectangle process. Afterward, an initial model from a predefined catalog is determined whose parameters are then refined through least-squares adjustment.

In (Huang et al., 2013), a pure top-down statistical scheme is proposed in which generative modeling is conducted based on a predefined primitive library to reconstruct building roof models. In order to keep the computational time low, the library only consists of three groups including eleven types of parametric roof primitives. More complex buildings are represented by an assembly of primitives that are, in contrast to many other reconstruction approaches, allowed to overlap each other. The selection process of roof primitives and their parameters is conducted based on reversible jump Markov chain Monte Carlo (RJMCMC) techniques with

specified jump mechanism. Although the proposed method has shown its capabilities on different data sets, some issues of uncertainty and instability still remain. Furthermore, the impact of the scene complexity and the prior knowledge on the completeness of the reconstructed models as well as the long computation time needs to be further improved for the reconstruction of large urban areas.

Another model-driven reconstruction approach for large urban areas based on building footprints and with focus on sparse LiDAR point clouds (1 point/m²) is presented in (Henn et al., 2013). The model catalog defined in this approach consists of ten basic primitives that are, analogous to (Kada and McKinley, 2009), fitted to a decomposed footprint. Due to the low number of input points and the low number of required points that are needed for model construction, several hypotheses are first generated using an enhanced version of RANSAC (Random Sample Consensus). The hypotheses represent a preselection of the best fitting building models from a predefined set of building templates. In order to select for each decomposition the most probable building model from the preselection, supervised machine leaning methods in form of support vector machine (SVM) techniques are applied. This enables, in contrast to other state-of-the-art methods such as minimum description length (MDL), the incorporation of further information (e.g., roof inclination, median point cloud height, etc.) during the classification process. The results of the supervised classification indicate a high accuracy and a good classification quality. However, shed roofs are often not well identified due to their small number of points available for model estimation. Furthermore, in some cases, a single gable roof is recognized as two shed roofs that represent one plane of its roof surface.

An approach that automatically fits parameterized roof shapes in 3D point clouds based on sequential importance sampling is proposed in (Nguatem et al., 2013). For this, the parameterized roof shapes presented in (Poullis and You, 2009) have been extended. The exploration of the search space for roof types is realized by using a likelihood model similar to (Henn et al., 2013) within a Monte Carlo setting similar to (Huang et al., 2013). But in contrast to many Monte Carlo based procedures, the proposed procedure comprises an inherent data parallelism. Their experiments reveal that the proposed approach is well suited to reconstruct simple roof shapes such as gable, hipped, pinnacle, and mansard roofs. A limitation of the approach is, however, that it can be only applied to roof shapes that feature a quadrilateral footprint. More complex roof shapes or combinations of them that feature an arbitrary footprint have not been tested. Furthermore, small roof superstructures are treated as noise and are therefore not reconstructed.

2.2 Data-Driven Building Reconstruction Approaches

In contrast to model-driven reconstruction approaches, data-driven approaches are not restricted to a predefined set of building shapes or limited by any restrictions but they resemble very closely the input data. Thereby, they are generally able to construct any building shape and they are thus well-suited for large areas in which many individual building shapes occur. A typical workflow of data-driven reconstruction methods consists of the

following three steps: (i) Building points are aggregated to planar segments which represent roof faces. For this, several segmentation methods have been developed and adapted during the last decades that improve the partitioning of a point cloud into planar regions so that under- and over-segmentation errors are decreased. An overview of segmentation methods in the context of data-driven building reconstruction approaches is given in subsection 2.2.1. A methodological summary of three of the most frequently applied segmentation method is later presented in section 3.1. (ii) The resulting segments are then combined to extract building modeling cues (e.g., intersection and step lines, building outlines, simple surface primitives, etc.). For this, additional data sources such as existing building outlines or images are sometimes incorporated as well. Some extraction methods for modeling cues based on a set of planar segments are summarized in subsection 2.2.2. (iii) Finally, 3D building models are constructed based on the extracted building modeling cues. Note, in order to improve the final shape of the building model, many data-driven reconstruction approaches subsequently apply regularization operations. The adjustment of a building model as part of the previously described workflow or as a subsequent step is particularly important for data-driven methods because the model or parts thereof can otherwise easily end up distorted or they can exhibit irregularities. An overview of different 3D building construction and regularization methods is given in subsection 2.2.3. A methodological summary of the applied modeling techniques during the construction is presented in section 3.3.

2.2.1 Segmentation of Planar Surfaces

A common segmentation method for the detection of planar areas is the 3D Hough transform. It is an extension of the well-known 2D Hough transform which has been initially introduced in (Hough, 1962) for the segmentation of lines and which is frequently used in image analysis, computer vision, and digital image processing for the extraction of features. For further details about the principles of 3D Hough transform, see section 3.1. Early approaches in which 3D Hough transform is utilized for the automatic reconstruction of buildings from point clouds are presented in (Vosselman, 1999) and (Maas and Vosselman, 1999). Here, planes in the object space are directly derived from the Hough space by taking the number of intersecting sinusoidal surfaces in a bin into account. Afterward, a connected component analysis (CCA) based on the Delaunay triangulation is applied to derive connected components of the points in the same plane. This is generally necessary because the segmentation problem is approached with the 3D Hough transform in a global way. In order to overcome over-segmentation problems, a subsequent refinement process is proposed in (Overby et al., 2004). Here, segments with a similar plane normal and a small perpendicular distance to each other are afterward merged if the distance variance of the points to the resulting plane is less than a predefined threshold. In addition to a merging step, a splitting process is proposed in (Vosselman and Dijkman, 2001) to face the problem of under-segmentation as well. In the splitting step, segments are split with the intention that only one planar area is finally left per segment. For this, edges of existing ground plans and intersection and height jumps between adjacent segments are elongated and used for the segment splitting process. Afterward, all adjacent segments that represent the same planar area are merged in the merging step. In (Huang and Brenner, 2011), an enhanced version of the 3D Hough transform is proposed for

the segmentation of planar areas. Instead of selecting one particular plane and removing its associated points from the Hough space in each iteration, all clusters are determined whose bins in the Hough space feature a high concentration of surfaces. Thereby, the updating of the Hough space after each iteration is avoided and the processing time is reduced. Furthermore, points may belong in this implementation to more than one plane, which supports the detection of smaller segments. After determining the clusters, planes are not derived by averaging all planes in the same cluster but in a top-down way by a rule-based estimation process which supports parallel and orthogonal structures in the resulting set of segments. The boundaries of the final segments are then determined by estimating ridges (based on plane intersections) and eaves (based on an edge sweeping procedure) for each plane. Further building reconstruction approaches based on Hough transform segmentation methods are, for example, presented in (Oda et al., 2004), (Novacheva, 2008), (Sohn et al., 2008), and (Maltezos and Ioannidis, 2016).

Another frequently applied method for the segmentation of planar roof areas is RANSAC. Analogous to Hough transform, the segmentation problem is approached in a global way. A comparison between Hough transform and a RANSAC based algorithm for the automatic segmentation of planar areas from point clouds is presented in (Tarsha-Kurdi et al., 2007a). Early reconstruction approaches that utilize RANSAC in the context of building reconstruction are presented in (Ameri and Fritsch, 2000) and (Brenner, 2000). In the former approach, a region growing segmentation method is first applied to partition an image into 2D regions. Afterward, all extreme point outliers are discarded and an initial plane is estimated for each 2D region using RANSAC. The initial plane is then refined by considering the estimated error variance in an iterative reweighting M-estimator algorithm (Huber, 1981; Hampel et al., 1986). In the second approach, planar regions are derived from a standard RANSAC algorithm and labeled according to their orientation and the edge orientations of an existing building outline. Then, based on the sequence of labeled regions along an edge of the building outline and a predefined set of transformation rules, segments are either merged or discarded. A general drawback of this approach is that all segments that are not adjacent to an edge of the boundary cannot be labeled according to the proposed concept and are therefore discarded. In (Forlani et al., 2003; Forlani et al., 2006), RANSAC is applied to correct an initial set of segments, which is obtained by clustering the building points of each building according to their gradient orientation. For this, RANSAC is applied in each cluster to split segments that include several adjacent roof segments with slightly different orientations or slopes. An extended RANSAC variant for the detection of planar roof areas is presented in (Tarsha-Kurdi et al., 2008) that often results in improved planar segments. For this, not only the number of points close to a plane candidate is taken into account as condition during the plane detection validation but also the standard deviation of the point distances to the estimated plane. Afterward, points of a segment that belong to other roof planes are eliminated from the segment and remain unassigned if they are not inside another detected plane to which they can be assigned by mathematical morphology procedures. A further variant of RANSAC based on a grid structure is applied in (Chen et al., 2012). Here, point normal directions, point distances and standard deviations to estimated planes are incorporated to guarantee topological consistency between the rooftop segments.

A third common method for the segmentation of planar roof areas from point clouds is surface growing. In contrast to Hough transform and RANSAC, surface growing is a segmentation method which only considers a local neighborhood. Thus, segments are usually derived that each represent one roof face so that the subsequent determination of connected components becomes obsolete. In (Alharty and Bethel, 2004), surface growing is applied to a regularized raster of building points to construct planar segments by a cell aggregation technique. Instead of a single seed point, a group of adjacent cells is selected whose points feature a low RMSE (Root Mean Square Error) to a plane that can be directly derived from them. Then, further adjacent cells are assigned to the group of seed cells if they feature similar properties in terms of the slope in x-direction, the slope in y-direction, and the height intercept. In (Rottensteiner, 2003) and (Rottensteiner et al., 2005), surface growing is applied on a DSM. Here, segments are initialized with homogeneous regions of connected points whose normal vectors feature a similar direction and further neighboring points are added if they are close to the initial plane of the segment. With particular focus on high resolution data sets, an adapted surface growing method is presented in (Dorninger and Nothegger, 2007). In order to reduce the computational time to determine appropriate seed clusters, a hierarchical clustering of local planes is carried out in a 4D feature space. Afterward, neighboring points are added to a seed cluster if they meet the distance and the normal vector direction criteria of the growing segment. In (Dorninger and Pfeifer, 2008) planar segments are derived by making use of surface growing and the mean shift based algorithm presented in (Melzer, 2007). Further surface growing based segmentation methods in the context of 3D building reconstruction are, for example, used in (Park, et al., 2006), (Zhou and Neumann, 2008), (Zhou and Neumann, 2012), (Sun and Salvaggio, 2013), (Abdullah et al., 2014), and (Awrangjeb and Fraser, 2014).

Further segmentation methods have been proposed in the context of data-driven building reconstruction approaches. In (Kim and Shan, 2011), for example, planar roof segments are determined by minimizing an energy function formulated as a multiphase level set; for further information about level set methods that describe surfaces, see (Osher and Sethian, 1988). A fuzzy k-means method based on the surface normal of points that are derived from a planar Voronoi neighborhood is carried out in (Sampath and Shan, 2010). A new initialization method for the k-means algorithm is proposed in (Kong et al., 2013) that determines the number and coordinates of the initial clusters by applying mathematical morphology and Hough transform techniques on an elevation image of the building rooftop to be segmented. Two robust statistical approaches (i.e., DetRD-PCA and DetRPCA) that utilize the deterministic minimum covariance determinant estimator and robust PCA (Principle Component Analysis) are introduced in (Nurunnabi et al., 2014). Two parameter-domain clustering approaches for the segmentation of planar and linear/cylindrical features are presented in (Lari and Habib, 2014). A segmentation method based on a Gaussian mixture model is presented in (Xiao et al., 2015). An optimization approach to further improve the resulting segments from an automatic segmentation process is proposed in (Yan et al., 2014). It is based on a global energy function consisting of the number of planes, the spatial smoothness between points, and the point distances to the initial planes. Two octree based segmentation methods are introduced in (Vo et al., 2015) and (Su et al., 2016).

2.2.2 Modeling Cue Extraction

After a set of segments has been determined, modeling cues can be extracted for the subsequent construction of 3D building models. For this, intersection lines, step lines, and building outlines are frequently extracted. Although building outlines can be considered as a special type of step lines, they are often determined in a separate processing step. There are several approaches that focus, on the one hand, on the generation of a rough building outline and, on the other hand, on their simplification or regularization. These approaches are, for example, based on RANSAC (Neidhart and Sester, 2008; Jarzabek-Rychard, 2012), alpha shape (Dorninger and Pfeifer, 2008; Albers et al., 2016), graphs with shortest path search (Neidhart and Sester, 2008), structured grids (Zhou and Neumann, 2008; Sun and Salvaggio, 2013), or line simplification methods (Neidhart and Sester, 2008) such as Douglas-Peucker (Douglas and Peucker, 1973).

For the detection of step edges, height discontinuities between adjacent segments are searched in (Vosselman, 1999). In (Rottensteiner et al., 2005), step edges are determined based on statistical tests and robust estimation. A so-called compass line filter (CLF) is proposed in (Sohn et al., 2008) which determines the local edge orientation for each step edge. In contrast to the development of special procedures for the extraction of modeling cues that are based on height discontinuities, intersection lines are usually directly obtained by the intersection of two planes that are derived from a pair of adjacent segments.

Other modeling cues are, for example, extracted in (Overby et al., 2004). Here, segments are enlarged to roof faces by intersecting all segments, regardless of their adjacency to each other, and vertical wall planes generated from the building outline with each other. The vertical wall planes are, however, only incorporated in cases where their resulting intersection line is within a predefined distance to the point cloud and if the resulting face is located inside the building outline. An approach to detect modeling cues in form of ridge lines utilizing RANSAC is presented in (Fan et al., 2014).

2.2.3 Model Construction

The extracted building modeling cues, including inter alia a set of segments, their boundaries, and a set of intersection and step lines, are then used for the automatic construction of 3D building models. In many data-driven reconstruction methods (e.g., (Rottensteiner et al., 2005), (Park et al., 2006), (Novacheva, 2008), (Dorninger and Pfeifer, 2008), (Zhou and Neumann, 2008), and (Xiao et al., 2015)), a 3D polyhedral model is directly generated based on the modeling cues. For this, the extracted lines are extended and connected with each other so that each roof surface is bounded by a closed sequence of connected line segments. Further faces with a vertical orientation are then added at step edges so that each edge in the polyhedral model becomes part of two surface defining line segment sequences. The implementation of such a direct polyhedral model generation method often comes, however, with quite a few problems because there are usually some ambiguities how the line segments are to be connected to guaranty the planarity of each polyhedral face without any gaps in between.

Another common approach for the construction of a polyhedral model includes the generation of a mesh model. In (Rau and Lin, 2011), for example, a construction method based on TIN (Triangular Irregular Network) merging and reshaping is applied. The construction of a polyhedral model based on a mesh model resulting from the intersection of all detected roof and wall planes is presented in (Overby et al., 2004). It includes a filtering of faces, a removing of degenerated parts, and a filling of holes, which might result from the previous step, utilizing the filling holes method described in (Barequet and Kumar, 1997). A construction method that is based on a mesh model resulting from a given set of building points is presented in (Zhou and Neumann, 2010). It can be generally applied on a set of segments and guarantees watertight polyhedral building models, even for buildings with arbitrarily shape. It extends the classical dual contouring as proposed in (Ju et al., 2002) into a 2.5D approach so that a simultaneous optimization of the surfaces and the roof layer boundaries is achieved while keeping connecting wall faces vertical. The topological precision in the proposed 2.5D contouring approach is, however, not guaranteed so that distorted building outlines might be generated or roof pieces might be missing. In order to overcome this drawback, an extended 2.5D contouring approach with topology control is presented in (Zhou and Neumann, 2011).

A reconstruction approach based on binary space partitioning, in which building modeling cues are used to partition the space into homogeneous convex polygons, is presented in (Sohn et al., 2008). Due to its relevance to this thesis, a description of this approach with more details is given in section 7.1.

Since building models derived from data-driven reconstruction approaches resemble very closely the input data, several regularization and optimization methods have been developed. For this, most of the above described reconstruction methods incorporate the main orientation of the building and support orthogonal and parallel structures. An adjustment model that considers the building topology by Gestalt observations to improve the shape of a building model is, for example, proposed in (Rottensteiner, 2006). In (Zhou and Neumann, 2012), global regularities are incorporated during the construction. This includes orientation and placement regularities between two roof surfaces, parallelism and orthogonality between building outlines and the normal of their owner planes, and regularities between two boundary edges in terms of their height and position. In order to automatically determine independent and consistent constraints, a greedy algorithm is proposed in (Pohl et al., 2013) while Gröbner bases are utilized in (Meidow and Hammer, 2016). In (Sohn et al., 2013) and (Jarzabek-Rychard and Maas, 2017), a refinement of building models reconstructed in a data-driven way is proposed based on aerial images from which building edges are detected and incorporated in the reconstruction process. An implicit regularization process in the framework of MDL in combination with hypothesize and test (HAT) is, for example, proposed in (Jung et al., 2017).

2.3 Reconstruction of Roof Superstructures

The automation rate of building reconstruction methods has nowadays increased so that many buildings can be automatically reconstructed from airborne point cloud data. Depending on the building shape complexity and the data quality (e.g., density, accuracy, completeness, etc.), modern reconstruction approaches are sometimes even able to generate realistic looking 3D building models with roof and facade structures. However, problems with existing automatic reconstruction approaches often occur if the given input data does not meet the desired or required quality. In such a case, only coarse and generalized models without details can be created. Measured points from roof superstructures (e.g., chimney, dormers, etc.) are thereby usually considered as noise and are therefore ignored during the reconstruction process. Several approaches have been developed that offer users the possibility to add them in a semi-automatic way or that utilize additional data sources like images (Rottensteiner and Briese, 2003; Habib et al., 2010). But most of them are not applicable in practice for a fully automatic large-scale reconstruction process due, for example, to a limited amount of available reconstruction time or to missing additional data sources.

Up to now, not much research deals explicitly with the automatic reconstruction of smaller roof objects. One reason might be the fast development of sensors that provide nowadays denser and more accurate point clouds than before. However, if an urban area from a certain time needs to be reconstructed, these new sensors cannot be utilized. Therefore, there will be always a demand for automatic reconstruction approaches that are able to handle and to reconstruct as many details as possible from low-density point clouds.

A simple reconstruction approach that tackles this problem for flat superstructures on flat building roofs is described in (Stilla and Jurkiewicz, 1999). It creates a histogram and makes use of the peaks to segment the height data and to recognize areas of possible superstructures. Minor peaks with a certain gap to the main and to other minor peaks are then examined regarding their extent and compactness. Afterward, points of an accepted minor peak are approximated by a prismatic object. This approach is usually robust but in practice only applicable to a small number of building roofs in an urban area and strongly depends on predefined parameters.

Therefore, most research effort has been put to the following strategy: first, an initial model that consists of the main building components is reconstructed and then the best fitted parametric shapes from a predefined library are added during the (model-driven) reconstruction of smaller roof objects. An example is given in (Brédif et al., 2007), in which the problems of reconstructing smaller objects on rooftops are discussed and which can be extended to point clouds. It presents a parametric roof superstructure reconstruction that makes use of an MDL energy minimization technique. Further examples that define different dormer shapes as superstructures, which are then placed on top of a base model, are presented in (Milde and Brenner, 2009) and (Vosselman and Dijkman, 2001). These model-driven approaches are generally better suited than data-driven methods. But on the one hand, they fail if the point density is too low and on the other hand the reconstruction capability is always

limited to a predefined library. To weaken the latter disadvantage, the number of shapes in the predefined library can be extended by causing an increased computational time.

To reduce the impact of the exhaustive search on the computational time, some approaches integrate a coarse superstructure detection step to detect the shape type before a shape refinement step is carried out. For example, in (Satari, 2012) and (Satari et al., 2012) a support vector machine is used for the recognition of three predefined dormer types. It utilizes the gradient and azimuth values of the normal vectors of the superstructure and the underlying roof plane as discriminating factors. Afterward, the initial superstructure model is refined and added. Another example for a coarse superstructure detection step is given in (Dornaika and Brédif, 2008).

For point clouds with a very high density, a generative statistical top-down approach for the reconstruction of superstructures is presented in (Huang et al., 2011). It searches for superstructures in the area above a base roof by using two simple parametric primitives (flat and gable roofs). The success of the method is, however, highly dependent on the point density.

Summarizing the previous research, it deals mainly with model-driven approaches because they are usually more stable for sparse point clouds than data-driven approaches. To overcome the computational time caused by an exhaustive search technique, a coarse detection of the superstructure type can be performed. However, a more or less high density is required to get an accurate model because superstructures are independently reconstructed from each other.

2.4 Procedural Building Modeling

Since several years, procedural modeling techniques have been utilized to generate large urban environments; an overview is, for example, given in (Smelik et al., 2014). A well-known technique is the Lindenmayer-System (L-system) which has been successfully adapted to generate realistic street networks and buildings with simple shapes. In (Parish and Müller, 2001), for example, a system called CityEngine is proposed for the procedural modeling of cities based on L-systems. The presented system mainly consists of the following three steps: Firstly, a street network is generated that divides an area according to image maps and their proposed L-system into smaller blocks so that each block is surrounded by streets. The blocks are then further subdivided to define the allotments for the placement of individual buildings. For this, a recursive algorithm, which considers that most buildings feature a convex and rectangular outline, divides the block areas into smaller lots until each lot covers an area under a predefined threshold. Finally, a stochastic, parametric L-system is used to generate one building for each lot that has both direct access to a street and a proper area. For this, lots are utilized as building outlines and further modified by the L-system. The proposed L-system consists, on the one hand, of transformation, extrusion, branching, and termination modules and includes on the other hand geometric templates for different building components. Depending on the zone in which a lot is located, different sets of production rules are applied that describe the style of either skyscrapers, commercial buildings, or residential

houses. Thus, the resulting building models basically consist of simple scaled and translated boxes on which facade details are added with a shader. The generation of geometric facade details, however, has not been addressed. A similar strategy, strongly influenced by CityEngine, is also presented in the interactive city editor CityGen (Kelly and McCabe, 2007).

With the objective to generate complex city models in reasonable time, a modification of a context-free L-system has been presented in (Marvie et al., 2005). In the so-called FL-system (Functional Lindenmayer-System), terminal symbols are replaced by functions that can be executed during the rewriting process to instantiate or modify generic objects on the fly. By extending the L-system language, these generic objects can be used as rule parameters. Another amendment concerns the natural parallelism of a rewriting process in L-systems. In order to provide the possibility of controlling the parallel rewriting process, a synchronization operator has been introduced, which cuts the parallel derivation and first completes the rewriting of all nonterminal symbols on the left side of the marked rules before their rewriting starts.

As stated in subsection 3.2.2, L-systems are generally well suited to model the growth of plants because they were initially designed for the description of a natural growth process. Street networks, analogous to that of plants, can be considered as the result of a natural growth process which starts in open space with main roads followed by side streets. The increasingly dense network can therefore be conveniently simulated with L-systems as it has been also shown in (Coelho et al., 2007), (Chen et al., 2008), and (Weber et al., 2009). In contrast, the construction of buildings usually does not follow a typical growth process but a sequence of partitioning steps in a coarse to fine fashion (Wonka et al., 2003). Furthermore, geometric conditions as they frequently occur in buildings are difficult to integrate and lead to intricate production rules (Becker et al., 2013). Consequently, L-systems cannot be easily adapted to the modeling of buildings, particularly if their shapes are complex.

In order to explore different designs in urban planning applications, split grammars for building facades are proposed in (Wonka et al., 2003). Split grammars are a specialized type of set grammars (Stiny, 1982) operating on basic shapes but they circumvent the sub-shape matching problem of shape grammars (Stiny and Gips, 1972); for further details see subsection 3.2.3. These basic shapes are simple geometric objects such as parameterized cuboids and prisms which represent simple building blocks. More precisely, a split grammar is a set grammar over the vocabulary $B = \{f(s)\}$, where s is a basic shape and f an affine transformation, where two rule types of the form $a \rightarrow b$ are allowed: split rules and conversion rules. In a split rule, a is a connected subset of B that is split into multiple shapes so that b contains the same elements as a with the exception of the element to which the rule is applied. In a conversion rule, a contains a basic shape that is in b transformed to another basic shape on conditions that it is enclosed in the volume of the basic shape in a and all other elements in b remain the same as in a. A major difference between split and conversion rules is that a split rule always preserves the total volume of its elements, whereas it can be smaller after applying a conversion rule. The restrictions on these two types of rules simplify the derivation process in contrast to general shape grammars and enable an effective modeling of buildings.

An example of split grammar rules and their derived building facade is illustrated in Figure 2.2, in which nonterminal and terminal shapes are represented as white and colored areas, respectively. In combination with a separate control grammar that controls the attribute propagation and rule selection of the split grammar, objects in urban areas can be modeled so that they correspond to architectural principles. The proposed split grammar is suitable for the modeling of objects with repetitive structures as this is particularly the case for building facades.

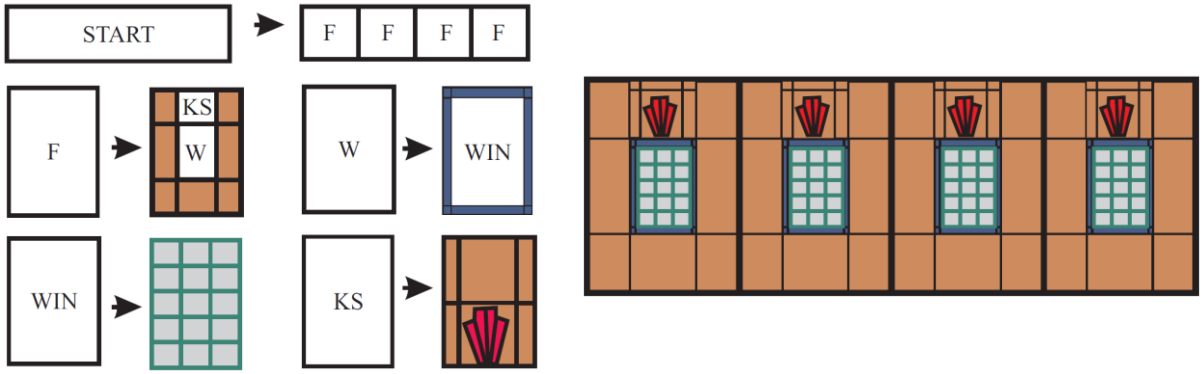


Figure 2.2. A set of simple split grammar rules (left) and their derived building facade (right) (Wonka et al., 2003).

In (Müller et al., 2006b), one of the most frequently used grammar for procedural building modeling is introduced as the so-called CGA (Computer Generated Architecture) shape. It basically combines and extends the work presented in (Parish and Müller, 2001) and (Wonka et al., 2003) by making use of transformation operations similar to L-systems and split operations to generate large and detailed urban environments. Note, although CGA shape features characteristics of L-systems and split grammars, it is a sequential grammar similar to Chomsky grammars and does not follow the parallel derivation of L-systems. Major objectives of CGA shape are the creation of sufficient geometric details in mass models, which is a limitation in (Parish and Müller, 2001), and the reduction of an excessive number of splits that are required in (Wonka et al., 2003) for complex mass models. Additionally, CGA shape addresses the problem of both approaches that facade elements (e.g., windows, doors, etc.) are often cut in an unnatural way whenever different volumes of a building model intersect each other. This is the case, for example, if a volume is placed in front of another volume so that elements on the partially occluded facade become partially occluded as well. The reason for this is that volumes in (Parish and Müller, 2001) and (Wonka et al., 2003) are not aware of each other. In order to overcome these issues, the CGA shape framework defines a shape as a set consisting of a terminal or nonterminal symbol to unambiguously identify the shape and geometric and numeric attributes. The geometric attributes include the position and three orthogonal vectors, defining the local coordinate system of the shape, and a size vector which represents the extent of the shape in all three dimensions. These geometric attributes define the so-called scope of a shape which can be considered as an oriented bounding box that contains the shape. For the modification of a shape with CGA shape, four types of production rules are used: (i) Scope rules are used to modify the scope of a shape by translating, rotating

or resizing the shape. (ii) Basic split rules are equivalent to the split rules in split grammars with the addition that they are able to deal with both absolute and relative values. (iii) Repeat rules tile a specific element along an axis of the local coordinate system with as many instances of another element as there is space. (iv) Component split rules divide a shape into its geometric components based on keywords such as faces or edges. With this, a shape can be split into shapes of lesser dimensions to access, for example, the faces of a mass model. In order to address the problem of partially occluded facade elements, an occlusion query test for intersecting shapes is carried out that differentiates between no, partial, and full occlusions. This information is considered whenever a facade element is placed. Furthermore, snap lines are introduced to alter existing shape rules in order to snap elements to a dominant line or face in the shape configuration so that the facade layout is further improved. Besides the reconstruction of building facades, some special shapes are defined that can be placed on top of a flat shape to generate different roof shapes.

Due to its simplicity and powerful expressiveness several applications, extensions and variants of shape grammars in general and CGA shape in particular have been developed. Some are outlined in the following. In (Müller et al., 2005), for example, the practical applicability of (Parish and Müller, 2001) and (Wonka et al., 2003) is demonstrated to automatically reconstruct Roman housing architecture for the famous Pompeii site. Similarly, CGA shape grammar is used in (Müller et al., 2006a) to procedurally generate archaeological sites in Mexico consisting of Puuc-style buildings which is a style of Pre-Columbian Mayan architecture and in (Dylla et al., 2009) to reconstruct the entire city of ancient Rome. An interactive visual editing framework based on the language elements of CGA shape is presented in (Lipp et al., 2008) to provide direct and persistent local control over generated instances. In (Hohmann et al., 2009), Cityfit is presented to fit shape grammars to images and derived point clouds in order to reconstruct detailed facades that include all significant elements larger than 50 cm. For this, the Generative Modeling Language (Havemann, 2005) as grammar description language and the main concepts of CGA shape is combined in the shape grammar system of CityFit. In the procedural modeling language G^2 (Generalized Grammar) proposed in (Krecklau et al., 2010), various concepts in the field procedural modeling are adapted to increase the freedom of simple boxes in CGA shape and thus its descriptive power. For this, the use of nonterminal symbols for procedural modeling is generalized in G^2. On the one hand, multiple classes of nonterminal objects with domain-specific attributes and operators are introduced to encapsulate different modeling strategies such as box-like modeling or freeform deformations. On the other hand, G^2 enables the definition of abstract structure templates for flexible reuse within the grammar by accepting nonterminal symbols as rule parameters. The capability of the G^2 framework is demonstrated in (Krecklau and Kobbelt, 2011a) in which bridges, roller coasters, and wall-mounted catenaries are exemplarily generated. In (Müller Arisona et al., 2013), a methodology that combines CGA and photogrammetry to create building models with detailed facade structures and textured roof geometry is shown. More recently, CGA++ was presented in (Schwarz and Müller, 2015). A major objective of the advanced CGA version is to enhance the interaction between shapes. Thereby, it can be ensured that, for example, certain elements like doors are generated for each building in an appropriate number, masses

can be merged to avoid overlapping geometries, and partially occluded elements are not necessarily omitted but adjusted together with their contextual related elements. The latter is often the case for those windows of a floor where only some of them are partially occluded. Instead of removing all partially occluded windows, CGA++ is able to adjust all windows of the floor accordingly.

In (Vanegas et al., 2010), another grammar-based reconstruction method for buildings with flat rooftops is presented that follows the Manhattan-world assumption (Coughlan and Yuille, 1999), i.e., the predominance of three mutually orthogonal directions in the scene. Here, buildings are decomposed into floors with a constant height value and the outline of each floor is defined by a two-dimensional polyline. Since Manhattan world assumption is assumed, each polyline can be formulated as a string based on the turtle graphics formulation using the alphabet $\{f(l), +, -\}$, where $f(l)$ reflects the length of a line segment l, and $+$ and $-$ the angle between successive line segments (i.e., either +90° or -90°). The rectangular outline of the first floor shown in Figure 2.3 is, for example, defined by $f(l_0) - f(l_1) - f(l_0) - f(l_1)$. Successive floors are then generated by applying rewriting rules that represent the changes, observed in multiple calibrated aerial images, to the previous floor string. The transitions are formulated as rewriting rules that replace a letter with a sequence of new letters and follow the generalized rewrite rule (GRR)

$$f(l) \rightarrow f(a) - f(c) + f(l - a - b) + f(c) - f(b), \tag{2.1}$$

such that $a + b < f(l)$ and $c \geq 0$. As shown in Figure 2.3, this is the case for transitions of the type L-shape ($a = 0$ or $b = 0$), U-shape ($a \neq 0, b \neq 0, c \neq 0$), and push-back ($a = b = 0, c > 0$). Additionally, in order to ensure a plausible structure, several intra- and inter-floor constrains are considered during the application of GRR (e.g., the size of a successive floor must be either equal or smaller than the size of the previous floor).

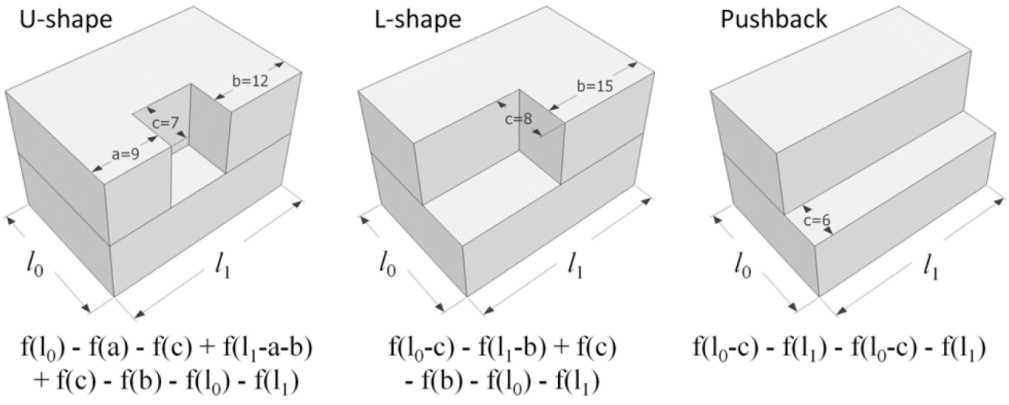


Figure 2.3. The representative strings of the second floors generated with the rewriting rules presented in (Vanegas et al., 2010).

The presented procedural modeling techniques thus far are very well suited to generate large urban environments with realistic looking buildings in a reasonable amount of time. But a

general problem of procedural modeling frameworks is that production rules are frequently assumed to already be available from the outset (van Gool et al., 2013). In order to provide a variety of complex buildings, a large number of sophisticated production rules is needed which is often done in a manual way. The manual set up of rules is, however, often not only laborious and time-consuming but also requires expert knowledge (Becker et al., 2013). To overcome this issue, techniques for the inverse procedural modeling have been invented which automatically derive rules from observations. In (Aliaga et al., 2007) style grammars, an extension of the grammars presented in the Build-by-Number system (Bekins and Aliaga, 2005), are presented that facilitate the interactive creation of new buildings in a style comparable to already generated buildings. For the generation of the grammar, images are manually mapped to simple geometric building models, which in turn are then decomposed into smaller facade features (e.g., floors, doors, windows, etc.). Based on the decompositions, repetitive patterns are detected and a style grammar is formulated. Applied on a composition of building blocks, new buildings can be created in a style of existing buildings. Methods for the automatic derivation of shape grammar rules for building facades from single facade images are shown in (Müller et al., 2007) and (van Gool et al., 2007). In (Becker, 2009), first a data-driven reconstruction procedure is performed on terrestrial LiDAR data. The result is then used as a knowledge base from which production rules are automatically defined for further processing of partially or completely occluded facades. Another inverse procedural method is proposed in (Bokeloh et al., 2010) that automatically extracts general rewriting systems, context free hierarchical rules, and grid-based rules from a given model to create objects with a similar shape. A machine learning approach based on Inductive Logic Programming (ILP) is applied in (Dehbi and Plümer, 2011) to automatically derive grammar rules for building parts. Note, logic programs and attribute grammars, which extend context-free grammars by attributes and semantic rules, can basically be considered as the same language with syntactic differences. The capability of ILP and its limitations have been demonstrated for straight stairs. In (Vanegas et al., 2012), an interactive framework is presented that combines forward and inverse procedural urban modeling strategies to alter the input parameters according to the desired output. A framework that automatically extracts split grammars and evaluates their meaningfulness is illustrated in (Wu et al., 2014). In (Dehbi et al., 2017), a statistical relational learning approach is proposed for the automatic derivation of grammar rules. Here, a weighted context-free grammar is generated based on SVM techniques and it is further extended by parameters and constraints learned by a statistical learning method using Markov logic networks. Another algorithm that automatically derives formal grammar rules to describe the structure of building footprints and facades in a compact way based on information about symmetry, regular patterns, and palindrome-like structures is presented in (Dehbi et al., 2016a).

For some time, research also focusing on parallelizing shape grammars has been carried out to reduce the computational time for the generation of large urban areas within procedural urban modeling. In (Marvie et al., 2012), GPU shape grammars are proposed for the interactive generation of large urban areas with modern graphics hardware. They enable the creation of massive, detailed models within a short period of time. One limitation is, however, that the roof generation does not preserve the overall footprints. Therefore, roof structures are

generated on CPU, caching only the output within GPU buffers. To overcome this issue, GPU roof grammars (Buron et al., 2013) have been proposed. They extend the GPU shape grammars with further rules to guide the roof creation and thus bring parallelism to grammar based roof generation.

Another significant step forward in parallel evaluation of procedural shape grammars on GPU has been presented in (Steinberger et al., 2014a). The proposed schema for parallel generation of architecture (PGA) explicitly expresses independence in the grammar and introduces intra-rule parallelism which distributes the evaluation of a single rule to more than a single thread. Furthermore, PGA avoids slow global memory access by local rule grouping strategies and intelligent rule scheduling, and provides full modeling capabilities. The expressive power is assured since PGA is compatible with CGA shape grammars in the sense that existing rule sets can be ported with little effort to PGA. The amount of generated geometry, however, exceeds in practice for large urban areas the common storage capabilities of consumer graphics hardware. Therefore, an extension of PGA is presented in (Steinberger et al., 2014b) that considers the visibility and different levels of detail in the evaluation so that the amount of generated geometry is reduced. Furthermore, the combined generation and rendering approach further decreases the processing time and overcomes the limitations of streaming pre-generated data. Another extension of PGA is presented in (Dokter, 2014), in which the focus is on exploring different strategies to execute procedural modeling on GPU. In this regard, methods of controlling the GPU rule evaluation process as well as the benefits and drawbacks of deriving precompiled rule sets and interpreting them at runtime are investigated.

With particular focus on real-time generation of flat building facade textures, a grammar-driven method has been proposed in (Krecklau and Kobbelt, 2011b). Analogous to the F-shade procedure presented in (Haegler et al., 2010), procedural facade textures are not explicitly generated but implicitly archived on-the-fly on the graphics card by a per-pixel evaluation. For this, facades are initially subdivided along a certain axis by a split grammar so that the pixel evaluation starts with the full facade that is traced down until a terminal symbol is found. The terminal symbol in turn is associated with a texture and thus defines the color of the pixel. The proposed grammar evaluation method has been extended in (Krecklau et al., 2013) to render textured facades in a view dependent manner so that the number of polygons is reduced. Furthermore, 3D geometric details are introduced in addition to flat textures.

Although most procedural modeling approaches for buildings focus on facades, some research has been carried out that integrates procedural modeling techniques in the generation or reconstruction process of other building parts. With respect to indoor modeling, for example, grammar supported automatic reconstruction methods of two-dimensional indoor maps are presented in (Becker et al., 2013) and (Philipp et al., 2014). Examples for the procedural generation of three-dimensional indoor maps are given in (Gröger and Plümer, 2010) and (Becker et al., 2015). In addition, some approaches for the incremental refinement of building models have been proposed using, for example, string grammars (Kada, 2014) or weighted attribute context-free grammars (Dehbi et al., 2016b). Furthermore, an engine with the primary objective to provide building models in multiple LODs by generating them with

procedural modeling techniques has been presented in (Biljecki et al., 2016a). Regarding roof modeling based on procedural modeling techniques, however, not much research has been carried out. As, for example, recently stated in (Edelsbrunner et al., 2016), numerous procedural modeling systems have been proposed that allow the creation of immense and realistic details, but roofs still pose a problem. A majority of procedural frameworks are either limited to flat roofs (e.g., (Parish and Müller, 2001)) or they use a basic roof shape from a predefined library that is, for example in (Müller et al., 2006b), obtained by a straight skeleton computation (Aichholzer et al., 1995). In contrast, a building reconstruction approach is proposed in (Milde et al., 2008) that first detects simple roof sub-shapes in a model-driven way. Then, a context free, attributed grammar is used in which five basic sub-shapes correspond with nonterminal symbols and that can be derived from the start symbol in the first derivation step. In order to combine these basic sub-shapes, three different connectors are introduced in the grammar. A major limitation of the proposed approach is, however, that the selection of rules is carried out manually. One of the first approaches with particular focus on procedural modeling of roofs for arbitrary building outlines is presented in (Kelly and Wonka, 2011). With the proposed interactive procedural roof modeling framework, more complex roof shapes such as curved and overhanging roofs can be interactively created.

3. Fundamentals

In this chapter a methodological overview is given on methods and techniques used in this thesis. More precisely, section 3.1 is related to chapter 4 and summarizes three groups of frequently applied segmentation methods of planar areas. In section 3.2, different types of grammars are described that form the basis for the building knowledge derivation procedure presented in chapter 5. The remaining subsections are related to chapter 6 and 7 and summarize different solid modeling techniques which are generally applicable for building model reconstruction purposes, different types of data clustering methods, and a commonly used method to register point clouds.

3.1 Segmentation of Planar Areas

In the context of data-driven reconstruction approaches, automatic detection of planar areas is a crucial operation. Many segmentation methods have been therefore developed of which three groups are briefly summarized in this section. A more detailed review of these methods is, for example, given in (Vosselman et al., 2004) and (Vosselman and Klein, 2010).

For the detection of planes in point clouds, the principles of 2D Hough transform (Hough, 1962) can be extended to the three-dimensional space. For this, a plane in the object space (X, Y, Z) is defined as

$$d = X \cos\alpha \cos\beta + Y \sin\alpha \cos\beta + Z \sin\beta. \tag{3.1}$$

In addition to the object space, a 3D parameter space (α, β, d), also called a Hough space, is defined by the plane parameters α, β, and d so that each point in the parameter space defines a plane in the object space. For each point in the object space, all possible planes that have this point in common are translated into the parameter space. As a result, each point in the object space is transformed into a sinusoidal surface in the parameter space. Points in the object space that are coplanar feature a common intersection in the parameter space. Thus, the values of this intersection represent the parameters of the plane in the object space on which the coplanar points are located. A simple example is shown in Figure 3.1. Here, the three black points in the object space (1, 0, 0), (0, 1, 0), and (0, 0, 1) are transformed into the three sinusoidal surfaces in the parameter space. The values of their intersection define in the object space the parameter values of the orange plane on which the three points are located. In order to keep the computational time low, the parameter space is usually discretized into bins and defined in a local coordinate system whose origin is located in the center of the given point cloud. The discretization of the space is also needed to deal with noise in the measured point coordinates which causes that the sinusoidal surfaces of coplanar points do not exactly intersect in one point. By iteratively determining the bin with the largest number of intersect-

ing surfaces and removing these surfaces from the parameter space, planes and their corresponding points in the object space can be detected. For the determination of the plane parameters based on a single bin, often either the parameter values of the bin in the parameter space are used or a least-squares fitting to its corresponding points in the object space is applied. The iterative procedure stops if no further bin with a minimum number of surfaces can be selected in the parameter space. For a survey of Hough transform with more details see, for example, (Mukhopadhyay and Chaudhuri, 2015). The principles of the 3D Hough transformation to determine planar regions in point clouds have been extended in (Rabbani and van den Heuvel, 2005) for the segmentation of simple primitives such as cylinders as well as in (Khoshelham, 2007) for 3D objects with arbitrary shapes.

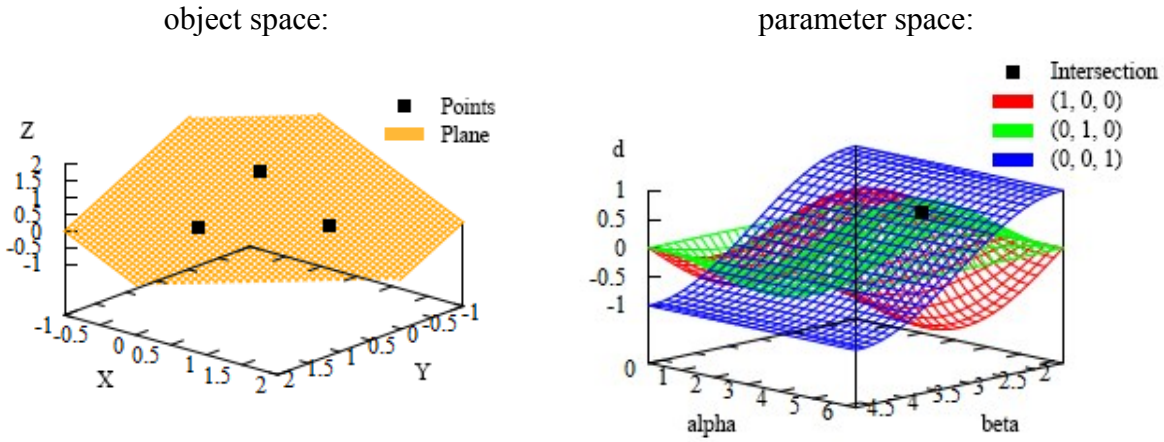


Figure 3.1. Transformation of three coplanar points in the object space (left) into the parameter space (right).

Another frequently applied group of methods that iteratively detect planes in point clouds makes use of RANSAC, which was originally introduced in (Fischler and Bolles, 1981) as a general approach to robust model fitting. Analogous to Hough transform, it determines in each iteration the best-fitting plane. For this, first three points are randomly selected from the data and the parameter values of the corresponding plane are calculated. Then, the consensus set of it is identified which consists of all points of the point cloud that can be assigned to this plane according to a predefined distance threshold. These two steps are applied several times until all possible planes and their consensus sets have been determined. By determining the plane with the largest consensus set out of this group of planes and consensus sets, the best fitting plane can be extracted. By repeating this procedure and ignoring during the next iterations all those points that were assigned to an extracted plane, all planes in a point cloud can be determined. The procedure stops if the number of points in the largest consensus set of an iteration is less than a predefined threshold. Several variants have been proposed to reduce the computational time. A common method is, for example, to limit the number of repetitions in an iteration which causes, however, that not always the best-fitting plane is determined. A more advanced optimization method of RANSAC is presented in (Schnabel et al., 2007). Here, an octree (see section 3.3.1) is utilized to organize the point cloud and to reduce the overall computational cost. In (Bretar and Roux, 2005), a normal-driven RANSAC approach

is proposed in which not all possible triples of points are selected to determine a plane but only those whose points share the same orientation of normal vectors with respect to a predefined threshold. Thereby, a lower number of consensus sets needs to be tested.

Hough transform and RANSAC are global approaches that ignore local point neighborhood. Thereby, both segmentation methods are generally robust against local outliers and can provide good results even in noisy point clouds. However, their segmentation result is sensitive to their chosen parameter values which is, for example, the minimum number of points that is required to extract a plane. Additionally, the bin size of the parameter space in Hough transform based segmentation methods needs to be carefully chosen. On the one hand, if the bin size is too small, sinusoidal surfaces of actually coplanar points might pass through different nearby bins due to some noise in their coordinates so that their plane cannot be detected. On the other hand, the use of a large bin size causes that estimated plane parameters might become less accurate or that incorrect planes are detected. This analogously applies in RANSAC based segmentation methods to the parameter value that defines the tolerance range during the determination of the consensus sets. Another disadvantage of both segmentation methods is that due to their global approach, it has to be considered that coplanar points belonging to different (non-coplanar) planes might be segmented together. A typical case which might appear is exemplarily illustrated in Figure 3.2. Here, the incorrectly detected plane features more corresponding points than each plane of a single roof plane. Thereby, only one instead of five roof planes is extracted. For roof plane extraction purposes, it also has to be considered that coplanar segment patches are extracted together without any distinction between them. As a consequence, additional effort in form of a connected component analysis needs usually to be carried out to derive all planar areas of neighboring points.

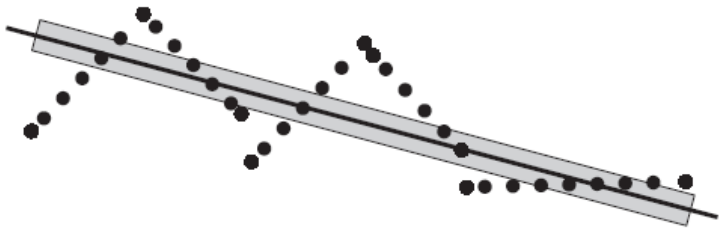


Figure 3.2. Side view of an incorrectly detected plane (black line) which is supported by the points in the grey area (Vosselman and Klein, 2010).

In the third group of segmentation methods, surface growing segmentation techniques are used to iteratively detect planar areas or other smooth types of surfaces. For this, the principles of the well-known region growing algorithm are extended to the three-dimensional space. For further details about the two-dimensional region growing see, for example, (Pratt, 2007) and (Gonzalez and Woods, 2008). In surface growing approaches, the spatial proximity of points is, in contrast to Hough transform and RANSAC, already taken into account during the segmentation process so that coplanar patches are separately extracted. An iteration of the surface growing algorithm basically consists of two parts: seed detection and segment growing. In the seed detection step, small sets of nearby points are examined in terms of their

properties to decide whether or not they constitute a part of a surface. If an initial surface is identified, surface growing continues with the growing step. Otherwise, the set of nearby points is discarded as the initial surface and another set of nearby points is examined. In the case that no initial surface can be identified, the iterative segmentation procedure stalls. In the growing step, all so far unassigned neighboring points of the previously detected seed surface are examined. If they share similar properties as the seed surface they can be added to the currently processed surface. Typical properties that are taken into consideration during the growing step are, for example, local surface normal direction, intensity, color, and distance between the currently grown surface and the point to be examined. If a neighboring point is assigned to a segment, then its neighboring points need to be examined as well if they can be assigned to the grown segment. Additionally, in order to improve the accuracy, properties of the currently grown segment can be updated before further neighboring points are tested. But in order to keep the computational time low, properties of the segment are often only updated if the surface has grown by a certain percentage. Once no further neighboring points can be added to the segment, the growing process stalls and the next iteration starts to detect the next surface.

3.2 Grammars

Grammars are a branch of linguistics and deal with morphology, the study of word structure and word formation, and syntax, the study of sentence structure. They have been studied for many years and became important in computer science as well. Particularly in theoretical computer science, formal grammars are used as base of compilers, regular expressions, parsers, state machines based on Markov chains, and many others. They have emerged from various origins of which the main root is mathematics (in particular combinatorics and algebra of semigroups and monoids). Probably the best-known pioneers in this research field are Axel Thue, who investigated in (Thue, 1906; Thue, 1912) avoidable and unavoidable patterns in long and infinite words, and Alan Turing together with Emil Post, who proposed in (Turing, 1936) and (Post, 1936) independently from each other a mathematical model of computation (i.e., Turing machine and Post-Turing machine) which defines in computability theory and computational complexity theory the set of allowable operations used in computation and their respective costs. A special type of formal grammar is the Lindenmayer system (L-system) which was invented in the 1960s. It represents a parallel rewriting system and is thereby particularly suitable to model the growth process in cellular biology. Other well-known types of grammars are the shape and the set grammar which were introduced in the 1970s and 1980s, respectively. They are both typically used for the generation of two- or three-dimensional geometric shapes. All previously mentioned grammars have in common that they operate either on strings or shapes and are thus not suited to model transformations of graphs. In order to overcome this issue, graph grammars were introduced in the late 1960s. The main aspects of these four different types of grammars are summarized in the following subsections. For some further details see (Ehrig, et al., 2006), (Rozenberg and Salomaa, 1997), and (Rozenberg, 1997).

3.2.1 Formal Grammars

In formal language theory, a formal grammar is a rewriting system over a finite sequence of symbols, so-called strings, from an alphabet that consists of a finite set of discriminable symbols. Analogous to semi-Thue systems (Thue, 1914), formal grammars define an alphabet and production rules for strings. But in contrast to a semi-Thue system, a distinction is made between terminal and nonterminal symbols. Additionally, a start symbol is specified in formal grammars amongst nonterminal symbols. More formally, a formal grammar G, as first proposed by Noam Chomsky (Chomsky, 1956; Chomsky, 1957), is defined as a 4-tuple (V, T, P, S), where

- V is a non-empty, finite set of nonterminal symbols.
- T is a non-empty, finite set of terminal symbols that is disjoint from V.
- P is a finite set of production rules of the form $\alpha \to \beta$ where $\alpha \in (V \cup T)^* V (V \cup T)^*$, $\alpha \neq \varepsilon$, and $\beta \in (V \cup T)^*$. Note, * denotes the Kleene star, ε is the empty string, and α and β are respectively called the left-hand side (LHS) and the right-hand side (RHS) of a production rule.
- $S \in V$ is the start symbol.

A sequence of applied production rules is called a derivation. The set of all possible strings of terminals that can be derived from a formal grammar G is called its language, denoted as $L(G)$ and defined as

$$L(G) = \left\{ w \mid S \overset{P^*}{\to} w, w \in T^* \right\}. \tag{3.2}$$

The simple grammar G_1 with $N = \{S\}$, $T = \{a, b\}$, and $P = \{S \to aSb, S \to ab\}$ defines, for example, the language $L(G_1) = \{a^n b^n \mid n \in \mathbb{N}^+\}$ in a compact way.

In (Chomsky, 1959), formal grammars are categorized in terms of their production rules into four different types of grammars. An overview of the so-called Chomsky hierarchy is given in Table 3.1. The grammar classes are ranked by decreasing order of expressive power. Moreover, the relationship between the derived languages of the four grammar types can be described as

$$L_3 \subset L_2 \subset L_1 \subset L_0. \tag{3.3}$$

Although grammars of classes with a lower rank are more powerful, they are in practice generally less useful because they cannot be efficiently parsed. For example, the languages that are generated by type-0 grammars may be semi-decidable. In contrast, languages are decidable if they belong to the class of context-sensitive grammars. In order to decide whether a string belongs to a language, exponential time is required for type-1 languages, polynomial time for type-2 languages, and linear time for type-3 languages. Since the parser complexity

of type-1 grammars is PSPACE-complete, mainly type-2 and type-3 grammars are applied in practice.

Table 3.1. Chomsky hierarchy.

Type	Grammars	Form of production rules	Languages	Generative power
0	Unrestricted	$\alpha \rightarrow \gamma$, where $\alpha \in (V \cup T)^* V (V \cup T)^*$, $\alpha \neq \varepsilon$, and $\gamma \in (V \cup T)^*$	Recursive enumerable	Turing machines
1	Context-sensitive	$\alpha A \beta \rightarrow \alpha \gamma \beta$, where $A \in V$, $\alpha, \beta \in (V \cup T)^*$, and $\gamma \in (V \cup T)^+$ ($S \rightarrow \varepsilon$ allowed if S does not appear on the RHS)	Context-sensitive	Linear bounded automata
2	Context-free	$A \rightarrow \gamma$, where $A \in V$, and $\gamma \in (V \cup T)^*$ ($S \rightarrow \varepsilon$ allowed if S does not appear on the RHS)	Context-free	Pushdown automata
3	Regular	$A \rightarrow aB$ (right-linear), $A \rightarrow Ba$ (left-linear), and $A \rightarrow a$, where $A, B \in V$, and $a \in T^*$	Regular	Finite automata

3.2.2 Lindenmayer Systems

A rewriting system with particular focus on the development of filamentous organisms is presented in (Lindenmayer, 1968). Similar to a formal grammar, an L-system can be defined as a three-tuple (V, P, ω), where

- V is a non-empty, finite set of symbols.
- P is a finite set of production rules.
- ω is the initial string, so-called axiom, from which the rewriting starts.

Starting from the axiom, production rules are iteratively applied. But in contrast to formal grammars, L-systems can apply in each iteration more than one production rule in parallel. Therefore, formal grammars are referred to sequential rewriting systems while L-systems are considered as parallel rewriting systems. Thereby, the natural growth of a biological organism, where many cell divisions may occur simultaneously at different positions, can be modeled in an appropriate way. A further significant difference to formal grammars is that L-systems do not differentiate between terminal and nonterminal symbols. Therefore, the termination of an L-system derivation is usually defined by a predefined number of iterations.

Similar to formal grammars, L-systems are categorized in terms of their production rules. Here, a distinction is made between context-free L-systems (0L-systems) and context-sensitive L-systems (1L-system). In contrast to 0L-systems, neighboring symbols of a symbol that is to be modified are incorporated in 1L-systems. For this, the symbols "<" and ">" are introduced to separate the left and the right context from the symbol that is to be modified. Both types of L-systems do not follow the categorization of the Chomsky hierarchy, but the relations between Chomsky classes of languages and language classes generated by L-systems can be described as shown in Figure 3.3.

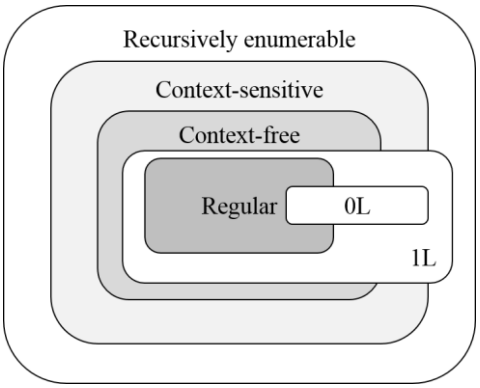


Figure 3.3. Relations between Chomsky classes of languages and language classes generated by L-systems according to (Prusinkiewicz and Lindenmayer, 1990).

For the geometric interpretation of strings, turtle geometry (Abelson and diSessa, 1981), as proposed in (Prusinkiewicz, 1986), can be adapted. The state of a turtle is a triplet (x, y, α), where the coordinates (x, y) represent the position of the turtle and α the direction in which the turtle is facing. According to a predefined step size d and an angle increment δ, the turtle follows the basic turtle commands represented by the symbols

- F: Move forward so that the current $state_i = (x, y, \alpha)$ changes to $state_{i+1} = (x + d \cdot cos(\alpha), y + d \cdot \sin(\alpha), \alpha)$ and draw a line segment between $state_i$ and $state_{i+1}$.
- f: Move forward as defined for F but without drawing a line.
- +: Turn right by angle δ so that the current $state_i = (x, y, \alpha)$ changes to $state_{i+1} = (x, y, \alpha + \delta)$.
- -: Turn left by angle δ so that the current $state_i = (x, y, \alpha)$ changes to $state_{i+1} = (x, y, \alpha - \delta)$.
- |: Turn away so that the current $state_i = (x, y, \alpha)$ changes to $state_{i+1} = (x, y, \alpha + 180°)$.

For all other symbols, the turtle preserves its state. There are several variants that extend the list of basic turtle commands. Useful extensions are, for example, the symbol "[", which pushes the current state of the turtle into a stack, and the symbol "]", which pops a state from the stack and makes it to the current state of the turtle without drawing a line between the last and current state. With these two symbols, the growth of more complex structures can be

compactly described that frequently occur in nature. In order to describe, for example, the growth of algae, a simple L-system can be defined, where $V = \{F, +, -, [,]\}$, $P = \{p_1: F \rightarrow F[+F]F[-F]F\}$, and $\omega = F$. The geometric interpretation of the first iterations with $\delta = 25°$ is shown in Figure 3.4. For further details regarding the modeling of plants with L-systems, see (Prusinkiewicz and Lindenmayer, 1990).

Figure 3.4. Growth modeling of algae with an L-system and its geometric interpretation for the first iterations.

3.2.3 Shape and Set Grammars

A widely applied formal description of designs is a shape grammar (Stiny, 1975). The concept of shape grammars was originally introduced in (Stiny and Gips, 1972) for painting and sculptures and since then often utilized in the context of architecture. In contrast to formal grammars and L-systems which define a grammar over an alphabet of symbols and generate one-dimensional strings of symbols, shape grammars operate on an alphabet of shapes and generate n-dimensional shapes. According to (Stiny, 1980), a shape is a finite set of so-called maximal lines, each defined by a set of two distinct points, that form an entity. More formally, a shape grammar is a four-tuple (S, L, R, I), where

- S is a finite set of shapes.
- L is a finite set of symbols.
- R is a finite set of shape rules (productions) of the form $\alpha \rightarrow \beta$, where α is a labeled shape in $(S, L)^+$, and β a labeled shape in $(S, L)^*$.
- I is a labeled shape in $(S, L)^+$ called the initial shape.

A labeled shape is defined in (Stiny, 1980) as an ordered pair $\langle s, P \rangle$, where s is a shape and P a finite set of labeled points, which are located with respect to the shape s but do not necessarily have to coincident with the lines in s. A labeled point $p: A$ in turn is a point p with a symbol A associated with it.

Production rules in shape grammars may be applied both serially as in Chomsky grammars or parallel as in L-systems. In order to apply a shape rule, all possible locations of the shape on

the LHS of the rule, taking into consideration geometric transformations (scale, translation, rotation, mirror image), need to be determined in the current shape. The derivation process of shape grammars thus involves a sub-shape matching problem which is much more intricate than the one-dimensional sub-string matching problem in formal grammars and L-systems. Therefore, around 50 % of the implementation effort of shape grammars is according to (Gips, 1999) related to this issue.

A simple variant of the shape grammar formalism that circumvent the sub-shape matching problem is presented in (Stiny, 1982). Here, designs are treated as symbolic objects and not as spatial objects so that the sub-shape matching problem of shape grammars is reduced to a simple symbol matching problem. Thereby, rules in set grammars are applied to subsets of sets of labeled shapes to produce other such sets while rules in shape grammars are applied to sub-shapes of labeled shapes to produce other such labeled shapes. This usually makes set grammars more suitable for computer implementation because designs in languages defined by set grammars consist of shapes in S and not of shapes and sub-shapes of shapes in S as in the case of shape grammars.

3.2.4 Graph Grammars

Graph grammars (GG) were introduced in the late 1960s (Pfaltz and Rosenfeld, 1969; Schneider, 1970) and since then they have been applied in several domains such as logic programming, pattern recognition, model transformation, compiler construction, modeling of concurrent systems, and many others. They can be considered as a natural generalization of classical Chomsky grammars where strings are replaced by graphs and thus they can be used for the formal description of transforming an original graph into a new graph. Note, the term graph grammar and the terms graph replacement system and graph rewriting system are sometimes equivalently used in literature. The main components of a GG are an initial graph and a finite set of production rules. According to (Engelfriet and Rozenberg, 1997), a production rule is, in general, a triple (M, D, E) where M is the so called mother graph, D the so called daughter graph, and E an embedding mechanism. A production rule is applied to a given host graph G by removing all occurrences of the mother graph M in G, replacing each of them by an isomorphic copy of the daughter graph D, and finally applying the embedding mechanism E to attach D to the remainder graph G^- of G. Two prominent basic examples of graph grammars are the node replacement graph grammar in which a node of a given graph is replaced by a new subgraph and the (hyper)edge replacement graph grammar in which a (hyper)edge of a given (hyper)graph is replaced by a new sub(hyper)graph; for further details see (Engelfriet and Rozenberg, 1997) and (Habel, 1992; Drewes et al., 1997).

During the last decades, several approaches to GG have been developed. Traditionally, there are two main groups of approaches that can be distinguished by their embedding mechanism: set-theoretic (or algorithmic) approaches (Nagl, 1987) and algebraic approaches (Ehrig et al., 1973). In set-theoretic approaches, a so called connecting mechanism is carried out, which embeds D in G^- by establishing edges between certain nodes of D and certain nodes of G^- according to the connection instructions from E, while in algebraic approaches, a so called

gluing mechanism is applied in which certain parts of D are identified in G^- and accordingly glued together.

In algebraic approaches, there are two main approaches for the replacement of M in G that were introduced in (Ehrig et al., 1973) and (Löwe, 1993): the double pushout (DPO) approach and the single pushout (SPO) approach. Both are based on category theory in which a pushout is defined for two given morphisms $f: A \to B$ and $g: A \to C$ as a triple (D, α, β) that consists of an object D, the two morphisms $\alpha: B \to D$ and $\beta: C \to D$, and that meets the universal property $\alpha \circ f = \beta \circ g$. For further details of pushouts in category theory, see, for example, (Mac Lane, 1998) and (Awodey, 2010). According to (Corradini et al., 1997), pushouts in suitable categories with graphs as objects and graph homomorphisms as arrows can be used to model direct derivations in DPO and SPO approaches. Other approaches to GG are, for example, logic based approaches (Schürr, 1997), algebraic and logic based approaches (Courcelle, 1990), and double-pullback based approaches (Ehrig et al., 1998).

In DPO approaches, a production is specified by $LHS \overset{l}{\leftarrow} K \overset{r}{\rightarrow} RHS$, where l and r are graph homomorphisms from a common interface graph (or gluing graph) K, so that a direct derivation consists of two gluing diagrams of graphs and total graph morphisms as illustrated on the left side of Figure 3.5. The interface graph of a production represents elements that should be preserved by its application while elements of $LHS - K$ are deleted to obtain the context graph D and elements of $RHS - K$ are added. The deletion of $LHS - K$ can be described via diagram (1) as an inverse gluing operation while the adding of $RHS - K$ is modeled in the second gluing diagram. In order to avoid conflicts during the replacement of subgraphs, each match m in a DPO approach must satisfy the so called gluing condition which consists of the following two parts: dangling condition and identification condition. The dangling condition states that the deletion of a vertex of G requires the deletion of all incident edges of that node. The identification condition ensures that only one pre-image is present for every element of G that should be deleted. In DPO approaches, rewriting is forbidden in all cases where the gluing condition in not met.

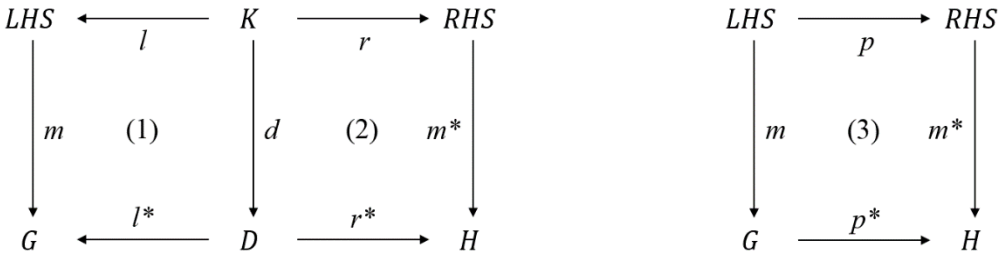


Figure 3.5. Schematic representation of a direct derivation based on (Corradini et al., 1997). Once for DPO approaches (left) and once for SPO approaches.

In contrast, a production in a single pushout approach is a partial graph homomorphism so that a direct derivation is given by a single gluing diagram as illustrated on the right side of Figure 3.5, where the match m must be total. For the application of a production rule, no

gluing condition needs to be satisfied. Here, deletion has in a case of conflict always priority over preservation. Moreover, an edge is automatically removed if one or both of its nodes is deleted even if this is not explicitly specified in the production rule. As a consequence, the co-match m^* might become a partial homomorphism. For a more detailed comparison between DPO and SPO approaches see (Ehrig et al., 1997).

3.3 Solid Modeling

Solid modeling deals with the geometric modeling of three-dimensional physical solid objects with the aim of providing a representation that can be utilized to algorithmically solve arbitrary geometrical problems in an automatic way. There are several techniques for modeling solids that generally dictate the reconstruction process of buildings from point clouds and that have an impact on their resulting shapes. Regarding (Mäntylä, 1988), there are three major approaches to solid modeling: decomposition models, constructive models, and boundary models. The first two techniques describe solids as a closed point set of a three-dimensional Euclidean space. While decomposition models generally combine basic building blocks from a fixed collection of simple primitives with gluing operations, constructive models include more general operations than mere gluing. Boundary models, on the other hand, represent the bounding surfaces of solids and can be described as a collection of faces. The main aspects and common data structures of these three major groups of solid modeling techniques are described in the following subsections. For some further details see (Hoffmann, 1989), (Mortenson, 1997), (Samet, 2006), (de Berg et al., 2008), (Stroud and Nagy, 2011), and (Elias, 2014). In addition, a good comparison between several representations is given in (Requicha, 1980), (Foley et al., 1990), and (Agoston, 2005).

3.3.1 Decomposition Models

Decomposition models describe solids as a collection of primitives that are glued together. Different variations arise out of the kind of used primitives and allowed gluing operations. Some decomposition models are summarized in the following paragraphs.

In spatial-occupancy enumeration, which is also known as exhaustive enumeration, solids are represented as a collection of non-overlapping volume elements with the same size and orientation so that the result is a regular subdivision of space. Depending on whether a subspace belongs to the solid it is marked as occupied and otherwise as vacant. To obtain the represented solid, all volume elements marked as occupied have to be glued together. The most common volume element for this purpose is a regularized cube which makes just one coordinate sufficient enough for each cube; for example the coordinate for the center or a corner of the cube. This representation form is similar to the binary pixel representation in a two-dimensional space, why such volume elements are often referred to voxels. Since exhaustive enumeration is an approximative modeling technique, it is able to represent all kinds of solids within a certain granularity. The representation of a valid solid is unambiguous and unique for a fixed volume element and space of interest. However, their validity depends on the connections between the volume elements because they may intersect at a common

vertex, edge, or face. A major advantage of exhaustive enumeration is that it is well suited for manipulation operations because the occupation of a cell can be efficiently determined. Hence, closed operations such as Boolean set operations can be easily implemented. The storage space, on the other hand, increases significantly with high resolutions due to the approximative nature of this modeling technique.

To overcome the huge memory consumption and the approximative accuracy of exhaustive enumeration, adaptive subdivision schemes were introduced. They replace the regular space partitioning by an adaptive space partitioning and make use of a lossless data compression scheme. The compression scheme is based on the observation that neighboring cubes tend to be part of the same class. Instead of keeping each cube separate, neighboring cubes of the same class are treated as a single subdivision of the space. This principle is for example also applied in run-length encoding, which belongs to the group of entropy encoding. A typical representative for this kind of encoding is the well-known Huffman coding (Huffman, 1952).

Common representatives of adaptive space subdivision schemes are the octree representation (Jackins and Tanimoto, 1980; Meagher, 1982) and the quadtree representation (Finkel and Bentley, 1974) for the three-dimensional and two-dimensional space respectively. The basic idea of an octree is the recursive subdivision of the space along the coordinate axes into eight octants that are organized in a tree of degree eight. Thereby, the root of an octree represents the whole space and each child node represents an octant of the space defined by its parent node. Depending on the overlap with the solid to be modeled, only those octant nodes that are either entirely within or completely outside the solid compose the leaf nodes of the octree. Consequently, all internal nodes represent octants with a partial solid overlap. The whole solid results from the gluing of the space represented in the leaf nodes of the tree that are entirely within the solid. The properties of octree and quadtree representations are essentially similar to those of the exhaustive enumeration with the exception of reduced storage space. Several variants have been introduced such as Bintree (Tamminen, 1984; Samet and Tamminen, 1985), which recursively divides the space of partially overlapping nodes along a single axis into two equal-sized subspaces, and ATree (Bogdanovich and Samet, 1999), which enables to vary the number of subdivisions for all partial overlapping nodes in each level and allows for unequal-sized subspaces.

However, even with the relaxed partitioning conditions of ATrees, solid surfaces that are not coplanar with any of the coordinate planes of a subdivision can be only approximately represented. To overcome this issue, binary space partitioning (BSP) trees can be utilized. BSP recursively subdivides an n-dimensional space into convex subspaces by a set of (n-1)-dimensional hyperplanes of arbitrary position and orientation. To minimize the height of a BSP-tree, usually the faces of the solid to be modeled are formulated and used as the dividing hyperplanes. Similarly to octrees, only leaf nodes represent space that is either entirely within or completely outside the solid. Based on an approach first utilized by Schumacker et al. (1969), BSP-trees were originally developed in the context of 3D computer graphics for determining the visibility of surfaces during the rendering process of static scenes (Fuchs et al., 1980). Since then they have been adapted for a variety of applications such as

ray-tracing (Naylor and Thibault, 1986), shadow generation (Chin and Feiner, 1989), solid modeling (Paterson and Yao, 1990), and image compression (Radha et al., 1996). A drawback of BSP-trees is that solids can have several different valid representations depending on the sequence in which the hyperplanes are applied. In the worst case, a solid is represented by a heavily unbalanced tree. Another drawback in 2D and analogous in 3D is that non-convex faces are decomposed during the BSP-tree construction and remain as part of more than one leaf node. To minimize these drawbacks, much research effort has been carried out, such as in (Naylor, 1993), to find an optimal splitting sequence.

Another approach that tackles the problems of exhaustive enumeration is cell decomposition. It is similar to exhaustive enumeration but allows a certain variety of the used volume elements. For this, basic cell types are defined whose instances are parameterized and topologically equivalent to a sphere. This allows, for example, the direct modeling of curved surfaces. However, solids that contain a hole or that are complex have to still be decomposed and their cells afterward glued together. In this regard, cells usually have to be completely disjoint and are not allowed to touch each other except in exactly one corner, an edge or a face. Compared to exhaustive enumeration, cell decomposition is generally able to represent the exact surface of all kind of solids even though in practice the appropriate trade-off between the exact surface representation and the complexity of a cell has to be determined. The variety of the volume elements also causes some additional effort for certain tasks. In order to proof the validity of a cell decomposition, every pair of cells must potentially be tested for an intersection because of the disappeared regularized structure. Furthermore, the direct creation of a cell decomposition is generally very hard. Nevertheless, it is crucial for many computational analysis algorithms and for example an essential ingredient of the finite element method (FEM); for further details of FEM see for example (Reddy, 2006) and (Zienkiewicz et al., 2013). Therefore, cell decompositions are usually created by a conversion from another representation.

A solid modeling technique that can be considered as a special case of decomposition models is primitive instancing. It is closely related to parametric modeling and group technology, since a fixed set of parameterized primitives is used to create instances of them that may not only differ in terms of affine transformations but also on other descriptive characteristics of a solid. Complex solids, especially if they consist of repetitive structures whose number may vary for different instances, can often be defined by a few characteristics. In (Foley et al., 1990), an example of gears with different number of teeth is given. In primitive instancing all gears can be defined by a few parameters such as diameter, thickness, hole size and number of teeth, whereas other modeling techniques would require a more complex data structure. To guarantee the validity of the represented solid, it is sufficient to define a range of values for each parameter. A main drawback of primitive instancing is, however, that generally no gluing or other operations are supported. The only possibility to modify a solid in primitive instancing is by changing its parameter values. The expressive power is therefore limited to a fixed collection of primitives and to their parameters, which are not always obvious and uniquely to define.

In summary, decomposition models are generally suitable for numerical problems and analyses that passively examine the model without modifying it. The direct creation to best represent a solid, however, is not always unique and obvious to humans. Therefore, they are mainly converted from other representations. Additionally, expressive power generally and precision particularly are often directly related to the required storage space. A more convenient group of solid modeling techniques for the automatic reconstruction of complex building shapes and their compact representation is given in the following subsection.

3.3.2 Constructive Models

In contrast to decomposition models, solids in constructive modeling are considered as point sets that can be represented by one simple point set or a combination of many simple point sets. For the latter, constructive modeling techniques generally provide more powerful operations for the construction of solids than just simple gluing operations.

In sweep representations for example, solids are considered as point sets moving through space that can be defined by a moving object and its trajectory. There are different types of sweep representations. Distinctions are usually made on the one hand between different kinds of allowed trajectories and on the other hand between different kinds of allowed moving objects. In translational sweeping, a planar shape, which generates a valid surface, is moved along a linear path that starts on the shape and that is perpendicular to it. The resulting extrusion of the planar shape defines the solid to be modeled. Analogously, in rotational sweeping, solids are defined by a planar shape and its rotation around an axis. Some sweep representations also allow the use of arbitrary trajectories which combine translations and rotations. In general, the moving objects used in sweep representations are not necessarily required to be planar shapes. Extruded objects created by milling machines and that thus represent a rigid solid are therefore typically used as well in a solid sweep representation. Furthermore, general sweep representations may involve nonrigid shapes and objects that may change their orientation, size and shape on the trajectory. The advantage of sweep representations is that they are unambiguous and especially suited to represent many mechanical parts or manufacturing objects with translational or rotational symmetry. However, the creation of general solids is entailed with additional effort because the application of regularized Boolean set operations to sweep representations without converting them first to another representation is generally difficult. Furthermore, it is not guaranteed that the resulting solid can be again represented in a sweep representation.

Another constructive modeling technique is half-space modeling in which a solid is represented as a combination of simple point sets, each specified by a closed half-space. A closed half-space is the set of solutions to a non-strict inequality of a characteristic function. For example, given a characteristic function $f(p)$ that divides the infinite space U at $f(p) = 0$ into two subsets, the two closed half-spaces H^+ and H^-asssociated with $f(p)$ are defined as $H^+ = \{p \in U \mid f(p) \geq 0\}$ and $H^- = \{p \in U \mid f(p) \leq 0\}$. Two commonly used inequalities that describe closed half-spaces of the three-dimensional space are

$$ax + by + cz + d \geq 0, \tag{3.4}$$

$$x^2 + y^2 - r^2 \leq 0 \tag{3.5}$$

where the planar half-space in Equation (3.4) consists of all points on or in front of the plane $ax + by + cz + d = 0$ and where the cylindrical half-space in Equation (3.5) consists of all points on or in an infinite cylinder with the rotation axis z and the radius r. Because half-spaces are point sets, they can be combined with the Boolean set operations union, intersection, and difference. The Boolean combination of two valid half-space models always defines a new valid model. Solids that consist of convex point sets can be described as the intersection of half-spaces. In order to represent nonconvex solids, a decomposition of convex sub-solids that compose the entire solid can be united with the Boolean union operation. In general, any valid solid S that is constructed by the combination of the half-spaces H_{ij} can be expressed in the canonical form

$$S = \bigcup_i \bigcap_j H_{ij} \,. \tag{3.6}$$

The expressive power of half-space modeling is limited to the set of available half-spaces and operators. A real benefit is that fairly complex shapes can be readily represented with a small number of half-spaces. However, as half-spaces specify infinite point sets, half-space modeling is not very intuitive to humans and, moreover, special care must be taken that a resulting solid is valid in terms of finiteness.

A technique that avoids such problems by modeling solids with only bounded point sets as primitives, is constructive solid geometry (CSG). Here, a solid is represented as a Boolean combination of primitives on which rigid motions can be applied. To guarantee that the resulting solids are always regular (in the sense that they do not contain any isolated point, line or face), regularity-preserving variants of the Boolean set operations are usually used. For this, the regularized set operations of union, intersection, and difference are defined as

$$A \cup^* B = c\big(i(A \cup B)\big), \tag{3.7}$$

$$A \cap^* B = c\big(i(A \cap B)\big), \tag{3.8}$$

$$A \setminus^* B = c\big(i(A \setminus B)\big) \tag{3.9}$$

where $c(X)$ and $i(X)$ denote the closure and the interior of a point set X. A CSG model is usually represented in a tree structure whose leaf nodes represent the primitives and whose internal nodes represent either a regularized Boolean set operation as defined in the equations (3.7) to (3.9) or a rigid motion in form of a translation or rotation. For CSG trees, set operations are algebraically closed and represent valid solids, as long as their leaf nodes define valid primitives.

In summary, constructive modeling offers an easy and natural way to define many objects with low and high complexity. It is especially suited to describe mechanical engineering parts and other man-made objects precisely and compact without a large storage space. Furthermore, every half-space or CSG model represents a valid solid object, assuming that the comprising primitives are valid. For rendering purposes, however, constructive modeling is less suitable because the boundary evaluation is intricate so that local blending operations are not easy to implement. Thus, it is appropriate to convert constructive models for visualization purposes in a polygonal form as described in the following subsection.

3.3.3 Boundary Models

In boundary representation (B-rep), solids are modeled through the representation of their bounding surfaces. For this purpose, a B-rep model is composed of the two basic parts: topology and geometry. The topological description defines the structure of the shape by the connectivity and orientation of faces, edges and vertices. These topological elements are embedded in space by the geometric description such that a face is a bounded portion of a surface, an edge is a bounded portion of a curve and a vertex is located at a point. For the modeling of a closed 2-manifold solid, a B-rep is valid if it defines the complete boundary of the solid and satisfies the following conditions:

1. Each face has at least three edges, a boundary that consists of a simple polyline which is closed and located in a plane, and no intersection with any other face except at common vertices or edges.
2. Each edge has exactly two vertices, no intersection with another edge except at a common vertex, and belongs to exactly two faces.
3. Each vertex of a face belongs to exactly two edges of the face.

A boundary model that only consists of connected planar faces is the polyhedral model. Every simple 2-manifold polyhedron, which is topologically equivalent to a sphere, satisfy the Euler characteristic

$$V - E + F = 2 \tag{3.10}$$

where V, E, and F are the number of vertices, edges, and faces, accordingly. Many different data structures have been developed to represent polyhedral models and that utilize the Euler characteristic to reduce the storage space or to enable an efficient access to certain information.

The simplest form to represent a simple polyhedron is a set of faces, each defined by a sequence of coordinates that form a planar polygon. Models of this kind of B-rep data structures are called polygon-based boundary models. The orientation of a face in a polygon-based boundary model is often implicitly determined by the order of the coordinates in the sequence.

The representation of a set of faces as sequences of coordinates has the drawback that a coordinate appears as often as the vertex appears in the set of faces. To eliminate this redundancy, vertices in vertex-based boundary models are explicitly represented as independent entities with distinct identifiers. As a result, faces are specified by a sequence of vertex identifiers so that each coordinate appears only once in the representation. There are many variants of vertex-based boundary models that mainly differ in which further connectivity information, for example incident edges or faces, are explicitly stored in the vertices and which are left implicit.

Analogously, edges are explicitly represented as independent entities in edge-based boundary models. Therefore, a face is no longer defined by a sequence of vertices but by a closed sequence of edges. If all faces are consistently oriented, the face orientation can be implicitly determined by the edge order. Three of the most common data structures for edge-based B-reps are summarized in the following paragraph.

One well-known edge-based B-rep is the winged-edge data structure (Baumgart, 1972; Baumgart, 1975). It utilizes the fact that each edge in the polyhedral B-rep of a 2-manifold solid occurs in exactly two faces, once in its positive orientation and once in its opposite orientation. In addition to the start and the end vertex, each edge explicitly keeps the information on its two incident faces and the four adjacent edges that occur in the incident faces and form thus the so-called wings. Moreover, each vertex and each face is linked to exactly one of its incident edges. This data structure is especially suitable for certain questions concerning adjacency relations between vertices, edges and faces. For instance, the winged-edge data structure can efficiently provide the adjacency information between two faces, which is often needed in hidden surface removal and shading algorithms. Furthermore, it provides a data structure whose storage space can be calculated in advance and remains fixed and that is also sufficient to model curved surfaces. Another common edge based B-rep is the half-edge data structure (Weiler, 1985). It decomposes each edge into two directed half-edges with opposite orientations. Thereby, the orientation issue of the winged-edge data structure, i.e. that the vertices of an edge have to be traversed for some faces in the opposite direction, is solved. The consistent traversing among the edges of any face makes some operations more efficient. In addition to the start vertex and the companion half-edge, each half-edge keeps the information on the face that traverses the half-edge in a positive direction and the two adjacent half-edges that belong to this face. Analogous to the winged-edge data structure, each vertex and each face is linked to exactly one of its incident edges. A data structure for edge-based B-reps that follows a similar approach as the winged-edge data structure by splitting each edge into four quad-edges is the quad-edge data structure (Guibas and Stolfi, 1985). In contrast to the two previously mentioned data structures, quad-edges are able to model non-orientable 2-manifolds as well. A comparison between these three data structures is presented in (Kettner, 1999).

Compared to decomposition and constructive models, boundary models are particularly suitable for the accurate description of free-form surfaces and for efficient rendering algorithms. In addition, local modification of a model can be directly implemented in B-reps.

Since B-reps are usually not closed under the Boolean set operations, Euler operators (Baumgart, 1972) are widely used, which are closed and satisfy the generalized Euler-Poincaré formula

$$V - E + F = 2(S - H) + R \tag{3.11}$$

where V, E, F, S, H, and R are the number of the vertices, edges, faces, shells, holes and rings, respectively. However, the object definition tends to be complicated and the verification of validity in terms of geometric correctness is difficult. Furthermore, the storage space may become large because both geometric and topological constrains have to be stored.

3.4 Data Clustering

With the advances in hardware technology and the collection of large amounts of data, the reduction of data without losing relevant information becomes of particular importance. One technique for this purpose is data clustering which has been extensively investigated in data mining, pattern recognition and machine learning communities since several decades. The basic problem of clustering can be formulated as the partitioning of a given set of data objects into a set of groups in which the data objects share common characteristics. In this section, a brief overview of different data clustering types and their characteristics are presented. For a more detailed description and concrete standard clustering algorithms see, for example, (Gan et al., 2007), (Xu and Wunsch, 2009), and (Everit et al., 2011).

Generally, data clustering methods can be categorized as either hard clustering or fuzzy clustering. In fuzzy clustering, which is based on the fuzzy set theory as introduced in (Zadeh, 1965), data objects can be assigned to several clusters. More specifically, given a data object set X and a cluster set C, each data object $x_j \in X$, where $1 \leq j \leq |X|$, is assigned to a cluster $c_i \in C$, where $1 \leq i \leq |C|$, with a certain membership coefficient $p_{i,j}$ that indicates the degree of membership of the data object x_j to the cluster c_i. The resulting memberships of fuzzy clustering can be described by the $|C| \times |X|$ membership coefficient matrix

$$P = \begin{pmatrix} p_{1,1} & p_{1,2} & \cdots & p_{1,|X|} \\ p_{2,1} & p_{2,2} & \cdots & p_{2,|X|} \\ \vdots & \vdots & \ddots & \vdots \\ p_{|C|,1} & p_{|C|,2} & \cdots & p_{|C|,|X|} \end{pmatrix}. \tag{3.12}$$

Usually, but not necessarily, membership coefficients in fuzzy clustering follow the following two rules of probability theory:

$$p_{i,j} \in [0, 1], \tag{3.13}$$

$$\sum_{i=1}^{|C|} p_{i,j} = 1, \forall x_j \in X. \tag{3.14}$$

Since fuzzy clustering allow for overlapping clusters, their algorithms are usually advantageous in cases in which cluster boundaries are ambiguous and in which clusters are difficult to separate from each other. In addition, the membership coefficient matrix can easily be updated if further information is to be considered. For example, in cases in which the membership coefficients follow the rules of probability theory, the Bayes' theorem, as mathematically stated in Equation (3.15), can be directly applied if the current membership probability does not equal zero.

$$P(A|B) = \frac{P(B|A)\,P(A)}{P(B)} \tag{3.15}$$

Furthermore, fuzzy clustering algorithms provide a sufficient functionality for the description of uncertainties that often occur in real data. However, the amount of generated information in the membership coefficient matrix increases rapidly with an increasing number of clusters and data objects so that the interpretation may take considerable computation time. A survey of fuzzy clustering methods and their application are presented in (Baraldi and Blonda, 1999; Höppner et al., 1999; Sato et al., 1997; de Oliveira and Pedrycz, 2007).

In contrast, each data object in hard clustering is assigned to exactly one cluster. Consequently, it can be considered as a limiting case of fuzzy methods where the membership coefficients follow the constraints represented in Equation (3.14),

$$p_{i,j} \in \{0,1\}, \tag{3.16}$$

$$\sum_{j=1}^{|X|} p_{i,j} > 0, \forall c_i \in C. \tag{3.17}$$

These more restrictive constraints ensure that no empty clusters occur (each cluster contains at least one data object) and that, at most, as many clusters exist as there are data objects. In general, hard clustering methods can be further categorized as either partitional methods or hierarchical methods. In partitional methods, a set of data objects is partitioned into a predefined number of disjoint clusters so that the distance between each data object and the center of the cluster to which it belongs is minimal according to a distance function. One of the best-known partitional clustering method is the k-means algorithm (Lloyd, 1957; MacQueen, 1967) whose objective is to iteratively find a partitioning that assigns data objects to k clusters while minimizing the equation

$$E = \sum_{i=1}^{|C|} \sum_{j=1}^{|X|} \left\| x_j - \mu_i \right\|^2 \tag{3.18}$$

where μ_i represents the center of the i-th cluster. Partitional clustering is usually applied if a set of data objects should be assigned to a fixed number of disjoint clusters because the

computational time and the storage space is usually less than for hierarchical clustering algorithms.

In hierarchical clustering, clusters are represented in a hierarchical structure so that the result can be illustrated in a binary tree, which is called dendrogram. There are two major types of hierarchical clustering methods: agglomerative clustering and divisive clustering. Both have in common that particular care has to be taken during the iterative generation of the hierarchical structure because a wrong cluster fusion or merging can never be repaired during the subsequent iterations. In contrast to standard k-means algorithms, however, hierarchical clustering algorithms are deterministic so that they always provide the same clustering results on the same data set (Aggarwal and Reddy, 2014).

In agglomerative clustering, a bottom-up approach is used for the construction of the cluster hierarchy, whereas divisive clustering is based on a top-down approach. Agglomerative clustering methods starts therefore with each data object in a single cluster. Subsequently, clusters are iteratively merged based on a linkage metric which indicates the distance between clusters. There are several ways to define the distance between two clusters. A common one is the single linkage method (Florek et al., 1951; Sneath, 1957) that defines the distance d between two clusters C_{i_1} and C_{i_2} by

$$d\left(C_{i_1}, C_{i_2}\right) = \min d(a, b) \tag{3.19}$$

where $a \in C_{i_1}$ and $b \in C_{i_2}$. Because d represents the minimum of all pairwise distances between two clusters, it is also called the nearest neighbor method. Other common linkage metrics are, for example, the complete linkage method (McQuitty, 1960), which uses the largest distance between a and b, Ward's method (Ward Jr., 1963), which uses the minimum variance, and further metrics based on the cluster centroids or their medians. In practice, the computational effort of agglomerative clustering has to be taken into account because the number of merge combinations in the first iteration amounts to $\frac{|X| \cdot (|X|-1)}{2}$. Thus, the computational complexity of agglomerative hierarchical clustering algorithms is generally at least $O(n^2)$ and therefore not particularly suitable for large sets of data objects.

In divisive clustering methods, on the other hand, all data objects are initially assigned to the same cluster. Then, a cluster is split in each iteration based on the resulting distance between the resulting clusters until each cluster contains only a single data object. It is interesting to note that agglomerative and divisive hierarchy do not necessarily have to coincide. In the literature and in practice, divisive clustering techniques are often neglected (Kaufman and Rousseeuw, 2005) because the computational effort is even greater than for agglomerative clustering methods. An obvious reason for this is that all data objects belong in the first iteration to the same cluster so that $2^{(|X|-1)} - 1$ split combinations have to be considered. Consequently, the computational complexity of divisive hierarchical clustering algorithms is generally at least $O(2^n)$. It is therefore rather important to reduce the computational time, for example, by reducing the number of possible divisions in each iteration.

3.5 Registration of Point Clouds

For several decades, laser scanners have been used in an efficient way for the acquisition of point clouds that represent the surface of 3D objects (Vosselman and Maas, 2010). To cover the entire surface of an object, data from multiple standpoints are often collected. Because each standpoint is defined in its own local coordinate system, all data sets have to be transferred into a common coordinate system after collection. The alignment of one data set to another data set is a common problem in 3D scanning and is called registration. Many registration methods have been developed to register surfaces with an overlapping area so that the resulting transformation best represents their relative displacement; examples are given in (Silva et al., 2005). In general, those registration approaches that neither rely on the position and orientation of the laser scanner (direct geo-referencing) nor on manually placed targets, can be roughly divided into coarse and fine registration techniques.

In coarse registration methods no initial transformation or other prior knowledge of the relative spatial position is required. The general workflow consists of three steps: First, certain key points or features are detected from the input data sets to reduce the number of points that have to be considered during the registration process. Then, local shape descriptors assign values to the detected key points and features by taking into account their own properties and the properties of the shape around them. Finally, a search strategy is carried out to find correspondences between the key points and features of different data sets. Based on the detected correspondences, a transformation for each data set can be determined that optimally aligns the data set with respect to the others. A qualitative review on 3D coarse registration methods and some implementations of each step are given in (Díez et al., 2015; Salvi et al., 2007). Depending on the number of detected key points and features and the assignment certainty, these registration techniques usually provide only rough alignments and are therefore used as initial estimations. For a more precise alignment, this prior knowledge regarding initial transformations is then further used in fine registration approaches.

In contrast, fine registration methods first need a rough initial transformation before the transformation matrix is iteratively refined until the value of a predefined distance function, which considers distances between temporal correspondences, converges to a local minimum. One of the most common methods is summarized in the following paragraphs. Some further well-known fine registration methods are among others Chen's method (Chen and Medioni, 1991), which minimizes the distance between point and planes, and methods that are based on genetic algorithms (Chow et al., 2004) or signed distance fields (Masuda, 2002).

Iterative Closest Point (ICP) is an algorithm employed to register two (partially) overlapping point sets in a common coordinate system (Besl and McKay, 1992). For this purpose, it iteratively minimizes the mean squared error (MSE) between point sets by applying rigid transformations. Compared to other fine registration techniques, an exact point-to-point correspondence between the point sets is not required. Today, there exist many variants of ICP. An overview is given in (Rusinkiewicz and Levoy, 2001), (Grün and Akca, 2005), and (Pomerleau et al., 2015).

Essentially, for two given point sets A and B, which are already roughly registered, each iteration in ICP is composed of the following steps:

1. For each point, find in A the closest point in B.
2. Estimate a transformation that minimizes the MSE of the point pairs from the previous step.
3. Transform all points in B by applying the obtained transformation.
4. Start the next iteration from the first step if a predefined stopping criteria is not fulfilled.

In the first step, the correspondence search between points of A to B is performed by a nearest-neighbor search. To improve the speed of ICP, several approaches have been developed to increase the performance of computing corresponding points, because this is the most time-consuming part of ICP. Usually k-d trees (Bentley, 1975) are used to accelerate the search. Several variants of the k-d tree have been developed to improve the performance like a cached (Nüchter et al., 2007) or a GPGPU (Wu et al., 2015) version.

For the second step of each iteration, a rotation and translation is computed that minimize the MSE between both point sets. The objective function that has to be minimized for two given point sets $A = \{a_i\}$ and $B = \{b_j\}$ with $i = \{1, \dots, n\}$ and $j = \{1, \dots, m\}$ is therefore

$$f(R,t) = \frac{1}{n}\sum_{i=1}^{n}\left\|a_i - Rb_j - t\right\|^2 \rightarrow min \tag{3.20}$$

where b_j is the closest point in B to the point $a_i \in A$, R the rotation matrix, and t the translation vector. For the minimization of the objective function and the calculation of the rigid transformation, several methods exist; for example, singular value decomposition is used in (Arun et al., 1987), orthonormal matrices in (Horn et al., 1988), and dual quaternions in (Walker et al., 1991).

Afterward, the resulting rotation and translation can be directly applied in the third step to all points in B. Then, the algorithm starts again from the beginning if a lower MSE value than in the iteration before could be received. Otherwise the algorithm terminates and a sequence of transformations is given that can be merged to a single rigid transformation, which minimizes the MSE with the final point correspondences at once.

Due to the greedy nature of the ICP algorithm, it converges always to a local minimum but not necessarily to the global minimum. Therefore, to get useful results, it requires a good coarse a priori alignment of the given point clouds. Furthermore, the method as presented above is limited to rigid transformations (rotations and translations), so that, for example, it is not capable of handling shape morphing as shown in (Alexa, 2002) or scaling as shown in (Jiang et al., 2009). For further nonrigid registration methods see (Kumar et al., 2001), (van Kaick et al., 2011), and (Tam et al., 2013).

4. Roof Plane Segmentation

Since most buildings consist of approximately planar roof surfaces, several methods have been developed to segment building points into planar roof areas (see section 3.1). The resulting roof patches are generally well suited to support the reconstruction process of buildings, especially if only simple buildings without any roof superstructures (e.g., dormers or chimneys) are present. However, these patches increase in number, while becoming smaller, and at the same time more intricate in their shapes as the complexity of the roof structure increases. Each roof superstructure adds a hole or a concavity to the underlying roof segment or even divides it into two parts so that roof features become difficult to recognize. In great numbers, this could cause subsequent steps to fail in generating well-shaped building models even at their most basic level of detail. Therefore, a new segmentation method has been developed that overcomes these problems and supports the modeling and reconstruction process of buildings.

Current research activities concentrate on improving the quality of point cloud segmentation in order to get precise and consistent roof segments, which eases the interpretation and modeling stages. However, the obvious assumption that the captured points are surface measurements of an underlying volumetric object is rarely exploited. For the reconstruction of base roof shapes, it greatly helps if no roof superstructures are present at data collection. The point cloud would feature only large roof segments without any holes, which would make the reconstruction of a valid 3D building model a lot easier. Illustrated in the example of an airborne laser scanning measurement: if roofs would not feature any roof superstructures, then the laser would instead hit underlying roof faces, thus resulting in uniform roof segments. Changing reality is for obvious reasons not feasible, but the effect can be simulated in the segmentation process. The basic idea of the developed sub-surface segmentation method is therefore that virtual points may be added to any segment if this supports the reconstruction tasks. Although the number of points is generally unrestricted, they must be located under real point measurements. Meaning that they have the same horizontal coordinates as real surface points of the point cloud, but must be positioned at a lower elevation within the modeled volume. These virtual points are therefore called sub-surface points.

The proposed sub-surface segmentation has been implemented as an extension of the well-known surface growing approach; for details, see e.g., (Vosselman and Klein, 2010), thus the reason for introducing the term sub-surface growing. However, segmentation algorithms based on the 3D Hough transform in conjunction with a connected component analysis can also be extended in a similar way. The general workflow of the implemented sub-surface growing process consists of two major phases that are illustrated in Figure 4.1.

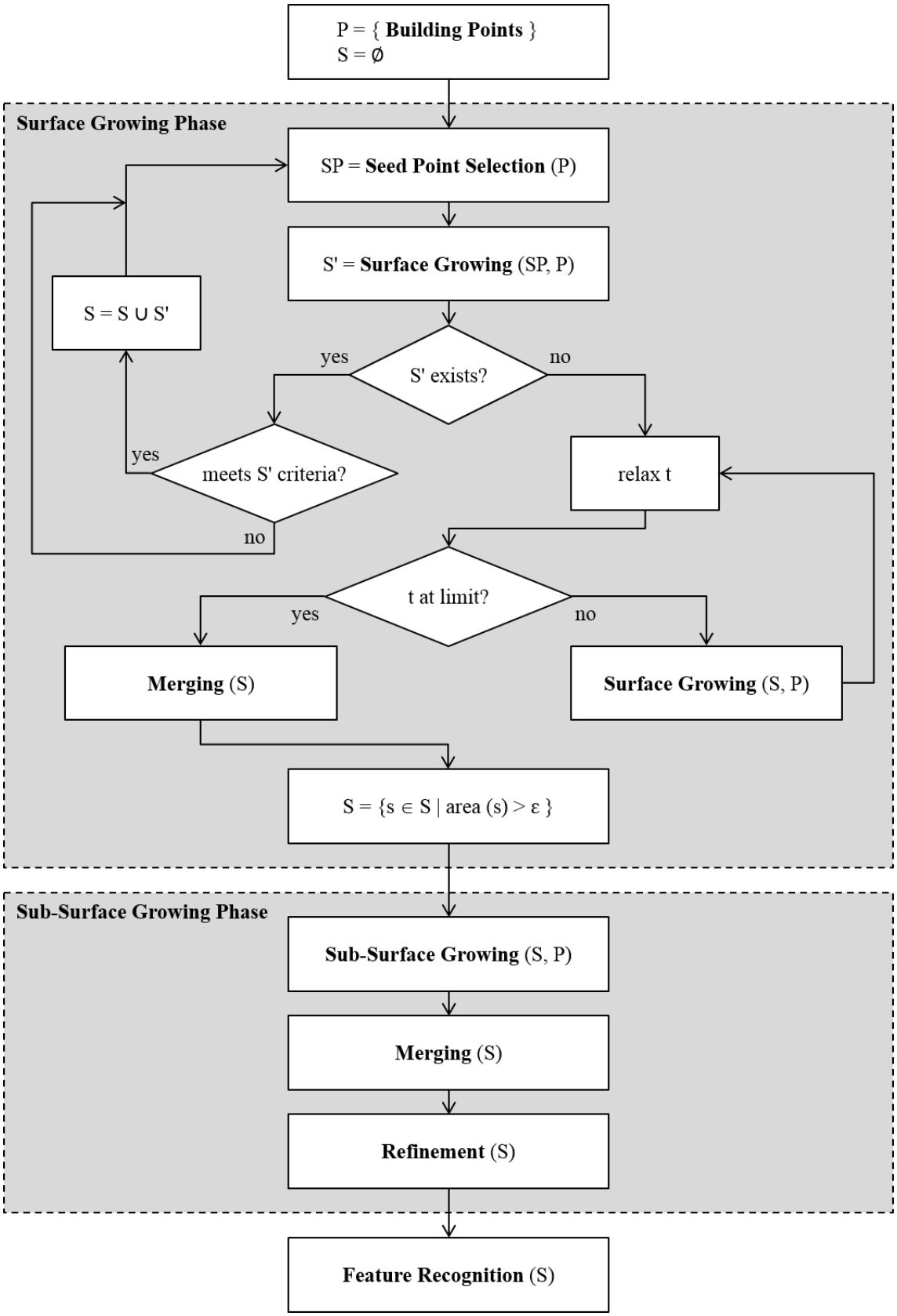


Figure 4.1. The sub-surface segmentation process (P = set of building points, S = set of segments, SP = seed point, S' = segment, t = thresholds for surface growing).

The details of the surface and the sub-surface growing phase are explained as well as some segmentation results for a large urban area are shown in sections 4.1 and 4.2, respectively. A more detailed investigation of the benefits of sub-surface growing is then presented in section 4.3.

4.1 Surface Growing

The surface growing process of planar areas starts with the selection of a suitable seed point from which segment growing can be initiated. A point is generally suited as seed point if it can be located on a plane so that its nearby points essentially fit to this plane. In order to determine such a point, all building points are initially considered as potential seed points. For each of them, the set of nearby points is identified based on a fixed radius and their natural neighborhood as defined by the two-dimensional Voronoi diagram. Thus, points are defined as natural neighbors if they are connected by an edge in the corresponding dual graph of the Voronoi diagram, which is the Delaunay triangulation. Since Delaunay triangulation only considers the empty circle property, which is that the circumscribing circle of any triangulation facet encloses no other points, the minimum angle of all triangles in the triangulation is maximized without considering the edge length of the triangles. Consequently, adjacent points may have a greater distance to each other than in a neighborhood with only a fixed radius. This is, for example, the case if gaps in the input data are present. Thereby, it is also taken into account that a point cloud might have different point densities, for example, due to some overlapping strips, and that neighboring points might be separated by a step line which causes a great distance. Once the set of nearby points is determined for each potential seed point, all those points are discarded as potential seed points whose set of nearby points feature only a small number of points. In this way, the subsequent estimate of planes becomes more reliable because sets with a small number of points usually provide in practice insufficient information for this task. As a consequence, points close to the building outline are removed from the set of candidate seed points so that the seed points are enabled in the next stage to grow in any direction.

In order to select the most suitable seed point from the set of potential candidate points, an evaluation process is started that rates each potential seed point according to the standard deviation of its set of nearby points to a plane that passes through the seed point and whose normal vector is determined by the plane with the least average distance to the set of nearby points. The normal vector of such a plane is calculated by utilizing PCA (Pearson, 1901; Hotelling, 1933); for further details about PCA see, for example, (Jolliffe, 2002). Generally, PCA is a multivariate statistical procedure to identify the smallest number of uncorrelated variables from a set of data that explains the maximum amount of variance. These uncorrelated variables are called principal components and are the eigenvectors of the covariance matrix of a data set. The covariance matrix for a set of three-dimensional points is defined as

$$C = \begin{pmatrix} cov(X,X) & cov(X,Y) & cov(X,Z) \\ cov(Y,X) & cov(Y,Y) & cov(Y,Z) \\ cov(Z,X) & cov(Z,Y) & cov(Z,Z) \end{pmatrix} \tag{4.1}$$

where the covariance for any two random variables X and Y is

$$cov(X,Y) = \frac{1}{n}\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y}) \tag{4.2}$$

and $\bar{x}$ and $\bar{y}$ the mean values of X and Y, respectively. Since $cov(X,Y)$ is equal to $cov(Y,X)$, the covariance matrix is symmetric positive (semi-)definite, hence the eigenvectors are orthogonal to each other. By taking into account that the corresponding eigenvalue of a principal component indicates the amount of variance in the direction of the eigenvector, the normal vector of the sought plane corresponds to the eigenvector with the lowest eigenvalue. For a given point and its nearby points, the direction of this eigenvector is called the normal direction of the given point.

All candidate seed points that feature point normal directions with low z-values are at this point discarded as potential seed points because they are assumed to be part of a vertical plane, i.e., a facade, which are not needed for the further reconstruction process. For the final seed point selection, the squared standard deviation σ^2 is calculated for each remaining candidate seed point based on the perpendicular distances D of its neighboring points to the plane that is defined by the point itself and its point normal direction as

$$\sigma^2 = \frac{1}{n-1}\sum_{i=1}^{n}(d_i - \mu)^2 \tag{4.3}$$

where $d \in D$, $n = |D|$, and $\mu = E(D) = 0$. The potential seed point with the lowest squared standard deviation that has not been selected as seed point and that has not been assigned to any segment before is then selected as seed point for the next step.

In the subsequent surface growing step, the selected seed point is grown to a planar segment by gradually adding neighboring points if they do not already belong to any other segment and if they fit some geometric criteria. For the latter, both a distance criterion and an angle criterion are used. The distance criterion ensures that the perpendicular distances of all added points are close to the plane that is defined by the seed point and its point normal direction. The angle criterion, on the other hand, guarantees that all point normal directions of a segment point in the same direction. This is especially reasonable if greater distances are accepted in the distance criterion. With appropriate thresholds, both criteria are able to deal with small inaccuracies of data points and with the fact that roof areas are usually not exactly planar.

Once no more neighboring points can be added, the resulting segment is checked and discarded in cases in which it consists of too few boundary points or in which it reflects a degenerated shape. For the investigation of the first condition, the boundary points of each segment are computed by utilizing the alpha shape algorithm (Edelsbrunner et al., 1983). An alpha shape can be regarded as a generalization of the convex hull of a point set but it has the advantages of being able to be used for both convex and concave shapes and for extracting

polygons with interior and exterior boundaries (Shen et al., 2011). As mentioned in (Dorninger and Pfeifer, 2008; Shahzad and Zhu, 2015), the alpha shape value can be adjusted to adapt the algorithm to different point cloud densities so that small details are preserved. In those rare cases where an alpha shape includes a topological irregularity (i.e., a boundary point has more than two incident edges), χ-shapes as proposed in (Duckham et al., 2008) are used as an alternative to alpha shapes. Here, special care has to be taken because outlines of χ-shapes become more sensitive to outliers than alpha shape outlines. For the second condition, an oriented bounding rectangle is calculated in the x-y plane for each segment. For this, a convex hull is first computed for each segment based on the extracted boundary surface points. It is implemented as Andrew's variant of the Graham scan algorithm (Andrew, 1979) and follows the presentation of (Mehlhorn, 1984). The algorithm requires $O(n \log n)$ time in the worst case for n input points. The resulting convex hull is then used to determine the minimum area enclosing rectangle of all boundary points based on a rotating caliper algorithm (Toussaint, 1983), which requires $O(e)$ time in the worst case for e edges of the convex hull. Shapes of segments are afterward classified as degenerated if one side of their oriented bounding rectangle is very short while the other is very long.

Subsequently, the surface growing procedure starts again with the selection of a seed point until no more segments can be detected. Note, the calculation of the point normal directions and the standard deviation values for the seed point selection step can be skipped in the next iterations because they only need to be calculated once. At the point when the segmentation process ceases, there are usually many unassigned points left, particularly in areas where a transition between planar roof surfaces is present. Points are still unassigned because their point normal directions do not meet the angle criteria of any adjacent roof surface. The reason for this is that these points feature sets of nearby points that belong to more than one planar roof surface so that they cannot be properly assigned to only one adjacent segment. Therefore, surface growing continues again for each segment, but the angle criterion is gradually relaxed from iteration to iteration until it reaches a defined limit. This allows the already detected segments to uniformly grow towards each other and points that are around edges are reliably assigned to the correct segments.

A key problem of segmentation methods is that planar regions are split into either too few (under-segmentation) or too many segments (over-segmentation). In both cases, the resulting segments need to be further processed. However, as for example mentioned in (Sigut et al., 2015), under-segmentation is often considered a more serious problem compared to over-segmentation because it usually requires the splitting of large regions in order to recover the true segments. Therefore, to avoid under-segmentation during the surface growing step, so that segments belonging to different roof surfaces are assigned to the same segment, strict thresholds are generally preferred. This causes, however, an over-segmentation where a planar roof surface may be segmented into several small roof patches. To overcome this issue, segments have to be merged afterward if they feature similar properties. For this, all segment pairs of close proximity are tested if their point sets feature similar plane normal vectors. In such a case, both segments are merged if the resulting normal vector has also not changed too much and if a sufficient number of points are still close enough to the plane of the merged

segment. Segments whose area remain small even after the merging process are finally discarded. The area of each segment, which is the area enclosed by the linearly connected boundary points, is calculated according to the so-called shoelace formula, also known as Gauss's area formula, which determines the area A of an arbitrary simple 2D polygon by

$$A = \frac{1}{2}\sum_{i=1}^{n} x_i(y_{i+1} - y_{i-1}) \tag{4.4}$$

where (x_i, y_i) is a corner point of the boundary that are defined cyclically (i.e., $(x_{n+1}, y_{n+1}) = (x_1, y_1)$), and n the total number of corner points. For further details see, for example, (Braden, 1986). Performed on the Vaihingen data set (see section 8.1), the surface growing procedure provides the segmentation result shown in Figure 4.2.

Figure 4.2. Segmentation result of surface growing performed on the Vaihingen data set.

4.2 Sub-Surface Growing

Once surface growing is completed, segmentation continues for each segment at a sub-surface level. For this purpose, virtual points are generated below real points as required in order to facilitate further growth of segments. Each virtual point must have the exact same horizontal position, but a lower elevation than a real point. This elevation is computed so that it lies exactly planar to the currently grown segment. The angle criterion is not checked during sub-surface growing, so the virtual points need only be located nearby the segment in order to be included. If more than one segment is to grow beneath a real point, then several virtual points may be generated. But sub-surface growing is also not unrestricted. Tests revealed that letting a segment grow too far above the maximum height or too far below the minimum height of the segment's real points does not reveal any valuable information and just slows the process down. The segment of a sloped roof face is therefore restricted to grow mainly sideways. Consequently, flat segments can spread unrestricted through the building as long as there are real points above it.

After adding virtual points to the segments, another segment merging step is carried out. Without differentiating between virtual and real points, two neighboring segments are merged if they are in close proximity to each other and if their normal vectors point roughly in the same direction. However, the merging is repealed either if the normal vector direction of the new segment is too dissimilar to the normal vector directions of the original segments or if too many points are too far away from the plane of the new segment.

Due to the greedy nature of surface growing, the order in which segments are generated during the surface growing phase usually affects the assignment of points around edges where segments meet. A typical example is shown on the left side of Figure 4.3, where the last detected red segment became small due to the first detected blue and orange segments. Therefore, all points of a segment that are close to its border are finally tested in a refinement step if they would better fit to a neighboring segment than to the current one. For this, each of these points is tested against its current and neighboring segment, on the one hand, in terms of its distance to both planes and, on the other hand, in terms of the angle between its normal direction and the normal vector of both segments. Additionally, if the point and both segments are part of a sloped surface, then the x-y directions of their normal vectors are also taken into account. This particularly helps to determine to which of the opposing segments a point should be assigned to. The proposed refinement step usually increases the processing time but the borders of the final segments become more accurate and reliable than before as shown on the right side of Figure 4.3.

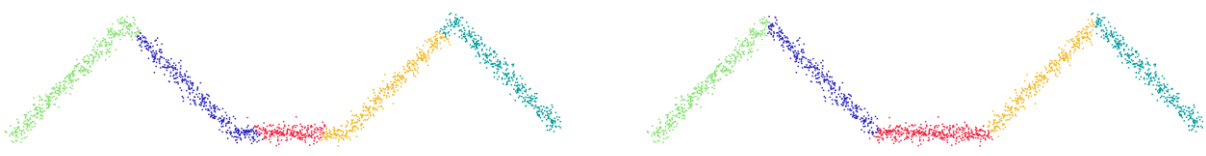


Figure 4.3. Segmentation results without (left) and with (right) reassignment of segment points; based on (Vosselman and Klein, 2010).

Performed on the Vaihingen data set, sub-surface growing provides the segmentation result shown in Figure 4.4. Compared to the surface growing segmentation result presented in Figure 4.2, there does not seem to be a big quality gain that would justify the additional computational time. But a closer look on the result as presented in the following section demonstrates the capabilities of sub-surface growing.

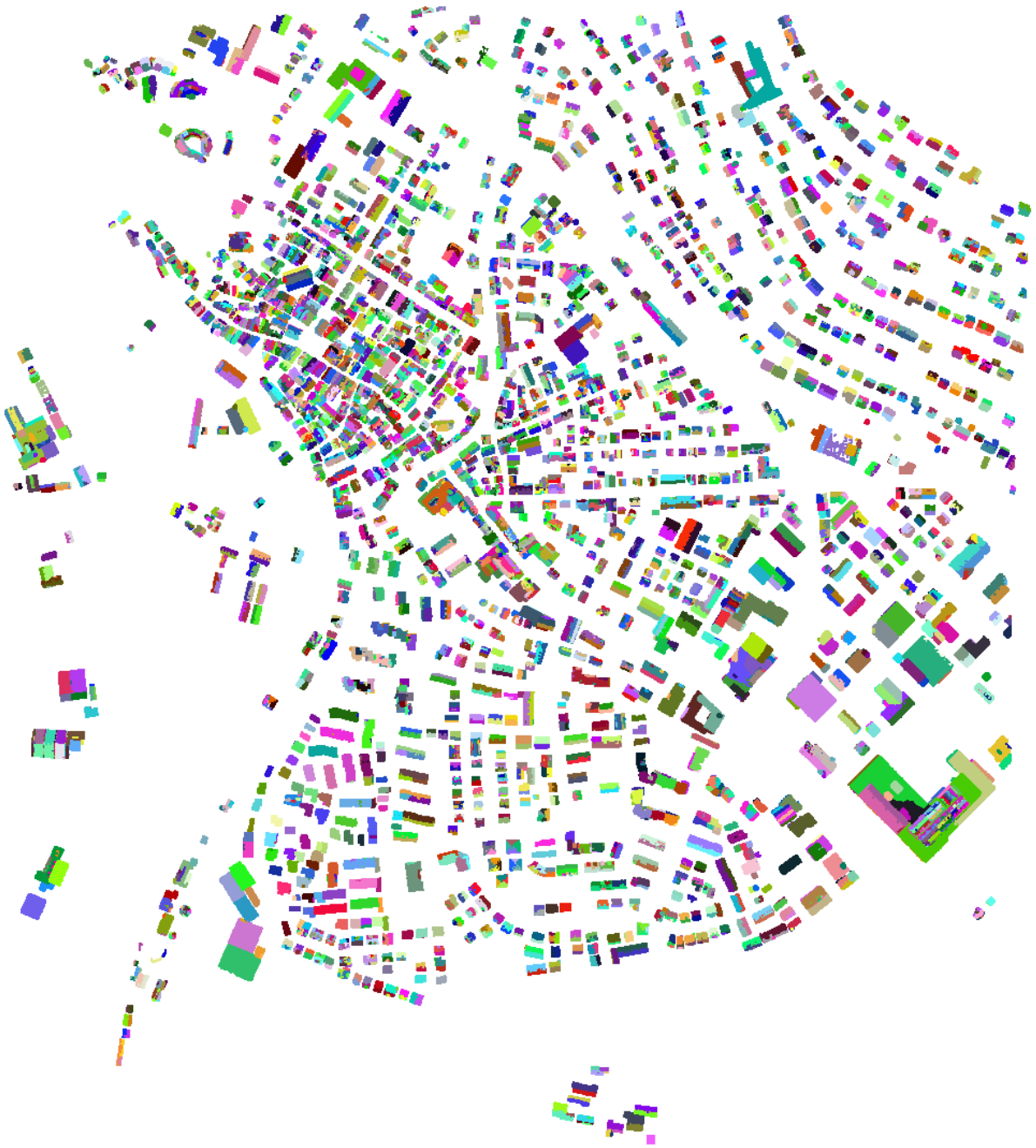


Figure 4.4. Segmentation result of sub-surface growing performed on the Vaihingen data set.

4.3 Benefits of Sub-Surface Growing

An obvious gain from sub-surface growing is the fact that holes in segments and gaps between disconnected roof sections that are the result of roof superstructures or other roof elements get filled with virtual points. The benefit of it becomes apparent, in particular during the identification of base roof shapes. As for example shown in Figure 4.5, base roof segments grow much larger than before, often by getting merged with other coplanar segments that are otherwise separated from one another due to superstructures or crossing roof parts. The resulting segments imply a higher accuracy of their estimated plane parameters as the points extend over a larger area and the merging of segments automatically establishes consistency between nearly coplanar segments.

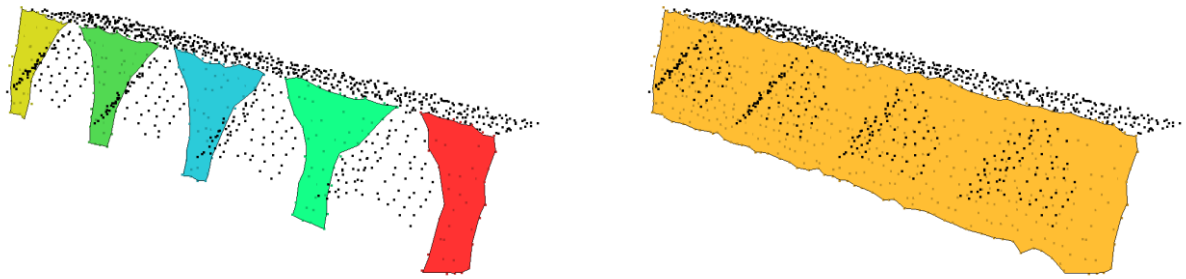


Figure 4.5. Planar segments of the front roof surface of the base roof as a result of surface growing (left) and sub-surface growing (right).

Additionally, the subsequent model construction is significantly simplified if it involves fewer and larger segments. With sub-surface growing and CSG for example, the construction of the building in Figure 4.6 is just the union of two elementary shape primitives: one gable and one barrel-arched roof. Thus, symmetries in building shapes are implicitly modeled. Regular segmentation, on the other hand, would result in two saddleback primitives, one to either side of the barrel-arched center. If model construction is performed in a B-rep, each primitive intersection can produce a significant amount of small faces if they are not properly aligned. This in turn can become tedious for complex buildings.

Figure 4.6. Segmentation of planar regions as a result of surface growing (left) and sub-surface growing (right).

Another important aspect is that points making up small details are now part of larger segments. Due to their low number, these points would usually not make up segments of their own, and the details are lost from reconstruction. An example for this is depicted in Figure 4.7. The points to the right side of the protruding hip are assigned to the adjacent segments, but sub-surface growing adds these points to the red segment at the back. As a result, the four segments can now reliably be identified and modelled as a hipped roof. Although in the end, the large perpendicular roof section to the right will occlude a large portion of the right side of the hipped roof part.

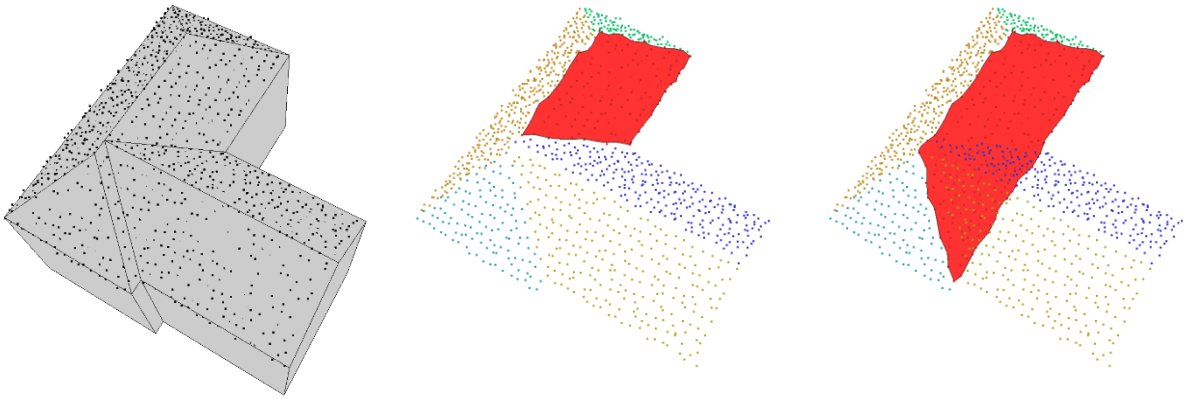


Figure 4.7. Building model overlaid with surface points (left), surface growing segmentation result (middle), and sub-surface growing segmentation result (right).

Due to sub-surface growing, segments are now closer to each other, share longer common boundaries or even intersect one another. The differences between surface growing and sub-surface growing segments are exemplarily shown for a hip and a mansard roof in Figure 4.8. Especially close to ridge and step lines, there is usually an increasing uncertainty as to which segment a point should be assigned to. Later stages of the reconstruction process strongly rely on adjacency information between roof segments, which is generally derived from the position and orientation of the segments using distance and angle thresholds and sometimes also a required minimum length for the potential intersection or step-edge lines. Erroneous point assignments can hinder finding such relations reliably without relaxing the values of the above-mentioned thresholds. However, this is generally not a viable solution as it also introduces the potential for false interpretations. But by adding virtual sub-surface points, all segments grow further towards the intersection lines and therefore closer towards each other. This enables a more reliable detection and precise tracing of ridge and step lines than without sub-surface points so that a reliable shape identification of roof elements is supported. Consequently, more complex and stringent rules with much higher certainty can be applied to define adjacencies between segments or roof parts.

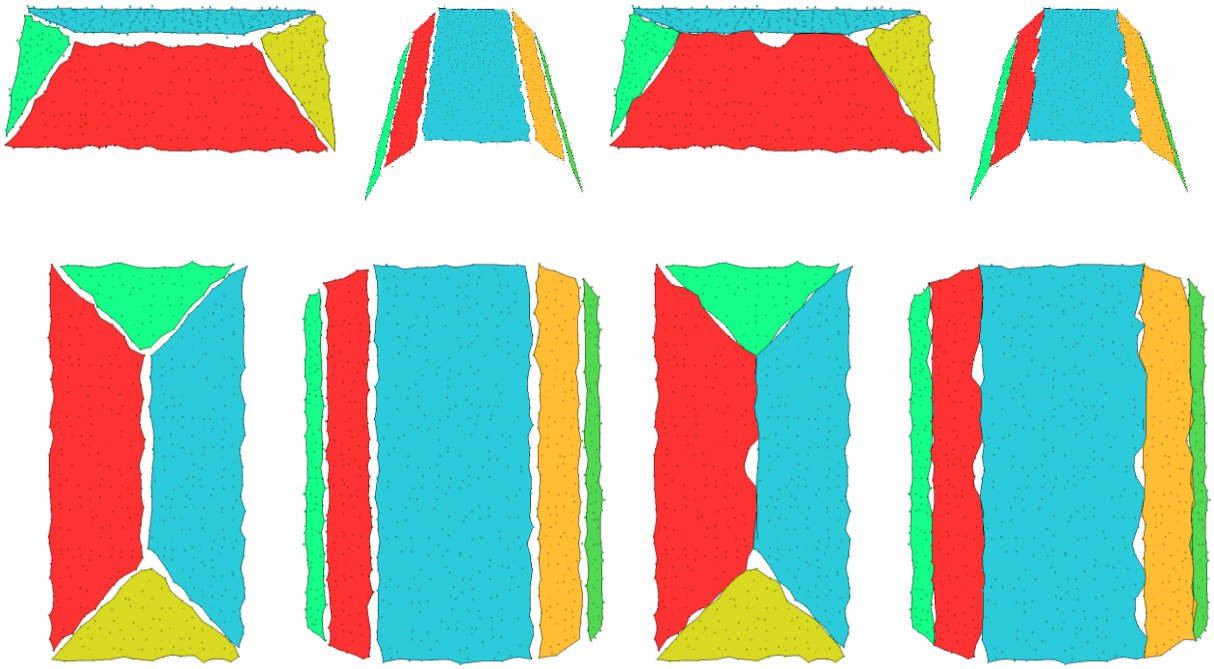


Figure 4.8. Resulting segments of surface growing (left) and sub-surface growing (right) from perspective (top) and top view (bottom) for a hip and a mansard roof.

It is also important to stress that segments from sub-surface growing might become adjacent below surfaces. As, for example, shown in Figure 4.9, without sub-surface points, the segmentation of the red and the blue segment would stop in front of the tower-like superstructure. Consequently, their adjacency would be rarely exploited and each of the saddleback roofs independently constructed. In contrast, with sub-surface points the corner configuration can be reliably identified and constructed even though the tower-like superstructure (olive green points) occludes all of the corner itself. Thus, gaps between the gable roofs and the tower-like superstructure are already closed in the segmentation level.

As with segments, ridge lines become continuous and are not broken into shorter line segments. With sub-surface growing, intersection lines are in general longer and if a lower gable roof or a dormer adjoins a larger roof face, then its ridge line does now in fact intersect the segment of the base roof. It does not end in front of it, which would require the use of a distance threshold and leads to difficulties to tell the case apart where the roof is not adjoining. An example that illustrates the differences between surface and sub-surface growing is given in Figure 4.10.

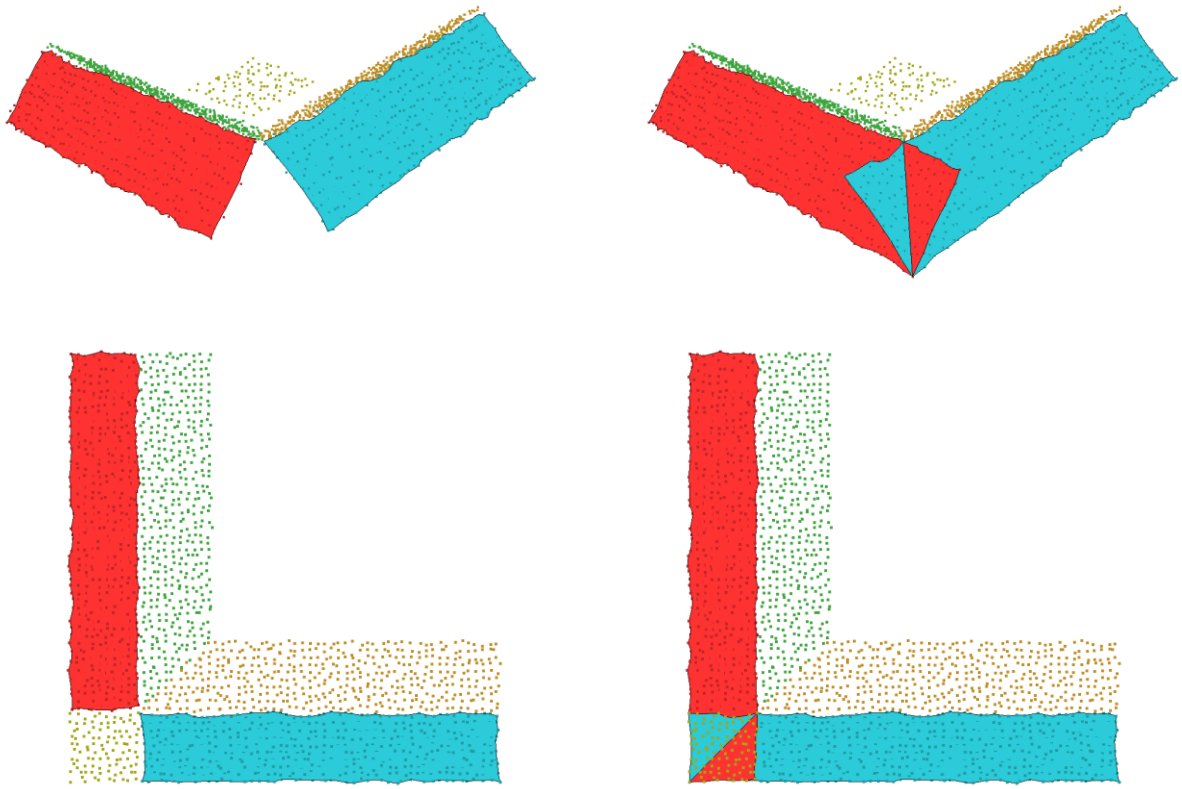


Figure 4.9. Segmentation result for an occluded building part. Once with surface growing (left) and once with sub-surface growing (right) from perspective (top) and top view (bottom).

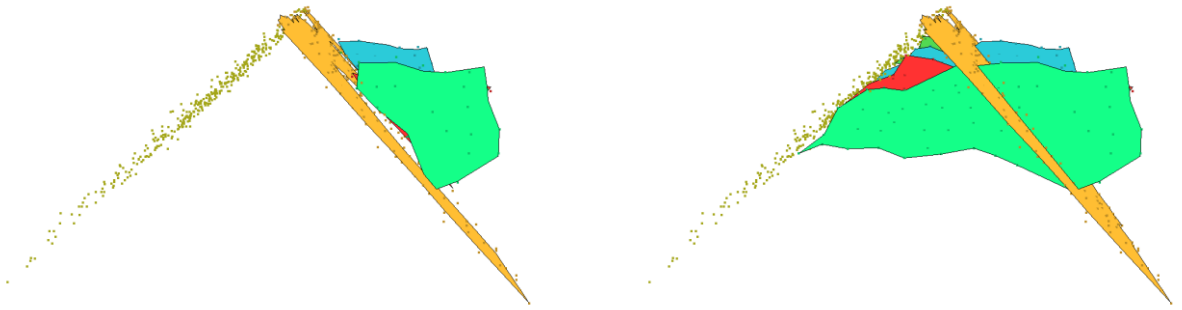


Figure 4.10. In contrast to dormer segments derived from surface growing (left), sub-surface growing dormer segments (right) intersect the underlying segment.

Moreover, many buildings possess dormers of the same type that are symmetrically arranged on their rooftops. The segmentation of dormer roof areas is usually difficult because they comprise only a small number of points. Common segmentation methods are therefore often either not able to detect them or the resulting segments can feature only unstable planes. In contrast, sub-surface growing connects coplanar dormer roof surfaces of the same base roof by growing through the common base roof. The resulting segments therefore feature more points so that more reliable planes can be derived than without sub-surface growing. In this way, typical regularities in a building are directly supported. The example in Figure 4.11

illustrates the differences between surface and sub-surface growing for coplanar dormer segments.

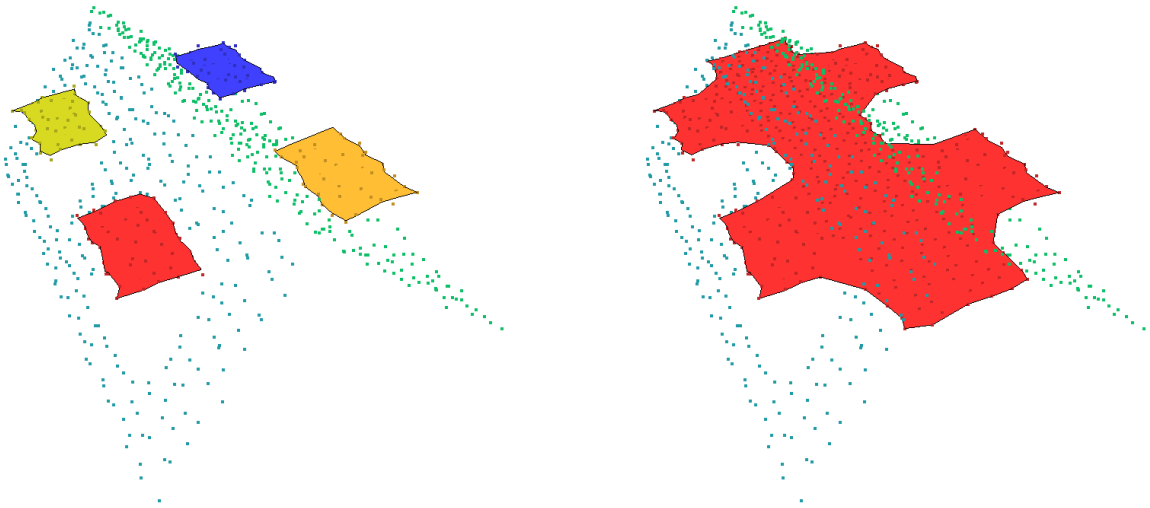


Figure 4.11. Dormer segments from surface growing (left) and sub-surface growing (right).

In conclusion, the presented sub-surface growing method enables lower segments to grow beneath higher ones. Thereby, fewer and larger segments are generated that grow closer together, preserve small roof details, and implicitly support typical building regularities. Thus, the identification of roof elements and their composition to more complex shapes becomes more reliable in later stages.

4. Roof Plane Segmentation

5. Building Knowledge Derivation

In the task of automatic 3D building model reconstruction from airborne laser scanning point clouds, the problem has to be faced that the set of planar roof areas might be only inadequately represented by the set of planar segments resulting from an automatic roof plane segmentation process. Therefore, building models are not directly constructed in this thesis from a set of planar segments but by additionally taking further building knowledge into consideration. Here, the focus is on topological and semantic building information rather than only on geometric information. In this way, invalid building models and unnatural roof shapes are avoided during the reconstruction stage. The automatic derivation process of building knowledge is presented in this chapter.

For the automatic extraction of building knowledge from a set of building points that are assigned to planar segments, several building features are defined that commonly occur in rooftops. Their presence and recognition in the set of planar segments supports, on the one hand, the detection of further building features and, by incorporating their semantic information, significantly helps to avoid the reconstruction of building parts that do not occur in reality. On the other hand, regularities in the final building model can be easily emphasized with regard to both geometric and semantic information so that the resulting building models are pleasing to the human eye. Generally, the set of predefined building features can be adapted to specific buildings or locations. Further building features can be added or ignored at any time without the need for changing other existing building features to guarantee that the resulting models still represent valid solids. However, in cases where only an incomplete subset of building features is considered, it is not guaranteed that the resulting model represents the entire building. In order to organize all recognized building features and their relationships to each other, the multi-scale knowledge graph has been developed. It provides a model for knowledge representation and object categorization in which all recognized building features are organized and in which their relationships and interdependencies are well represented. The details of the proposed multi-scale knowledge graph and some building feature examples are described in section 5.1.

In order to automatically recognize building features and to derive a multi-scale knowledge graph from a set of planar segments, a graph grammar has been developed. The proposed GG is defined for an RTG and applied in a bottom-up/top-down manner. Since grammars are well-known for their expressive power in generating a very large set of configurations from a relatively much smaller set of components (see section 3.2), a number of different building models can be derived from a rather small number of predefined building features. By formalizing the derivation process of a multi-scale knowledge graph, it is also ensured that only realistic building models are derived from the set of segments. Furthermore, since hierarchic and structural composition is one of the key concept behind grammars, production rules do not need to be applied in all possible configurations to detect all occurring building

features. Moreover, from the application side, the multi-scale knowledge graph in combination with the GG offers not only the possibility to receive the geometry of a building but to enhance the final building models with semantic information. The increased demand for semantically structured building models is thereby easily met with the developed feature-driven reconstruction process. As a further result, certain building details with regard to their semantic meaning can be optionally masked in order to decrease the computational complexity of later analyses or due to privacy issues. The details of the automatic multi-scale knowledge graph derivation process including the proposed GG are explained in section 5.2.

5.1 Multi-Scale Knowledge Graph

The multi-scale knowledge graph as introduced in this thesis is used to organize and manage meaningful building information during the reconstruction process. It represents a hierarchical decomposition of a building in a five-level graph structure. Each level of the multi-scale knowledge graph represents a different information level of the building to be reconstructed:

- Level 1: building model
- Level 2: super-primitives
- Level 3: primitives
- Level 4: primitive components
- Level 5: semantically labeled roof topology

Analogous to a scene graph in computer vision that describes a scene consisting of a single building, leaf nodes of the multi-scale knowledge graph represent the most basic building parts, interior nodes a grouping of lower-level building parts, and the root the building as a whole. Nodes of the same information level are “horizontally” connected in the multi-scale knowledge graph if their corresponding building parts are in spatial proximity to each other. “Vertical” edges, in turn, reflect the decomposition of a higher-level building part into its components or, conversely, the aggregation of lower-level objects to a higher-level object. Note, a node in the multi-scale knowledge graph can be more than one time vertically connected to a higher-level node if it belongs to several building features. Since the collection of building features is semantically interconnected, the occurrence of building features can be defined both by the segments and the context of the connected features. Thereby, only semantically compatible features are connected to higher-level building features, which prevents the misuse of features. The multi-scale knowledge graph of an example L-shaped building is shown in Figure 5.1. The details of each level, starting with the lowest level, are described in the following subsections. Generally, the set of node and edge labels can be adapted according to the building shapes to be reconstructed without changing the structure of the multi-scale knowledge graph. Thus, the multi-scale knowledge graph can be seen as a general model for the representation of building knowledge.

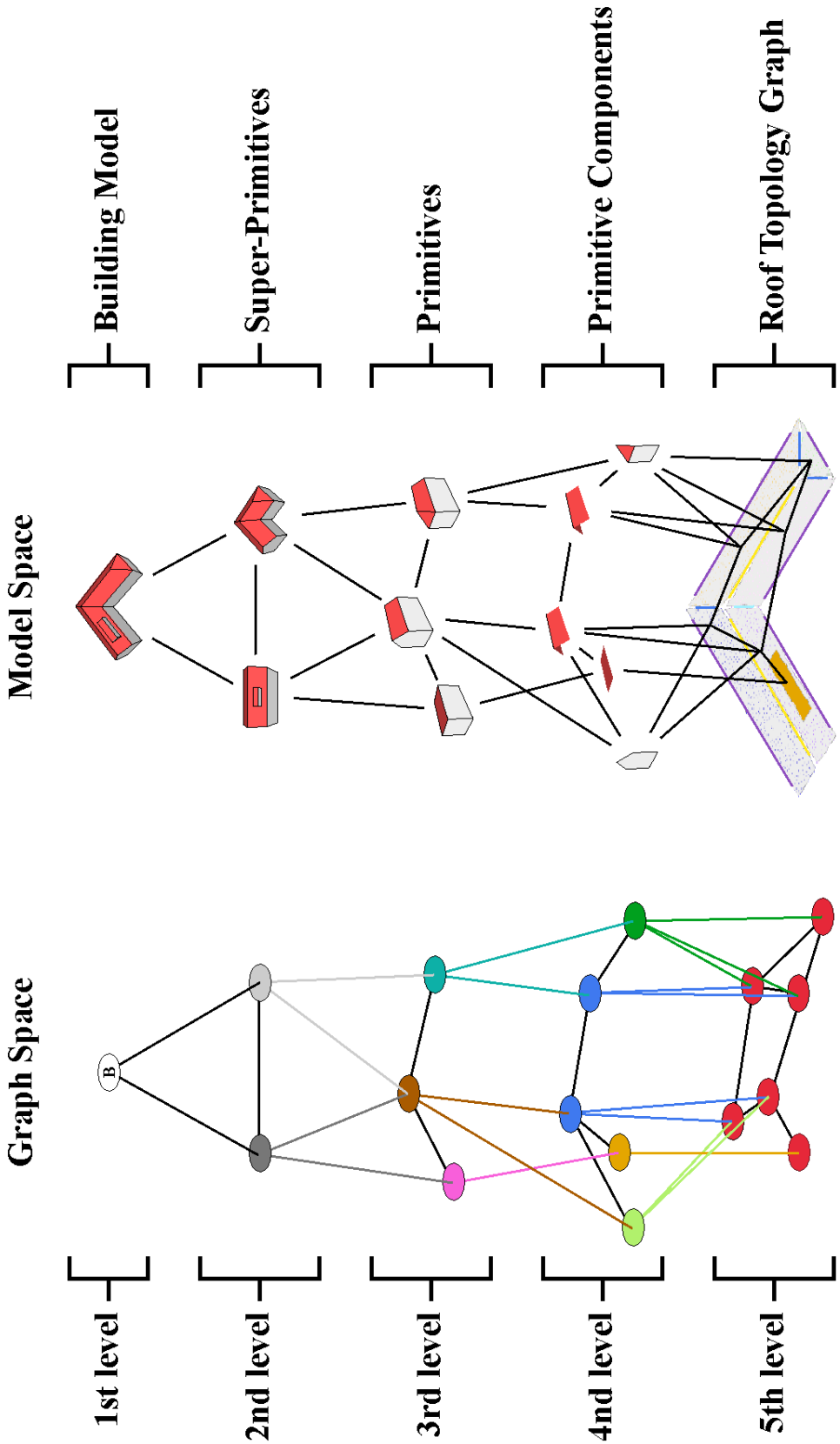


Figure 5.1. The multi-scale knowledge graph for an example building.

5.1.1 Roof Topology Graph

The lowest level of the multi-scale knowledge graph consists of an RTG which is based on the obtained segments of the roof plane segmentation process. In roof topology graphs, segments are represented as nodes and two nodes are connected by an edge if their segments are adjacent to each other. In order to distinguish between different types of nodes and edges, they are labeled according to different criteria. In the last couple of years, RTGs as initially presented in (Verma et al., 2006) became popular for the automatic reconstruction of buildings. Since then, several extensions and more recently automatic correction procedures have been proposed (Oude Elberink and Vosselman, 2009; Perera and Maas, 2014; Xiong et al., 2014b; Xiong et al., 2015; Xu et al., 2015; Jarząbek-Rychard and Borkowski, 2016). One reason for their popularity is that RTG based reconstruction methods usually keep the robustness of traditional model-driven reconstruction approaches while reducing the search effort and the computational time. This can be realized because the search for predefined models is no longer performed directly on the input data but on higher-level information in the so-called topology space.

The presented RTG in this thesis is especially designed to deal with segments gained from sub-surface growing. This is necessary because these segments are enlarged with virtual points so that they may grow below other segments and have thus different properties than segments from common surface growing methods. The proposed RTG defines, therefore, the existence of an edge different from other common RTGs and consists of an extended set of node and edge attributes. The node and edge attribute values are determined according to geometrical, topological and locational properties of their related segments. This includes, inter alia, the information about each edge if the related segments of its incident nodes feature only a two-dimensional adjacency or a three-dimensional adjacency as well, if their intersection results in a horizontal or a sloped intersection line, and if their normal vectors feature in the x-y direction a specific configuration such as parallel or orthogonal. Furthermore, the developed RTG deals with directed edges so that also one-sided relationships between nodes can be directly represented in the graph structure (e.g., the enclosure of a segment by another segment). The attributes of the nodes and edges are described in more detail in subsection 5.2.1.

In addition to geometrical, topological and locational features, nodes and edges are semantically labeled as well. Some examples of semantic features in the model space are illustrated in Figure 5.2. In contrast to the attributes, semantic features are not determined by functions but they are formulated as part of the proposed GG presented in subsection 5.2.2 and 5.2.3.

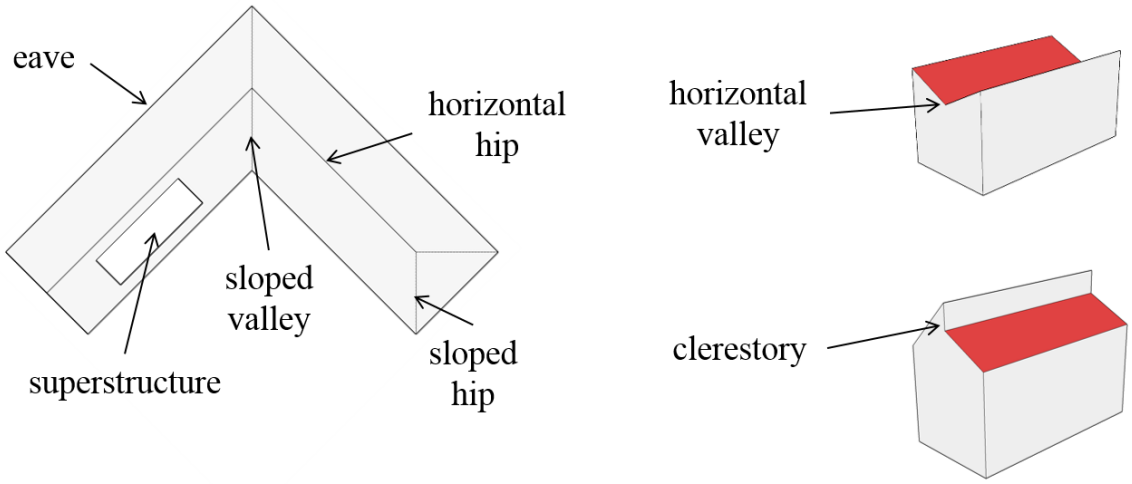


Figure 5.2. Examples of semantic labels in the model space.

5.1.2 Primitive Components

On the next higher level of the multi-scale knowledge graph, nodes represent primitive components. A primitive component belongs either to the group of basic components or to the group of ending components.

There are basically three different types of basic components. Each of them reflects in the x-y plane one of the central geometry in the basic component: plane, line, and point. In Figure 5.3, three different examples of basic components are illustrated as part of primitive instances.

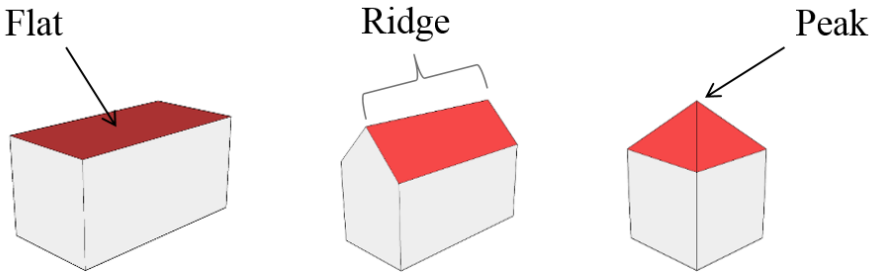


Figure 5.3. Examples of basic components as part of primitive instances.

Ending components, on the other side, represent possible endings of a primitive. Some examples of possible ending components are shown in Figure 5.4.

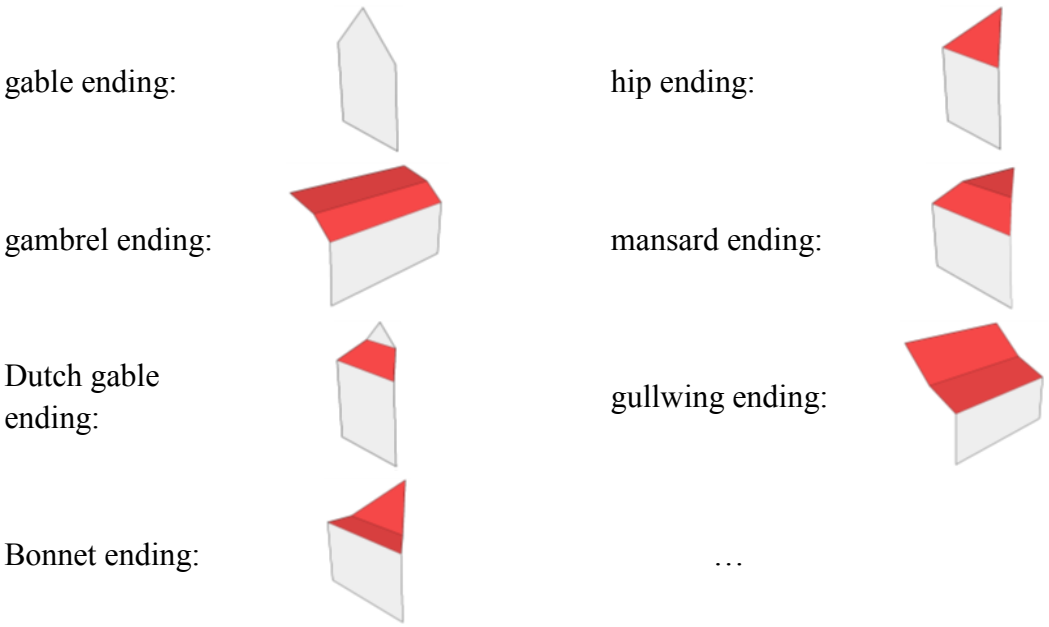


Figure 5.4. Examples of ending components.

Note, both primitive and ending components cannot exist alone but must be composed to higher-level objects. These higher-level objects are called primitives and are represented in the next higher level of the multi-scale knowledge graph.

5.1.3 Primitives

In the third level of the multi-scale knowledge graph, nodes represent frequently occurring building primitives. In contrast to the elements of the previous levels, primitives represent valid solids so that they can be converted into watertight B-rep models. Here, each building primitive consists of one basic element and four further elements that define the endings of the basic element. There are three different types of basic elements that each reflects one of the following central geometry in the basic element: point, line, and plane. By selecting four ending elements, specializations of these basic elements are derived. This approach offers the following advantages:

- Some ending elements can be used for different basic elements so that frequently occurring ending elements do not need to be individually defined for each specialization.
- The combination of different ending elements to derive a specialization from a basic element enables a very compact description of a large number of building primitives that are most common in reality.
- The types of primitives that can be represented in the multi-scale knowledge graph is semantically restricted so that unnatural primitives are avoided in the resulting building models.

Some examples of primitives in the third level of the multi-scale knowledge graph are shown in Figure 5.5. Here in the first row, the central geometry of each basic element is a plane. By

selecting different sets of ending types, different roof primitives are derived. As for example shown, by only changing the two opposing ending elements of a gambrel roof that each represent a gable ending, a mansard roof can be derived. Similar statements can be made about the primitives d) to l), whose central element reflects a line geometry. Note, as for example shown in the butterfly roof in l), the central geometry of a basic element does not necessarily have to be the highest geometric feature of a primitive. A representative of a point reflecting central geometry is finally shown in m).

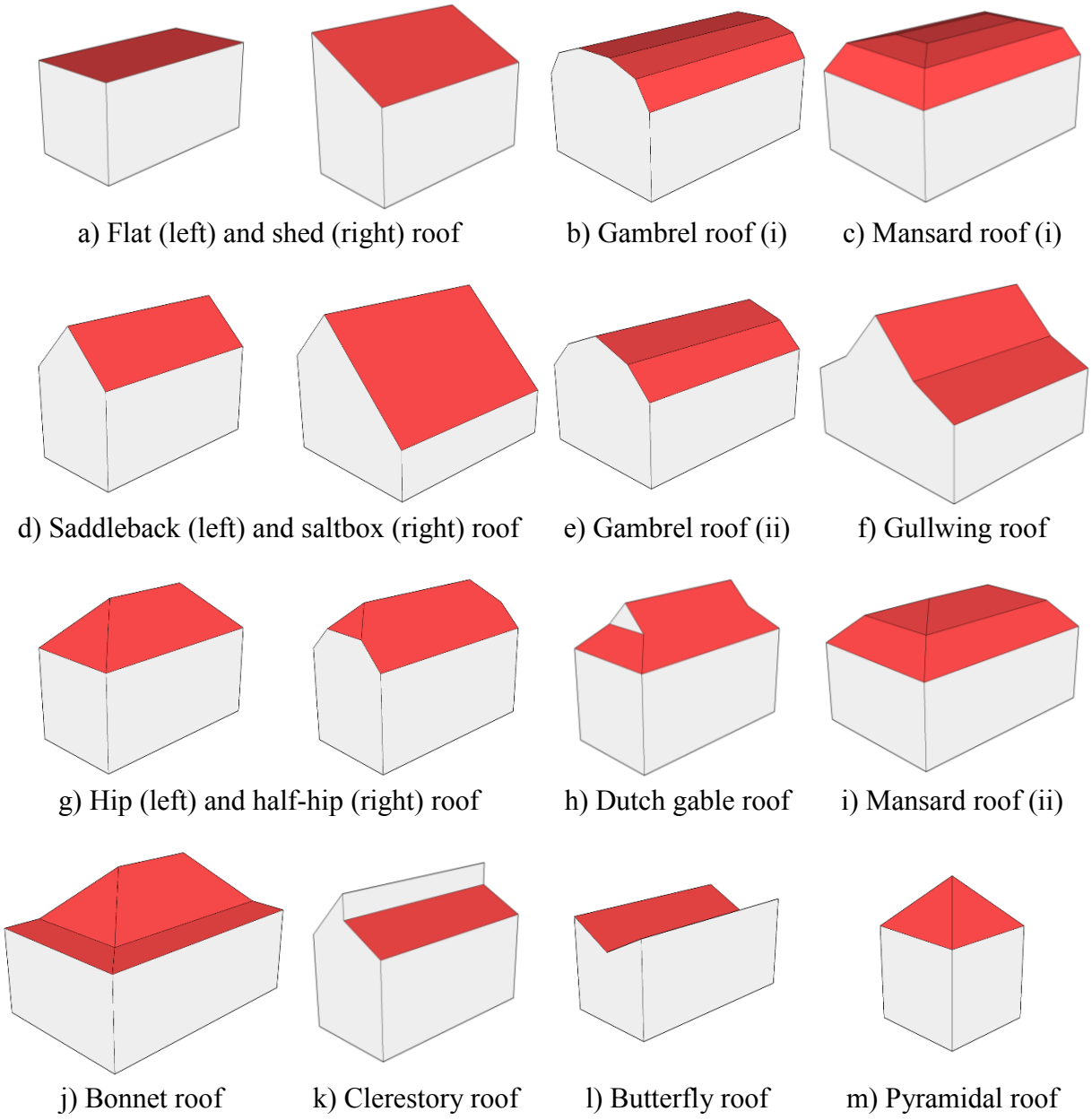


Figure 5.5. Some primitive examples that are of interest in the third level of the multi-scale knowledge graph.

Two nodes are connected in the third level of the multi-scale knowledge graph if their primitive components are adjacent to each other. A distinction is made between neighboring adjacencies and enclosing adjacencies, which means in the latter case that one primitive is

completely located on the other primitive (i.e., located within the other primitive in the x-y plane). The latter is, for example, particularly the case for roof superstructures.

5.1.4 Super-Primitives

In the next higher level of the multi-scale knowledge graph, nodes represent links between adjacent primitives of the third level to model more complex roof shapes. There are three different types of nodes to distinguish between the three most common types of links: T-shaped, L-shaped, and X-shaped links. For each type of link, different cases can be distinguished based on the height difference of the linked primitives. These cases are presented in the following on the basis of saddleback primitives but the differentiation is valid for other primitive types as well.

As illustrated in Figure 5.6 for two linked saddleback primitives that form a T-shape, three different cases can be differentiated depending on the height of the intersecting ridge compared to the height of the other ridge. In all of them, one of the ridges intersects a segment associated with the other ridge so that the segments form two sloped valleys. Due to the subsequent solid modeling technique described in chapter 6, all cases can be modeled during the construction step in the same way so that a distinction between them becomes obsolete in the multi-scale knowledge graph.

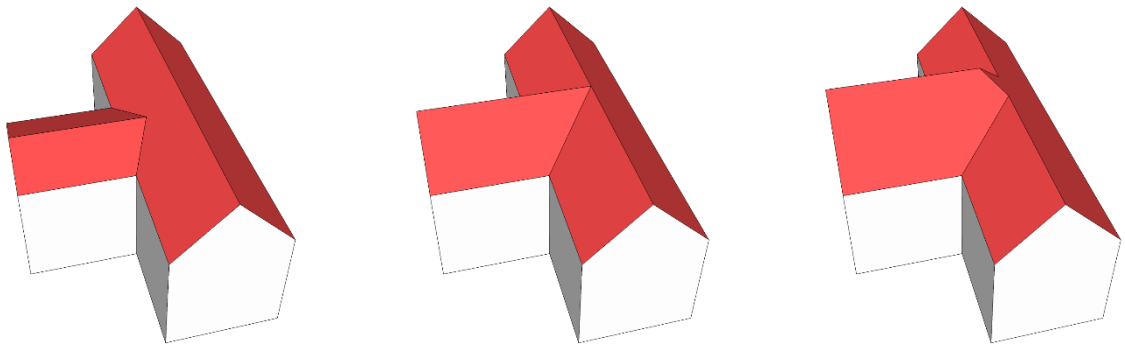


Figure 5.6. Three distinct cases of a T-shaped ridge intersection.

In an L-shaped link, there are essentially two cases that can be distinguished. Both cases are illustrated in Figure 5.7. In the first case, the ridges feature the same height so that they intersect each other, whereas this is not the case in the second one. Analogous to T-shaped links, both cases are constructed in later stages in the same way so that a distinction between both cases in the multi-scale knowledge graph becomes obsolete as well.

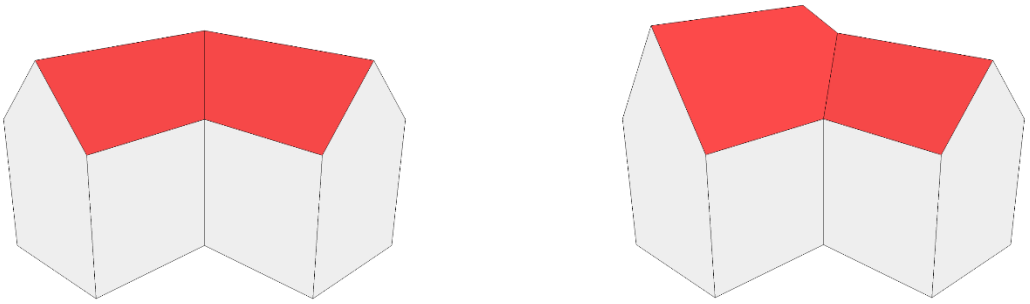


Figure 5.7. Two distinct cases of an L-shaped ridge intersection.

The same applies for the four distinct cases of an X-shaped link that are exemplarily shown in Figure 5.8. Note, generally it is not necessary to explicitly represent X-shaped links because they can be treated as multiple L-shaped links. However, their explicit presence in the multi-scale knowledge graph enables in later stages to further improve the regularities where sub-surface points are no longer sufficient for this task.

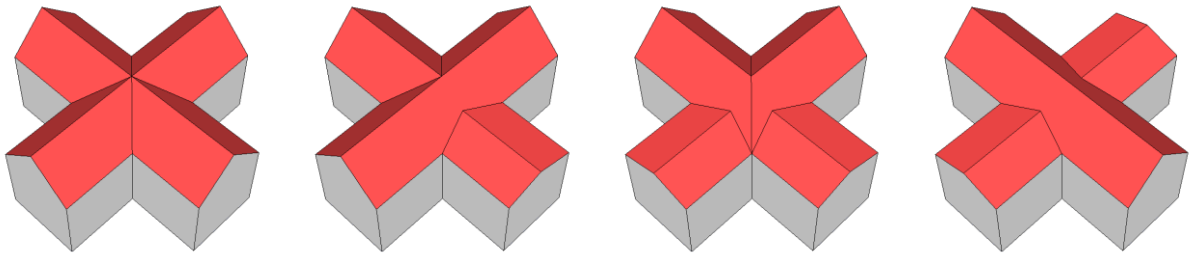


Figure 5.8. Four distinct cases of an X-shaped ridge intersection.

Two super-primitive nodes of the multi-scale knowledge graph are horizontally connected by an edge if they share a common primitive.

5.1.5 Building Model

The top level of the multi-scale knowledge graph consists of a single node that represents the whole building and completes the multi-scale knowledge graph. It is vertically connected to all super-primitive nodes that compose the building.

5.2 Multi-Scale Knowledge Graph Derivation

In this section, the derivation of the multi-scale knowledge graph based on a set of planar segments is described. For each building, a multi-scale knowledge graph is not pre-determined but iteratively constructed in a dynamic way. The whole generation process consists of two main phases: initialization phase and bottom-up/top-down phase. An overview of the whole derivation process is depicted in Figure 5.9.

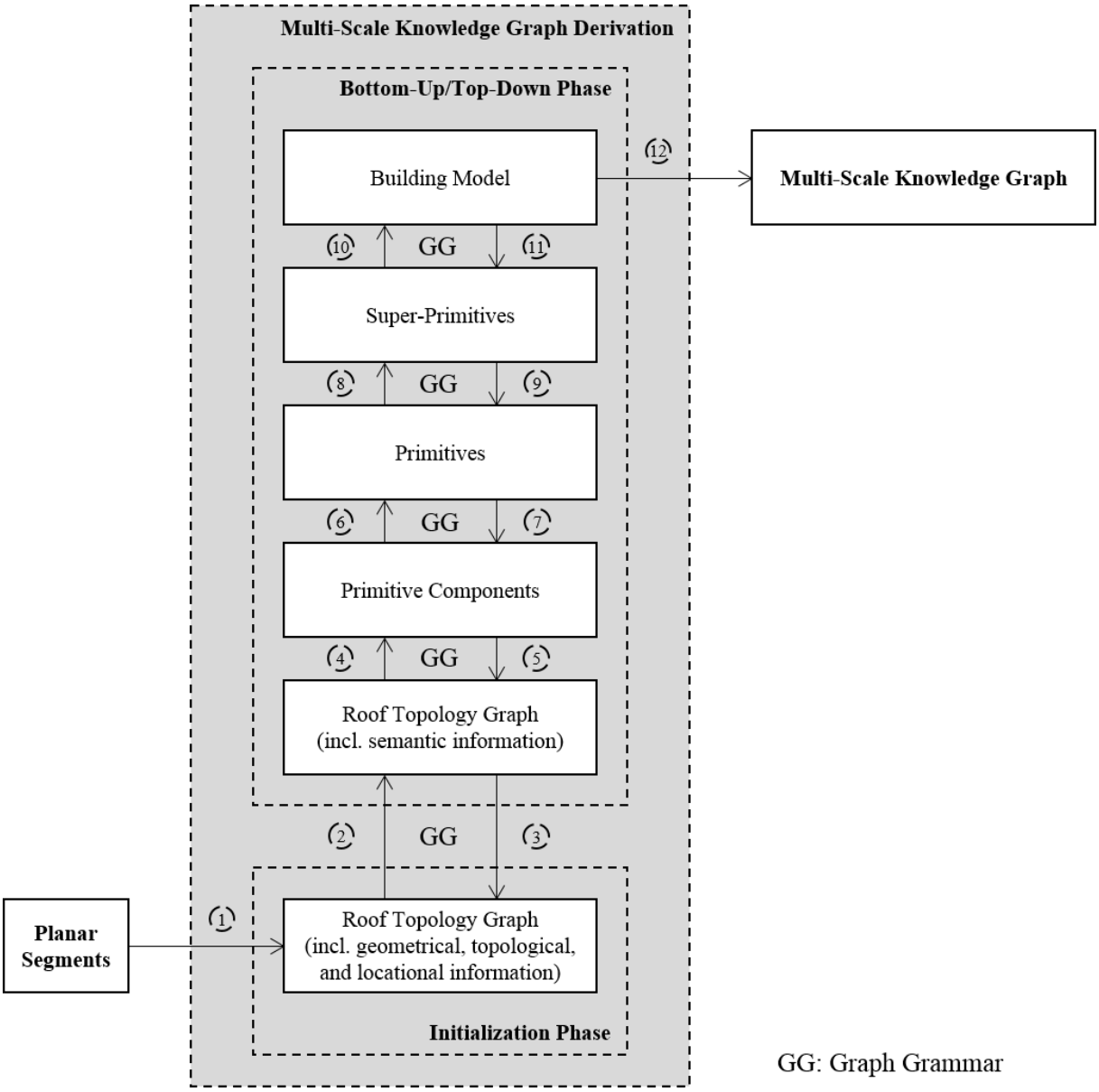


Figure 5.9. Workflow of the building knowledge derivation process.

In the initialization phase, an RTG is determined from a given set of planar segments. Since segments are here obtained from a sub-surface segmentation process, node and edge attributes of the RTG need to be specifically defined. Otherwise, the RTG would become ambiguous in cases in which, for example, a distinction between real surface and virtual sub-surface points is not made. The construction details of the RTG and its attributes are described in subsection 5.2.1.

Once all adjacencies between segments are determined and classified in terms of their geometrical, topological and locational properties, production rules of a GG are applied during the subsequent bottom-up/top-down phase to recognize higher-level building features (bottom-up phases) and to detect missing lower-level building features (top-down phases). For this, an iterative rule-based procedure is presented that selects and matches production rules via subgraph matching so that the initial RTG is finally transformed into the multi-scale

knowledge graph. The set of production rules defines the set of applicable graph transformations and each production rule represents the derivation of a particular building feature that supports the modeling process of a building. In order to deal with different point densities and to provide an appropriate degree of flexibility, each production rule is parameterized so that thresholds can be easily adapted to the properties of the input data. Note, however, that the developed rules rely more on the size of features than on the point cloud density itself; as long as the point cloud is dense enough to actually represent the building feature.

In general, the production rules can be defined for and applied to segments of any segmentation method. However, to perform the building feature recognition procedure on segments gained from sub-surface growing entails several advantages because in contrast to conventional surface growing, the segmentation process continues below other surfaces. As a result, segments have the property to be much more intertwined compared to segments from other segmentation methods. Consequently, strict thresholds can be applied during the building knowledge derivation process. Additionally, the enlargement of segments helps to better recognize adjacencies, intersections, and also the sub-shapes of building roofs. Moreover, the use of sub-surface growing segments leads to a reduced number of rules because some problems, as described in subsection 4.3, are already implicitly solved by sub-surface growing and do not have to be regarded as special cases in additional production rules. Consequently, also intricate rules can be applied to recognize more complex building features by keeping the number of rules small. Furthermore, small segments from roof superstructures can be omitted in the case that the reconstruction method is unable to form well-shaped roof parts thereof. The sub-surface points ensure that the remaining segments still cover the area of the missing segments in a coherent way.

As depicted in the overview of the whole multi-scale knowledge graph derivation process in Figure 5.9, the GG is alternately applied in bottom-up and top-down phases to derive the different levels of the multi-scale knowledge graph. In the bottom-up phases, building features are determined by aggregating subgraphs of lower-levels to nodes of higher-levels. If the multi-scale knowledge graph construction is only based on the bottom-up phases, it might be incomplete due to unrecognized lower-level features, which is usually caused by (partial) occlusions, too strict thresholds or a combination thereof. Therefore, further building features are detected in the top-down phases using top-down methods. For this, production rules are used that decompose higher-level nodes into a set of lower-level nodes so that missing lower-level nodes and edges can be specifically searched. The conjunction of both different kind of phases enables the recognition of important building features. The details of the bottom-up and top-down phases are described in subsection 5.2.2 and 5.2.3, respectively.

In some cases, a segment can be part of more than one building feature of the same type that is going to be derived. In order to ensure that all instances are adequately represented in the multi-scale knowledge graph, a segment splitting procedure has been developed. During this procedure, each segment that is part of a matched subgraph is split into patches until each segment patch belongs to at most one new derived building feature instance. As a result, the segment splitting process internally changes the shape and size of segments, making them

more applicable to other feature recognition rules. Note, segment splitting is not a specific problem caused by sub-surface segmentation but it is needed whenever a roof segment is part of many features. The details of the segment splitting process are explained in subsection 5.2.4.

5.2.1 Initialization Phase

In the initialization phase, the obtained segments from the roof plane segmentation process are organized in a roof topology graph structure. The proposed RTG is defined as $G = (V, E, \mu_V, \mu_E, s, t)$ where V is the finite set of attributed vertices (nodes), $E \subseteq \{(v_i, v_j) \mid v_i, v_j \in V, v_i \neq v_j\}$ the finite set of attributed edges, μ_V the finite set of vertex attributes, μ_E the finite set of edge attributes, and the two unary operations s and t with $s : e \to v_1$ and $t : e \to v_2$ for a given $e \in E$. Segments are represented in the directed RTG as attributed nodes so that there is a bijection $f: S \to V$, where S is the set of obtained segments. The relationship between two adjacent segments is represented as two directed attributed edges connecting the nodes with each other. In the following, node and edge features of the RTG are described. For this, let $v_1 = s(e)$, $v_2 = t(e)$, $v.points$ the set of points that belongs to the segment $f^{-1}(v)$, $v.plane$ the plane that is defined by the segment $f^{-1}(v)$, and $v.normal$ the normal vector of $v.plane$.

$O_z(v)$: Each node of the RTG has an attribute that describes the z-orientation of its segment in space. For this, four different classes are defined: horizontal, vertical, sloped, and unknown. The latter is assigned to those nodes whose segments feature an insufficient number of points to calculate a robust normal vector. Otherwise, a vertex is essentially classified based on the z-value of the normal direction of the plane that can be directly derived from the points of its segment:

$$O_z(v) = \begin{cases} vertical, & if\ |v.points| > t_1\ \wedge \\ & PH(v) < t_2 \wedge \\ & v.plane.c < t_3 \\ horizontal, & if\ |v.points| > t_1\ \wedge \\ & PH(v) < t_2 \wedge \\ & v.plane.c > 1 - t_3 \wedge \\ & v.points.height < t_4 \\ sloped, & if\ |v.points| > t_1\ \wedge \\ & PH(v) < t_2 \wedge \\ & t_3 \leq v.plane.c \leq 1 - t_3 \\ unknown, & otherwise \end{cases} \tag{5.1}$$

where

$v.plane.c$: slope in z-direction of the plane $v.plane$,

$v.points.height$: difference between the maximum z-value and the minimum z-value of all points that belong to $v.points$,

$PH(v)$: MSE of $v.points$ to $v.plane$ which reflects the plane homogeneity,

$t_1 \in \mathbb{N}^+$, $t_2 \in \mathbb{R}_0^+$, $0 \leq t_3 \leq 1$, and $t_4 \in \mathbb{R}_0^+$.

At this point, segments with a small number of points are not discarded but classified as unknown so that they can be still taken into account in higher levels of the multi-scale knowledge graph. The presence of their surfaces in the final building model has, however, a low priority. To differentiate between horizontal and sloped segments, it is important to have in addition to the normal vector criterion also a height criterion. The latter is especially needed for large segments because already small gradients may lead to non-horizontal surfaces.

Two nodes are connected in the first level of the multi-scale knowledge graph if their segments are adjacent to each other. In contrast to commonly used RTGs, the existence of an edge is independent from the length of a hypothesized intersection line segment. Instead, two segments are adjacent to each other if at least one point exists in each segment so that the distance between them is less than a predefined threshold:

$$E = \{(v_i, v_j) \mid \exists p_i \in v_i.points, \exists p_j \in v_j.points, \\ v_i \in V, v_j \in V, v_i \neq v_j : |\overrightarrow{p_i p_j}| < t\} \tag{5.2}$$

where

$t \in \mathbb{R}^+$.

In order to distinguish between different types of adjacencies, each edge has a set of attributes that characterizes the relationship of its incident nodes. There are in total seven edge attributes that are described in the following.

$AD(e)$: In order to distinguish between potential step and intersection lines at a later stage, a distinction is made between two- and three-dimensional adjacency, ignoring in Equation (5.2) the z-coordinates of the points in the two-dimensional distance calculation.

$Presence(e)$: Due to the proposed sub-surface growing procedure presented in chapter 4, a further distinction is made between adjacencies that require sub-surface points in Equation (5.2) and those adjacencies that are connected through their surface points. The differentiation is necessary because roof surfaces of the former kind of adjacencies are actually not adjacent in reality but only virtual below the roof surface. Without this distinction, virtual adjacencies could misguide the building knowledge derivation process.

$CV(e)$: To indicate the reliability of an adjacency relation, each edge is further attributed with a confidence value. For the confidence value calculation of a three-dimensional adjacency, first an intersection line l_{hi} between both segments is hypothesized that results from the intersection of those two planes that can be directly derived from the points of each sub-surface growing segment. Then, the confidence value is determined as follows:

$$CV(e, l_{hi}) = min\left(\begin{array}{cc} & \alpha\left(max\left(0, 1 - a \cdot PH(e)\right)\right) + \\ 1, & \beta\left(min\left(1, b \cdot SP(e, l_{hi})\right)\right) + \\ & \gamma\left(min\left(1, c \cdot OL(e, l_{hi})\right)\right) \end{array}\right) \tag{5.3}$$

where

$0 \leq \alpha, \beta, \gamma \leq 1$, $\alpha + \beta + \gamma = 1$,
$a, b, c \in \mathbb{R}^+$, and $0 < b \leq 1$.

The confidence value is based on three terms whose influence is balanced by the weight factors α, β, and γ and which can be adapted to different input data characteristics by a, b, and c. In the first term, the average distance of the segment points to their related segment plane is taken into account to determine the plane homogeneity:

$$PH(e) = \frac{1}{2}\left(PH(v_1) + PH(v_2)\right) \tag{5.4}$$

where

$PH(v)$: MSE of $v.points$ to $v.plane$ which reflects the plane homogeneity.

The second term considers the number of supporting points, which are close to the hypothesized intersection line:

$$SP(e, l_{hi}) = |A| + |B| \tag{5.5}$$

where

$A = \{p \in v_1.points \mid distance(p, l_{hi}) < t\}$,
$B = \{p \in v_2.points \mid distance(p, l_{hi}) < t\}$,
$distance(p, l)$: distance between a given point p and a given line l, and
$t \in \mathbb{R}^+$.

In the third term, the overlap length of both segments with regard to the hypothesized intersection line is determined:

$$OL(e, l_{hi}) = overlapLength(f^{-1}(v_1), f^{-1}(v_2), l_{hi}). \tag{5.6}$$

The calculation of the overlap length is illustrated in Figure 5.10. First, all points of the two segments that are within a maximum distance to the hypothesized intersection line are projected onto the hypothesized intersection line (black). Then, for each segment all line segments (blue and green) are determined that are supported by a continuous sequence of points on the intersection line that initially belong to the segment. The intersection length of the set of blue and green line segments yields the overlap length of the segments (red). The overlap length of two adjacent segments is thus the summed length of all line segments that are supported by a continuous point sequence of both segments.

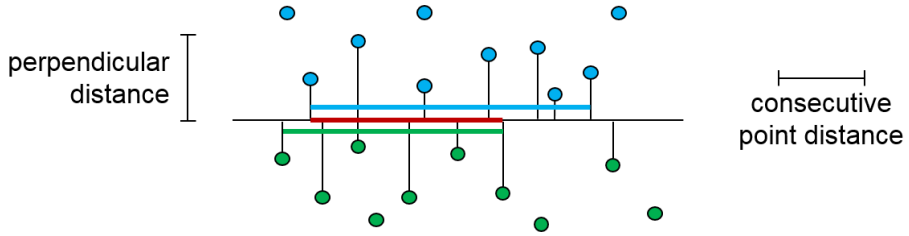


Figure 5.10. Overlap length of two adjacent segments represented by the red line segment.

The confidence values for two-dimensional adjacencies, which feature potential step lines, are estimated for sloped segments in a similar way. First, a hypothesized line is determined between two adjacent segments. For this, the intersection line is initially calculated that results from the intersection of those two planes that can be directly derived from the points of each sub-surface segment. Due to the height gap of these segments, the intersection line is then horizontally translated along its perpendicular x-y direction so that it best separates the surface points of the segments from each other. To determine this specific location, the principles of a support vector machine are adapted in a way that the x-y direction of the hyperplane normal is set to the normal of the initially calculated intersection line. To consider that the points of the sets are not always linearly separable, slack variables as used in a soft-margin SVM are incorporated. For the details of SVM see, for example, (Vapnik, 2000). Then, the confidence value is calculated in the x-y plane, ignoring the z-coordinates, based on the above mentioned three terms and assigned to the edge that represents the two-dimensional adjacency.

$O_{intersection}(e)$ and $Visibility(e)$: In addition to two- and three-dimensional adjacency information, common RTGs utilized in building reconstruction approaches usually incorporate the relative x-y orientation of two adjacent segments. Here, a distinction is often made, on the one hand, whether the segment faces form a convex (+) or a concave angle (-) and, on the other hand, between edges whose segments feature horizontal intersections (H) and edges whose segments feature sloped intersections (S):

$$O_{intersection}(e) = \begin{cases} H, & if \ -t < intersection(v_1.plane, v_2.plane).normal.z < t \\ S, & otherwise \end{cases} \quad (5.7)$$

where

$intersection(p_1, p_2).normal.z$: z-value of the normal vector of the intersection line that results from the intersection of the two given planes p_1 and p_2, and

$0 \leq t \leq 1$.

As shown on the left side of Figure 5.11, the corresponding RTG of a T-shaped building gained from a common segmentation method includes three H+ and two S- classified edges. However, the expressive power of these attribute values is limited and especially for segments gained from sub-surface growing not always sufficient. An example for this is given for the same building shape on the right side of Figure 5.11. Here, the red and the blue segment are,

due to sub-surface points, adjacent to each other. Their common edge can be classified with S+ if the left part of the blue segment is ignored. This however results in loss of information so that a distinction to other S+ adjacencies is not discernible anymore. An analogous problem occurs for the edge between the yellow and the blue segment. Moreover, the green and the orange segment are merged to one segment in sub-surface growing so that the relationship between the red and the green segment becomes S+ and S- at the same time. Similar problems still occur in RTGs that do not differentiate between convex and concave adjacencies but between adjacencies whose segment normals projected on the x-y plane point either away or towards each other.

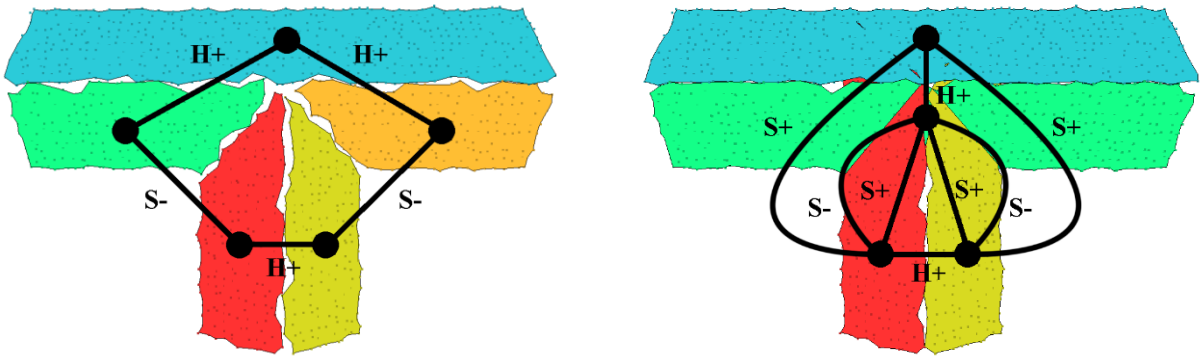


Figure 5.11. Corresponding topology graph of a T-shaped building based on segments gained from surface (left) and sub-surface (right) segmentation.

To avoid these problems, a visibility attribution concept, which indicates the relative surface point positions of one segment to another adjacent segment, is introduced in addition to the slope characterization of the intersection. Since edges are implemented as directed edges, the visibility of a segment to another adjacent segment is classified as either visible (+), partially visible (○), or not visible (-):

$$Visibility(e) = \begin{cases} +, & if \ \frac{1}{|A|}\sum_{p \in A} \mathbb{1}_{\{p \in A \mid p \cdot v_1.normal + v_1.plane.d > 0\}}(p) > t_1 \\ -, & if \ \frac{1}{|A|}\sum_{p \in A} \mathbb{1}_{\{p \in A \mid p \cdot v_1.normal + v_1.plane.d < 0\}}(p) > t_1 \\ \circ, & otherwise \end{cases} \tag{5.8}$$

where

$A = \{p \in v_2.points : distance(p, plane) > t_2\}$,
$distance(p, plane)$: distance of a given point p to a given plane $plane$,
$v_1.plane.d$: distance of the plane $v_1.plane$ to the origin (with a negative sign if the origin is in front of the plane),
$0 < t_1 \leq 1$, and $t_2 \in \mathbb{R}_0^+$.

Thereby, an edge $e : v_1 \rightarrow v_2$ is classified as visible if all surface points that belong to the segment $f^{-1}(v_2)$ are in front of the plane $v_1.plane$. If all surface points of $f^{-1}(v_2)$ are

behind this plane, then the edge is classified as not visible. Otherwise it is classified as partially visible because points from $f^{-1}(v_2)$ are both in front and behind the plane $v_1.plane$. To incorporate the uncertainty of the input data points and possible faulty assignments of border surface points during the segmentation process, points that are located in a close distance to $v_1.plane$ are in all of these three cases not considered. With respect to a minimum number of points, edges are classified according to the three visibility classes. In combination with $O_{intersection}(e)$, the corresponding edges of the T-shaped building from the previous example are obtained as shown in Figure 5.12.

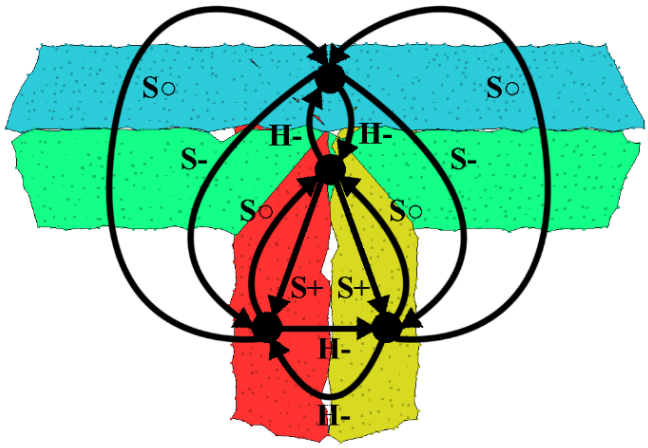


Figure 5.12. Corresponding topology graph of a T-shaped building attributed according to the intersection and visibility attribution concept.

$O_{xy}(e)$: For the direct recognition of some regularized structures, which are often present in a building, a further attribute is assigned to each edge that meets the condition $O_z(v_1) = O_z(v_2) = "sloped"$. It mainly represents parallel and orthogonal structures of sloped roof surfaces in the x-y directions:

$$O_{xy}(e) = \begin{cases} \rightrightarrows, & if\ Angle_{xy}(v_1.normal, v_2.normal) < t \\ \perp, & if\ 90° - t < Angle_{xy}(v_1.normal, v_2.normal) < 90° + t \\ \leftrightarrows, & if\ Angle_{xy}(v_1.normal, v_2.normal) > 180° - t \\ \neg(\parallel \vee \perp), & otherwise \end{cases} \tag{5.9}$$

where

$Angle_{xy}(n_1, n_2)$: angle between the two given vectors n_1 and n_2 in the x-y plane, and $0° \leq t \leq 45°$.

For the calculation of O_{xy}, first the normal vectors of each adjacent segment pair, whose incident nodes are both classified as sloped, are projected onto the x-y plane. Then, the angle formed by the projected normal vectors is analyzed. With respect to a predefined threshold, edges whose angles are close to 0°, 90°, and 180° are classified as parallel with same

direction (⇉), orthogonal (⊥), and parallel with opposite direction (⇆), respectively. All other edges that connect sloped segments are classified as neither parallel nor orthogonal (¬(∥∨⊥)). In higher levels of the multi-scale knowledge graph, this attribute is frequently considered for the extraction of higher-level information.

$CP\_PC(e)$: Furthermore, it is useful for later stages to represent the information for each segment if its surface points are enclosed in the x-y plane by another segments. The relationship in such a case is called a child-parent relationship (CP). Conversely, if a segment encloses the surface points of another segment, then the relationship is called a parent-child relationship (PC):

$$CP_{PC(e)} = \begin{cases} CP, & if\ f^{-1}(v_1)\ is\ enclosed\ by\ f^{-1}(v_2) \\ PC, & if\ f^{-1}(v_2)\ is\ enclosed\ by\ f^{-1}(v_1) \\ none, & otherwise \end{cases}. \tag{5.10}$$

In order to check if a segment is enclosed by another segment, it is tested with a simple ray casting algorithm based on (Shimrat, 1962) if its surface points are all enclosed by the slightly enlarged boundary of the other segment. Note, a segment might be enclosed by several segments at the same time if the segments result from sub-surface growing. Consequently, to determine the set of segments that encloses the surface points of a segment, all its adjacent segments need to be separately checked with respect to their CP_PC relationships.

5.2.2 Bottom-Up Phases

For the recognition of semantic building features, a set of production rules is defined. Each production rule formulates the derivation of one particular type of building feature and consists of two graphs which represent the LHS and RHS of the production. These production rules are applied during the bottom-up phases to recognize the semantic building features described in section 5.1. For this, production rules are sequentially selected and matched against the so far constructed multi-scale knowledge graph to determine all occurrences of the LHS via subgraph matching and to replace them by the RHS of the production rule if they fulfill the preconditions and postconditions.

In theoretical computer science, the recognition of a graph in another graph is referred to the subgraph isomorphism problem. An isomorphism from a graph $G = (V, E)$ to a graph $G' = (V', E')$ is a bijective mapping $h: V \to V'$ such that $(v_1, v_2) \in E \Leftrightarrow \big(h(v_1), h(v_2)\big) \in E'$. Since the subgraph isomorphism problem is a generalization of both the problem of testing whether a graph contains a Hamiltonian cycle, which is a closed path in the graph that visits each node exactly once, and the maximum clique problem, which finds a complete subgraph with the maximum number of nodes in a given graph, subgraph matching is NP-complete and can therefore probably not be efficiently solved (Garey and Johnson, 1979). In the presented building knowledge derivation process, however, the computational time for subgraph matching is in practice generally acceptable even for complex buildings due to the low number of nodes and edges. Moreover, by considering the

hierarchical structure of the multi-scale knowledge graph and the structural nature of grammars, the set of applied production rules can be divided into five subsets according to the five levels of the multi-scale knowledge graph. Since production rules are defined so that the occurrence of higher-level features depends on the existence of lower-level features, features of a lower-level need to be recognized first before higher-level features can be detected. Thus, production rules of only one of these five subsets need to be considered in each bottom-up phase.

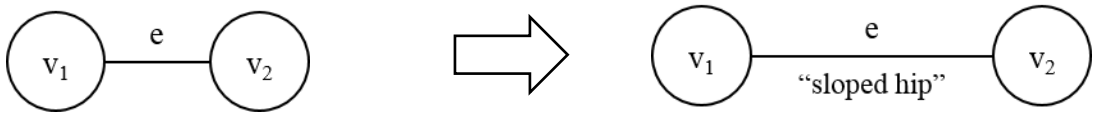
In addition to the subgraph isomorphism constraint, preconditions and postconditions are additionally defined to dictate if a production rule is finally applied to a matched subgraph or not. While preconditions are conditions or predicates that must always be true just prior to the graph transformation of the applied production rule, postconditions must always be true just after the graph transformation. Both preconditions and postconditions are individually defined for each production rule. A precondition, however, that have all production rules in common is that they are only applied to a matched subgraph if they have not been applied to it before.

Since production rules are iteratively applied during bottom-up and top-down phases, a production rule is selected in each iteration and as many matches in the multi-scale knowledge graph are detected via subgraph matching according to the LHS of the selected production rule. If the subgraph matching process stalls, i.e. no more instances of the LHS can be detected in the so far derived multi-scale knowledge graph, and the set of matched subgraphs is not empty, then those matched subgraphs that meet the preconditions and postconditions are transformed to the RHS of the selected production rule and their geometry is adjusted with regard to other features. For example, ridges and eaves become horizontal, the direction of intersecting ridges are rectified with regard to the intersected ridges by favoring orthogonal orientations, eaves are adjusted that they become parallel or orthogonal to each other, etc. The adjustments can be enforced in a straight-forward way due to the semantic information accompanied by the feature rules themselves. The adjustment step is carried out to support the recognition of further features during the next iterations. In particular for point clouds with a low-density, this step is essential to detect a sufficient number of features so that a reasonable building model can be constructed. In the following, some production rules and their preconditions and postconditions are described in more detail.

In the first bottom-up phase, production rules are applied to label nodes and edges of the initial RTG with semantic labels based on their geometrical, topological, and locational attributes. For this, production rules are defined whose preconditions take the node and edge attributes of the initial RTG into account and whose RHS introduce semantic labels. Some of these production rules are exemplarily presented in the following.

Some frequently occurring building features are, for example, the hip and the valley features. Both features represent an intersection of two adjacent roof sides. The only difference between a hip and a valley is that the internal angle of the hip forming roof sides is less than

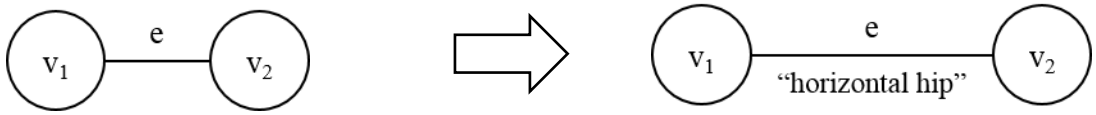
180° so that they form a convex subcomponent while the internal angle of the valley forming roof sides is greater than 180° so that they form a concave subcomponent. Thus, the production rules for hips and valleys are defined in a similar way. According to the slope of their intersection, there are two different variants of hips and valleys: sloped and horizontal hips and valleys. The production rule for the recognition of sloped hip features is, for example, defined as:



Precondition:

$$O_z(v_1) = "sloped" \wedge O_z(v_2) = "sloped" \wedge AD(e) = "3D" \wedge$$
$$Presence(e) = "Surface\ Points" \wedge O_{intersection}(e) = "sloped" \wedge$$
$$Visibility(e) = " - " \wedge CP\_PC(e) = "none" \wedge$$
$$\left(O_{xy}(e) = " \perp " \vee O_{xy}(e) = " \neg(\parallel \vee \perp) "\right) \wedge CV(e) > threshold$$

By changing the value of the visibility attribute from "$-$" to "$+$", sloped valley features are derived. In order to derive horizontal hips and horizontal valleys, the value of $Orientation_{intersection}$ need to be changed to "horizontal" and $Orientation_{xy}$ is not allowed to be "$\perp$" or "$\neg(\parallel \vee \perp)$". Thus, the production rule for the derivation of horizontal hips is defined as:



Precondition:

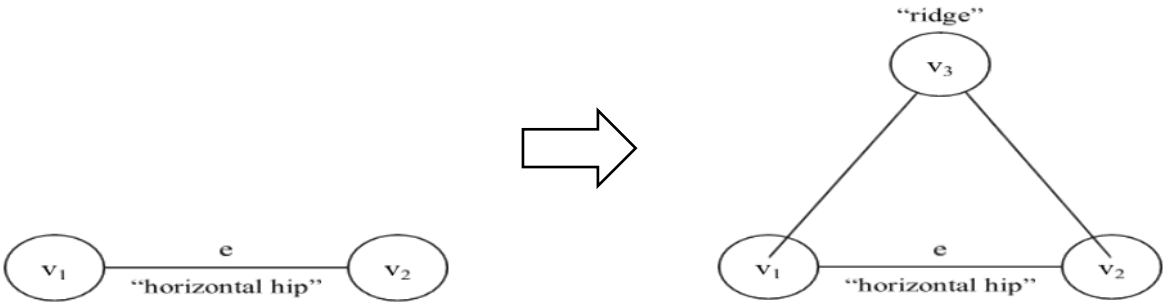
$$O_z(v_1) = "sloped" \wedge O_z(v_2) = "sloped" \wedge AD(e) = "3D" \wedge$$
$$Presence(e) = "Surface\ Points" \wedge O_{intersection}(e) = "horizontal" \wedge$$
$$Visibility(e) = " - " \wedge CP\_PC(e) = "none" \wedge$$
$$\left(O_{xy}(e) = " \leftrightarrows " \vee O_{xy}(e) = " \rightrightarrows "\right) \wedge CV(e) > threshold$$

Further production rules are defined in a similar way for each semantic label of the first level. Note, the presented recognition framework is not limited to a particular set of production rules. Depending on the building characteristics of the input data region, existing production rules can be adapted or discarded and new ones can be added to the building knowledge derivation process. Furthermore, a building feature can be defined in several different ways so that there can be more than one production rule for the recognition of the same building feature.

In contrast to the first bottom-up phase, in which only already existing nodes and edges are semantically labeled, production rules are applied in the remaining bottom-up phases to introduce new nodes in higher levels of the multi-scale knowledge graph. For this, building

features of a lower level are aggregated to a higher-level building feature in the next higher level of the multi-scale knowledge graph. For the aggregations, production rules are defined whose LHS consists of nodes and edges of the same level while the RHS introduces a further node in the next higher level. In order to represent an aggregation between two consecutive levels of the multi-scale knowledge graph, all building features of the lower level are vertically connected to the higher-level building feature of which it is composed. Since nodes of the same level need to be horizontally connected if their associated building features are in spatial proximity, further production rules are defined that horizontally connect higher-level nodes with each other. Some production rule examples of the remaining bottom-up phases are given in the following.

In the second bottom-up phase, for example, nodes of the fifth level are aggregated to a node of the fourth level if they compose a primitive component. As described in subsection 5.1.2, a primitive component is either a basic component or an ending component. One frequently occurring basic component in rooftops is the ridge feature. According to common definitions and to the National Roofing Contractors Association[2] (NRCA), a ridge is the "highest point on a roof, represented by a horizontal line where two roof areas intersect, running the length of the area". The production rule for the recognition of its occurrence is defined as:



Precondition:

$$O_z(v_1) = \text{"}sloped\text{"} \wedge O_z(v_2) = \text{"}sloped\text{"} \wedge O_{xy}(e) = \text{" } \leftrightarrows \text{ "} \wedge CV(e) > threshold$$

Postcondition:

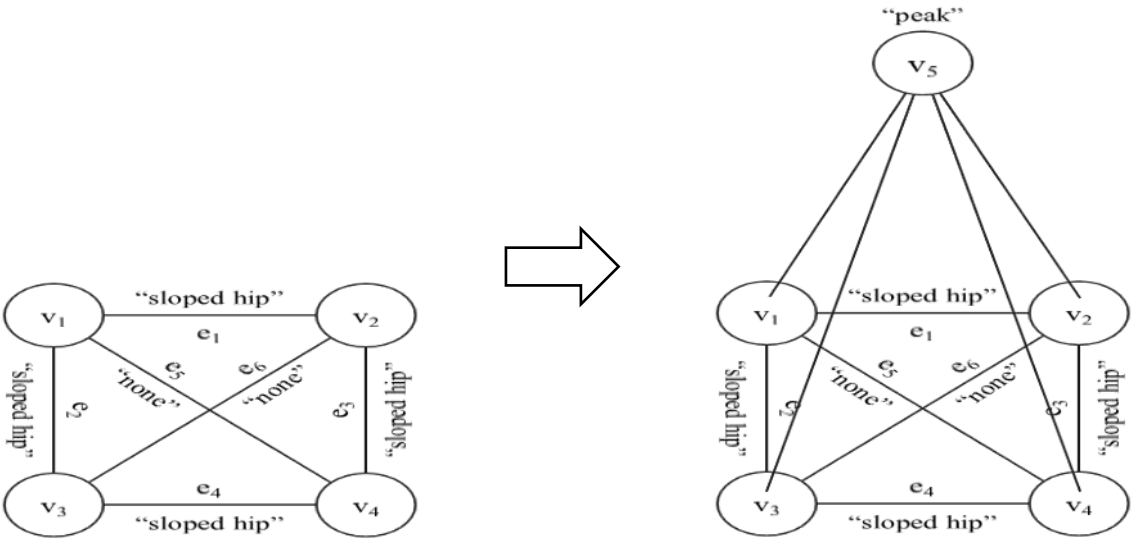
$$OL\big(e, intersection(v_1.plane, v_2.plane)\big) > threshold$$

In order to avoid short ridge line fragments that would result in unnatural details in the final building model, a postcondition is defined for this production rule. The postcondition states that a ridge must have a minimum overlap length (see subsection 5.2.1) and it must be located on the hypothesized intersection line between both segments. In this way, it is ensured that the geometries of all ridge lines in the multi-scale knowledge graph have a minimum length.

Another basic component is the peak. In contrast to the basic components flat and ridge, a peak may originate from different numbers of roof surfaces. Therefore, several production

[2] http://www.nrca.net/

rules are defined to derive different kinds of peak primitives. In cases where the peak is originated from four adjacent roof surfaces, the LHS of the production rule consists of four sloped nodes that are connected with each other. Four of these connections are semantically labeled as sloped hips whereas the remaining two connections do not feature any semantic label. The production rule for the recognition of this kind of peak features is thus defined as:



Precondition:

$$O_z(v_1) = \text{"}sloped\text{"} \wedge O_z(v_2) = \text{"}sloped\text{"} \wedge$$
$$O_z(v_3) = \text{"}sloped\text{"} \wedge O_z(v_4) = \text{"}sloped\text{"} \wedge$$
$$O_{xy}(e_1) = \text{" } \perp \text{ "} \wedge CV(e_1) > threshold \wedge$$
$$O_{xy}(e_2) = \text{" } \perp \text{ "} \wedge CV(e_2) > threshold \wedge$$
$$O_{xy}(e_3) = \text{" } \perp \text{ "} \wedge CV(e_3) > threshold \wedge$$
$$O_{xy}(e_4) = \text{" } \perp \text{ "} \wedge CV(e_4) > threshold \wedge$$
$$CV(e_5) < threshold \wedge CV(e_6) < threshold$$

Postcondition:

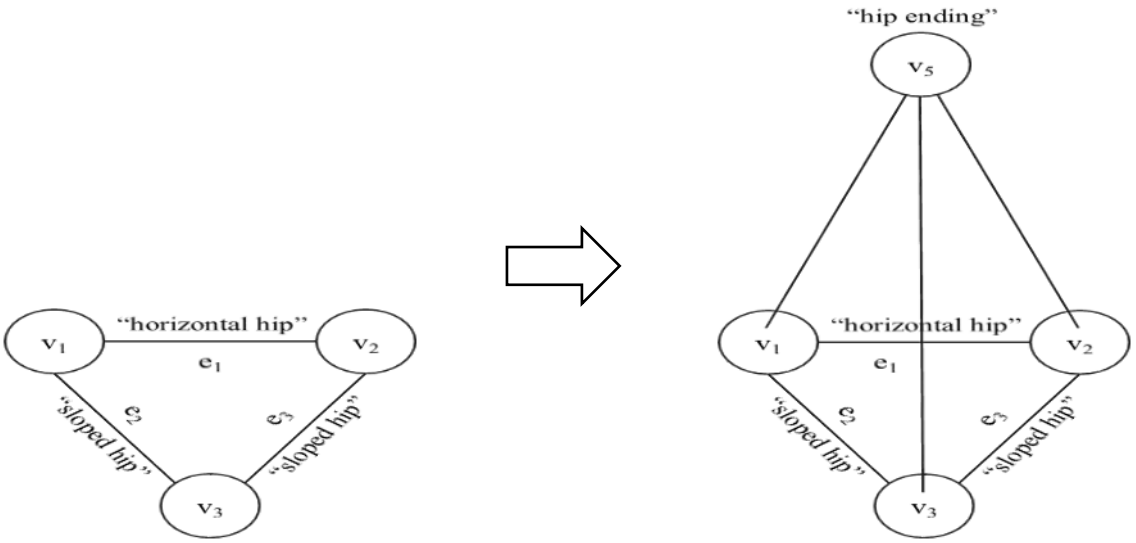
$$distance(peak, v_1.plane) < threshold \wedge$$
$$distance(peak, v_2.plane) < threshold \wedge$$
$$distance(peak, v_3.plane) < threshold \wedge$$
$$distance(peak, v_4.plane) < threshold,$$

where

$peak$: average point that results from the intersection points of the elements in

$$\{U \subset \{v_1.plane, v_2.plane, v_3.plane, v_4.plane\} : |U| = 3\}$$

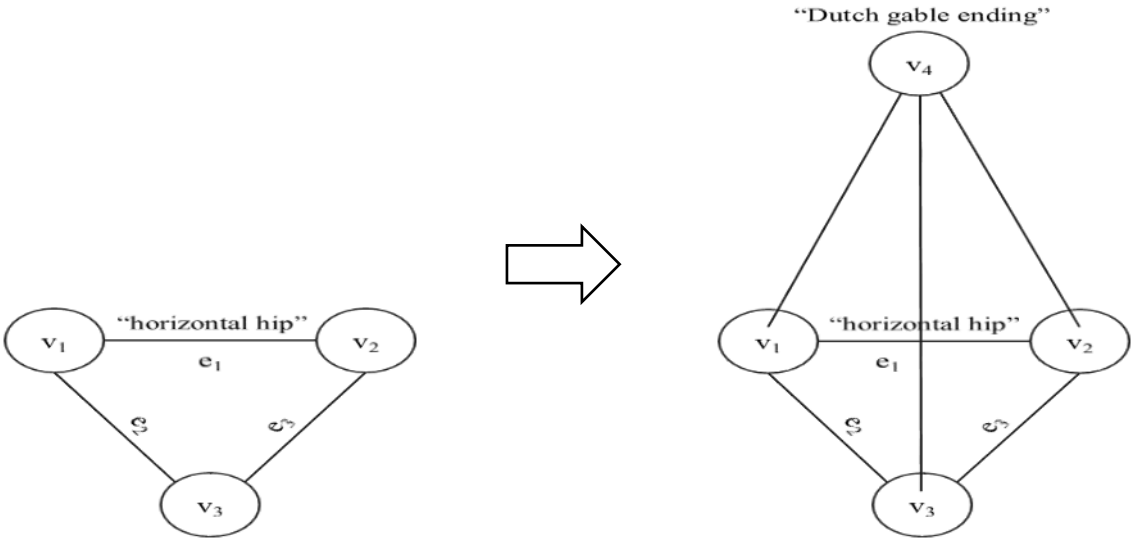
Beside the three basic component types, the nodes in the fourth level may represent different ending components. Note, the LHS of the production rules that derive ending components take only subgraphs of the fifth level into account. Thus, ending components may be introduced even without any main component node in their spatial proximity. One example of an ending component is the hip ending, which can be used in later stages to limit a ridge feature by a sloped roof surface. Due to the selected solid modeling technique described in chapter 6, a distinction between different kinds of hip endings (e.g., to a half-hip end) is not needed because they are all constructed in the same way. The production rule for hip endings is defined as:



Precondition:

$$O_z(v_1) = \text{"}sloped\text{"} \wedge O_z(v_2) = \text{"}sloped\text{"} \wedge O_z(v_3) = \text{"}sloped\text{"} \wedge$$
$$O_{xy}(e_1) = \text{" } \leftrightarrows \text{ "} \wedge CV(e_1) > threshold \wedge$$
$$O_{xy}(e_2) = \text{" } \perp \text{ "} \wedge CV(e_2) > threshold \wedge$$
$$O_{xy}(e_3) = \text{" } \perp \text{ "} \wedge CV(e_3) > threshold$$

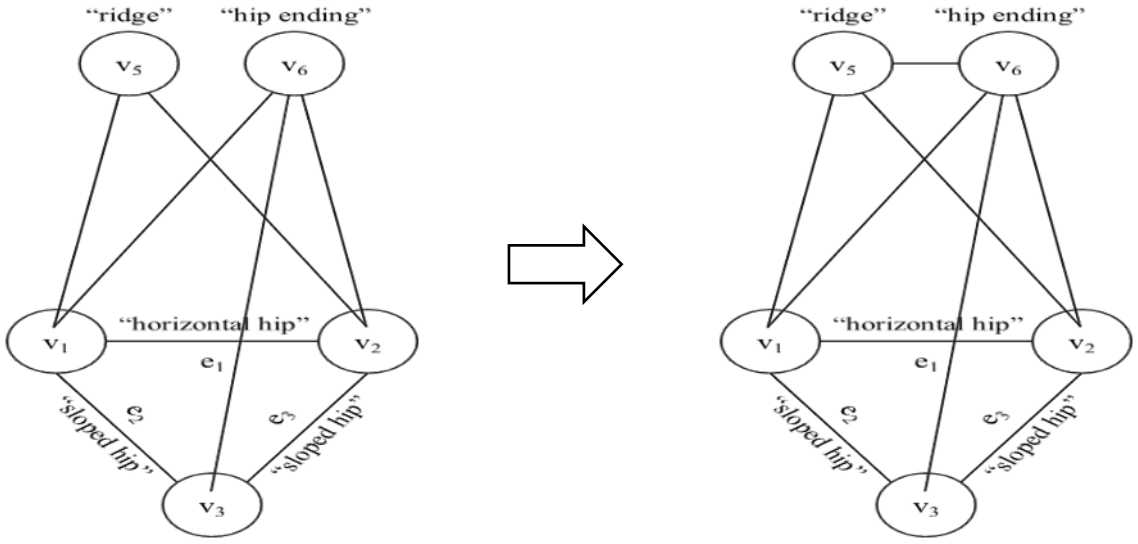
A further frequently appearing ending component is the Dutch gable ending. It combines elements of a hip ending and a gable ending by possessing a small gable at the top of a hip ending. Consequently, to be recognized as a Dutch gable ending, both horizontal hip forming segments have to be, in contrast to a hip ending, partly in front of the hip segment. The corresponding graph consists thus of a segment node that is connected to both segment nodes of a horizontal hip feature. The nodes and edges are labeled as



Precondition:

$$O_z(v_1) = \text{"}sloped\text{"} \wedge O_z(v_2) = \text{"}sloped\text{"} \wedge O_z(v_3) = \text{"}sloped\text{"} \wedge$$
$$O_{xy}(e_1) = \text{" } \leftrightarrows \text{ "} \wedge CV(e_1) > threshold \wedge$$
$$AD(e_2) = \text{"}3D\text{"} \wedge Presence(e_2) = \text{"}Surface\ Points\text{"} \wedge$$
$$O_{intersection}(e_2) = \text{"}sloped\text{"} \wedge Visibility(v_1 \rightarrow v_3) = \text{" } - \text{ "} \wedge$$
$$Visibility(v_3 \rightarrow v_1) = \text{" } \circ \text{ "} \wedge CP\_PC(e_2) = \text{"}none\text{"} \wedge$$
$$O_{xy}(e_2) = \text{" } \perp \text{ "} \wedge CV(e_2) > threshold \wedge$$
$$AD(e_3) = \text{"}3D\text{"} \wedge Presence(e_3) = \text{"}Surface\ Points\text{"} \wedge$$
$$O_{intersection}(e_3) = \text{"}sloped\text{"} \wedge Visibility(v_2 \rightarrow v_3) = \text{" } - \text{ "} \wedge$$
$$Visibility(v_3 \rightarrow v_2) = \text{" } \circ \text{ "} \wedge CP_{PC(e_3)} = \text{none} \wedge$$
$$O_{xy}(e_3) = \text{" } \perp \text{ "} \wedge CV(e_3) > threshold$$

In order to horizontally connect two nodes of the same level of the multi-scale knowledge graph if they are in spatial proximity to each other, further production rules are defined. The LHS of these production rules ensure that their sets of vertically connected lower-level nodes have at least one node in common. This criteria alone is, however, not always sufficient, which is why production rules with individual preconditions and postconditions are specifically defined for most kinds of horizontal edges. For example, in order to horizontally connect a ridge and a hip ending node, their sets of vertically connected lower-level nodes must have two nodes in common as shown in the following production rule.



Postcondition:

Hip ending segment must be close to one ending of the ridge

In reality, however, a hip limits a ridge by a sloped roof surface. For this, the ridge must intersect the segment of the hip ending node. As this intersection is not always actually happening, a postcondition is defined to ensure that the intersection point of the elongate straight line of the ridge and the plane defined by the hip ending segment is at least close to one ending of the ridge feature. If the distance is, however, greater than a predefined threshold, then the production rule is not applied to the matched subgraph. In contrast, the postcondition of the production rule that introduces horizontal edges between a ridge and a Dutch gable ending node ensures that the intersecting point is not close to the surface points of the Dutch gable ending segment. In similar way, further production and postconditions are defined to derive horizontal edges between different building features of the same level that are in spatial proximity.

For the subsequent three bottom-up phases, further production rules are analogously defined. Nodes that are horizontally connected in the multi-scale knowledge graph are aggregated to a higher-level node if they compose a primitive, super-primitive, or the whole building model. The LHS of a production rule that derives a primitive always consists of five horizontally connected nodes of which one represents a basic component and the others ending components. Two primitive nodes become generally horizontally connected if their segments of which they are composed are horizontally connected. In order to derive super-primitives, production rules are defined whose LHS consists of two horizontally connected primitive nodes. Note, due to the applied solid modeling technique described in chapter 6, all distinct cases of a T-, L-, or X-shaped as described in subsection 5.1.4 can be derived with the same production rule. A distinction in the multi-scale knowledge graph between them is thus not necessary. In order to avoid abnormal acute or straight connections between two connected primitives, postconditions are defined that test the x-y angle between their main orientations to decide whether or not the production rule is finally applied to a matched subgraph. In the last bottom-up phase, the GG is applied to aggregate the maximal connected subgraph of the

second level of the multi-scale knowledge graph with the largest number of nodes to a building model node. This ensures that the building model is only composed of a single connected component and that building primitives are thus not disconnected in a single building.

In general, it is not necessary for the recognition of building features to differentiate between building features that belong to a roof superstructures and those that are part of a base roof. However, the recognition of roof superstructure features is usually more difficult because their segments consist often only of a low number of surface points. This can result in inexact orientations of the involved segments and in features with a short geometric extent. Therefore, the orientation and length thresholds in the recognition rules are relaxed for those features that are part of a roof superstructure. A feature is in this context defined to be part of a superstructure if the surface points of the segments that form the feature are contained within other larger roof segments regarding their 2D geometry.

5.2.3 Top-Down Phases

The derived multi-scale knowledge graph based on the bottom-up phases alone might be incomplete due, for example, to (partial) occlusions, too strict thresholds or a combination thereof. Therefore, a top-down phase is initiated after each bottom-up phase to add weak or only partly recognizable higher-level building features and missing edges to the so far derived multi-scale knowledge graph. For this, production rules of the GG are selected and applied in a top-down manner by specifically searching for certain unrecognized lower-level building features and connections. The general workflow of a top-down phase is illustrated in Figure 5.13.

A top-down phase is initiated after the building features of level i of the multi-scale knowledge graph have been detected in a bottom-up manner based on the building features of level $i-1$. First, the node subset V^* is determined that consists of all detected nodes of the next lower level $i-1$ that have not been aggregated to a higher-level building feature during the previous bottom-up phase. If V^* is empty, then it is assumed that all building features of level i have been detected and the top-down phase is already stalled. In this case, the multi-scale knowledge graph derivation process is either continued with the next bottom-up phase or ended if the top level of the multi-scale knowledge graph has been already derived. Otherwise, the nodes in V^* are further examined to ensure whether or not they belong to any unrecognized higher-level building feature of the actual building. Without the top-down phases, these nodes and thus their associated building features would otherwise not be incorporated in the subsequent recognition stages and in the later reconstruction process because they are not part of the aggregations that compose the building model node.

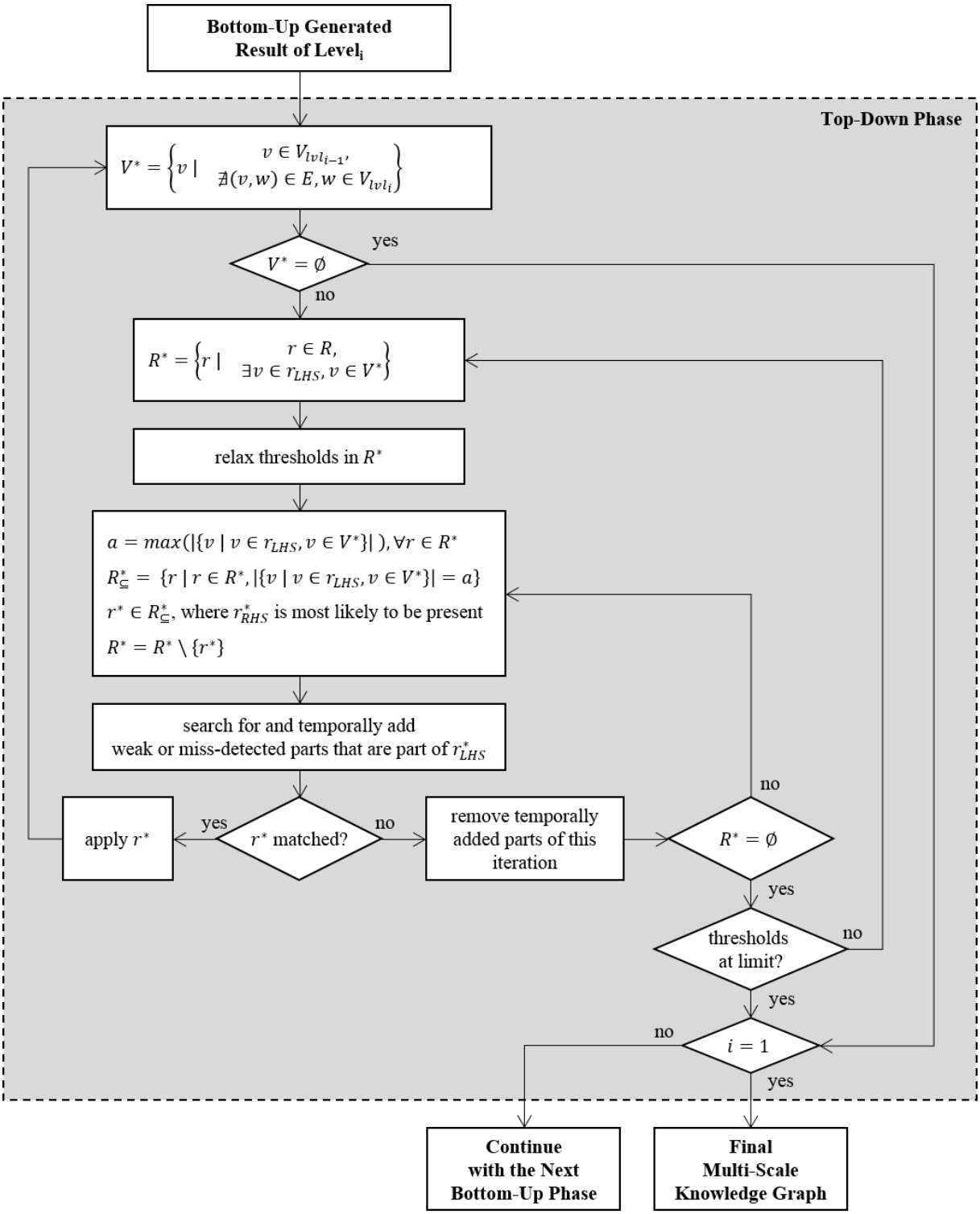


Figure 5.13. General workflow of a top-down phase.

With the intention to decide if nodes in V^* are part of a higher-level building feature, the production rule subset R^* is determined. It consists of all production rules whose matched LHS have an equivalent node in V^*. Consequently, they can be potentially applied to aggregate these nodes to a higher-level building feature. The thresholds in the resulting subset of production rules R^* are then relaxed, which mainly decreases the required confidence values

and the thresholds in the postconditions. Depending on the number of nodes in V^* and the defined set of production rules, R^* usually consists of a number of different production rules. In order to keep the number of production rules low that have to be tested during a top-down phase, all production rules in R^* are determined whose LHS comprise the highest number of nodes from V^*. From this resulting subset $R^*_{\subseteq}$, the production rule r^* is selected whose RHS is most likely to be present in the actual building.

In order to apply r^* so that nodes in V^* become part of a higher-level building feature, weak and miss-detected parts that are part of r^*_{LHS} are specifically searched and temporally added to the multi-scale knowledge graph. If r^*_{LHS} can be afterward successfully matched, r^* is applied and the top-down phase continues again with the determination of V^*. Consequently, the remaining production rules in R^* are not further tested for the formerly node set V^*. Otherwise, if r^*_{LHS} cannot be matched by incorporating the added weak nodes, then all temporally added building features are removed, the tested production rule r^* is discarded, and the next most likely production rule with the highest number of nodes from V^* in its LHS is tested.

Once all production rules in R^* have been tested but there are still some lower-level building features in level $i-1$ left that have not been aggregated to a higher-level building feature, then a new subset of production rules R^* is determined for the remaining nodes in V^*. Note, by each redetermination of R^*, the thresholds in the production rules are lowered until they reach their limit so that new higher-level building features can be derived. Thus, the top-down phase is stalled either if all lower-level building features are aggregated to a higher-level building feature, i.e. V^* is empty, or if all possible higher-level building features are discarded to be derived, i.e. the thresholds of their production rules cannot be further lowered.

A major aspect of the above described workflow is the selection of the most likely production rule r^*. As mentioned before, subgraph matching is NP-complete and therefore not well suited to large input. It would cause much processing time, especially for large buildings with many superstructures, if all productions in $R^*_{\subseteq}$ need to be successively tested and applied via subgraph matching to determine the most likely production rule. In order to overcome this problem, each production rule is assigned with a score that refers to the expected likelihood of occurrence for its RHS in the input data set. Building features that are represented in the RHS of a production rule with a high score are thus more likely to be present in the data set than building features that are derived by a production rule with a low score. During the top-down phases, the score dictates the order in which productions are tested. Consequently, production rules with a high score are more likely to be applied during the top-down phase than productions with a low score. Analogous to the probabilities of a stochastic (context-free) grammar, the score of a production rule thus indicates the likelihood of the production to be applied. However, in contrast to a stochastic (context-free) grammar, not the probabilities of all productions with the same LHS must sum to one (i.e., $\sum_{i=1}^{|LHS|} p(LHS \rightarrow RHS_i) = 1$, where $LHS \rightarrow RHS_1 \mid RHS_2 \mid \ldots \mid RHS_{|LHS|}$) but the scores of all production rules of the proposed GG.

The initial production rule scores are empirically determined based on the previous knowledge about the reconstruction area. For the adaption of the scores during the multi-scale knowledge graph derivation process, reinforcement learning is applied to the GG. Thereby, the score of a production rule is adapted according to its number of matches during the bottom-up/top-down phase. Furthermore, in order to satisfy the observation that neighboring buildings often feature the same roof type, knowledge of already derived multi-scale knowledge graphs is incorporated in subsequent derivations. The production rule scores are, therefore, not only based on the current building to be reconstructed but also on already derived multi-scale knowledge graphs. For this, the initial scores of matched production rules are increased for the next building while the initial scores of unmatched production rules are decreased. As a result, the recognition of similar building parts is supported for neighboring buildings.

5.2.4 Segment Splitting

After a set of matched subgraphs is determined according to the LHS of a production rule, a segment splitting procedure is carried out for certain building features. This is to take into account that more than one building feature of the applied production rule could potentially be derived from a single subgraph and that a segment may generally be part of several building features. In order to ensure that all building features are adequately represented in the multi-scale knowledge graph, segment splitting is applied to a segment if it is part of the surface of a building component but also has surface points that are outside this component. In this way, the segment splitting procedure supports the feature recognition step because the applied feature recognition rules can be better tailored to specific feature types, if it is safe to assume that segments with regard to their surface points meet certain size and shape criteria; making rules more effective for recognizing smaller roof structures. For example, two segments of a gable end dormer might not form a distinct ridge due to a slight skewness in their orientation. But if the dormer segments have the typical size of dormers, are located above segments of the base roof and no other segments are in their adjacency, than it is almost safe to assume that they do form a dormer. However, if one side of the dormer is part of a larger segment, than such a rule does not apply. Segment splitting is not a specific problem caused by sub-surface segmentation, but is needed whenever a roof segment is part of many features. Two examples where segment splitting is required for the green segment are presented in Figure 5.14. On the left side, two adjacent flat roofs of the same height feature canopies with different slopes. Analogously, on the right side, two adjacent gable roofs feature only on one side a coplanar plane that requires thus a segment splitting.

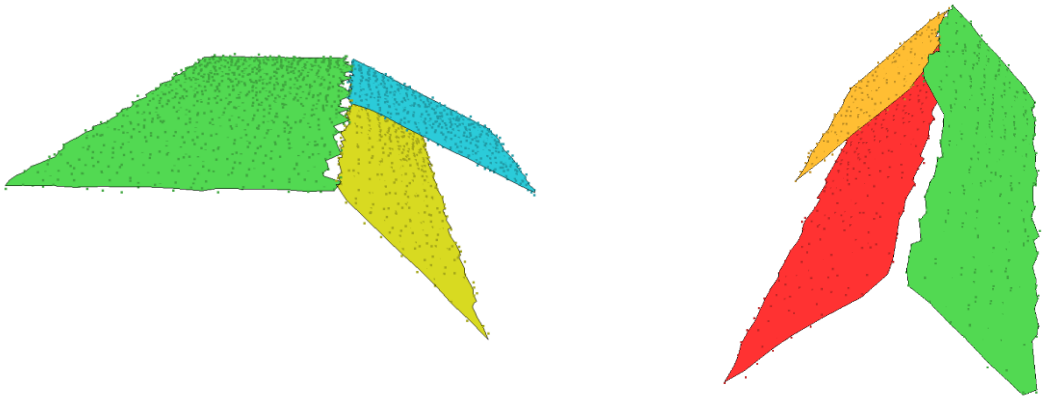
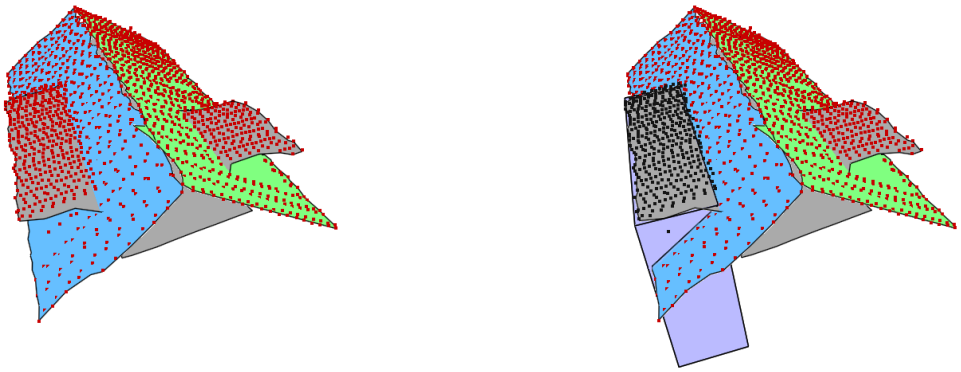


Figure 5.14. Examples where segment splitting is required.

In Figure 5.15, another example is given that requires a segment splitting. The blue building components in this subsection are included solely for illustration purposes. Those surface points that are already part of a feature defining building component that is going to be derived are colored in black, otherwise in red. As shown in Figure 5.15a, sub-surface segmentation provides instead of four only three distinct planar segments. This is because the smaller segments are coplanar and they are therefore merged whenever virtual sub-surface points are taken into account. Thereby, the grey segment becomes part of two dormer features. Once, a flat top dormer on the left side of the building has been detected as shown in Figure 5.15b, segment splitting checks if the grey dormer segment contains further surface points that are outside the detected dormer feature. If this is the case, it needs to be split. Otherwise the segment splitting procedure would stall.



(a) (b)

Figure 5.15. Example of segment splitting needed to model the two dormers originating from the same sub-surface segment (part 1).

The segment splitting operation itself is composed of a cloning step and a reclassification step with regard to surface and sub-surface points. In Figure 5.16, the general workflow is illustrated on the basis of the example in Figure 5.15. In the cloning step, an exact copy of the segment is constructed. Then the surface points of both segments are reclassified. All surface points of the original segment are turned into sub-surface points if they are located outside of

the building component. Consequently, it is not used anymore for the detection of further features in the next iteration. Vice versa, all surface points of the cloned segment are turned into sub-surface points if they are located inside the building component. Consequently, these surface points are part of not yet modeled components and can be still used to identify further features.

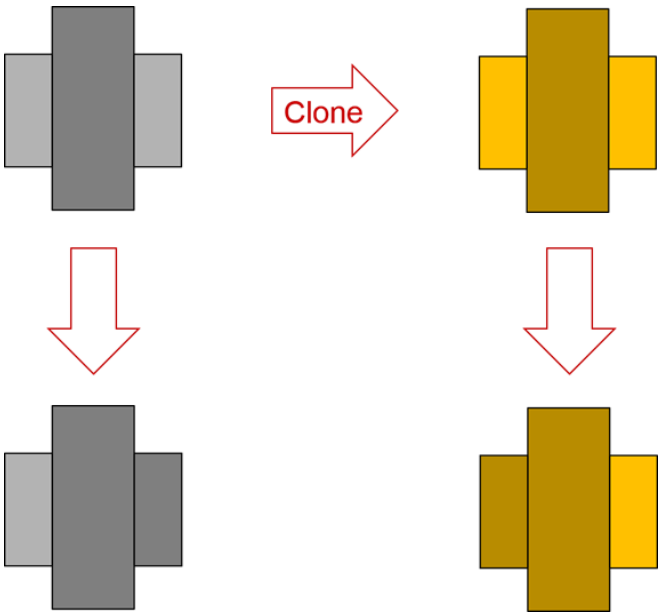
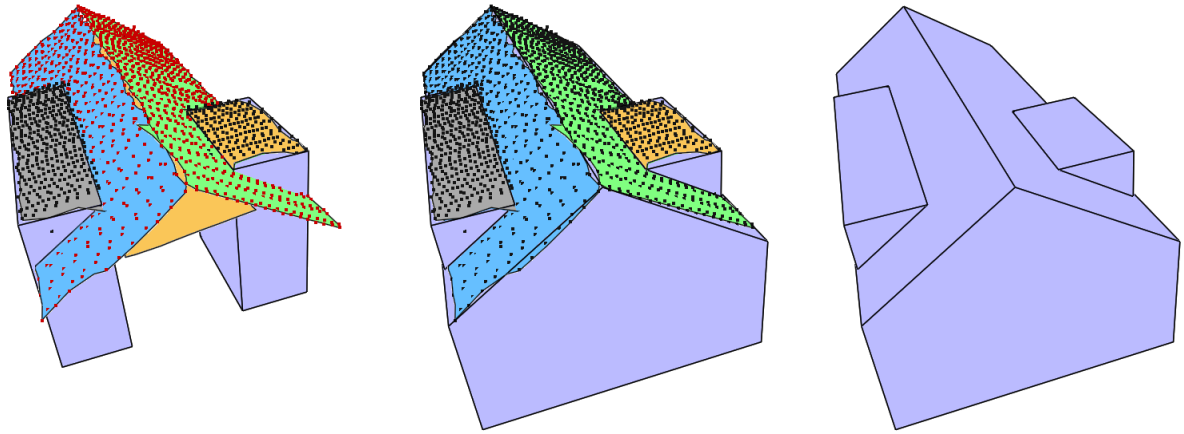


Figure 5.16. General workflow of segment splitting. Light and dark color gradation represent surface and sub-surface areas respectively.

After the grey segment in the example of Figure 5.15 has been split, another dormer can be recognized as shown in Figure 5.17a. At this point, no more split operation is required because all surface points of the segment are now located inside features. The reconstructed building model is shown with (Figure 5.17b) and without overlaid (Figure 5.17c) segments.



(a) (b) (c)

Figure 5.17. Example of segment splitting needed to model the two dormers originating from the same sub-surface segment (part 2).

The reason why segments are not actually split into two parts is that surface points are often part of more than one building component. If segments are just geometrically split, points are assigned to one segment and information is lost to the other segment. Therefore, these points are kept in all segments even if the new segment considers them only as sub-surface points. On the other side, if the segment splitting operation just consists of a cloning step without a subsequent reclassification step, the cloned segment would be used to generate the same feature over and over again. With the proposed segment splitting approach, no new surface points are created, but the cloned segments have the information of all points available to generate features and building components.

6. Building Model Construction Using Half-Spaces

A crucial aspect of any automatic reconstruction workflow is the selected solid modeling technique for the geometric construction of building models. A general overview of the most common solid modeling techniques was given in section 3.3. Regarding building model construction, many automatic reconstruction approaches have generated 3D models directly in boundary representation (Oude Elberink and Vosselman, 2009; Sampath and Shan, 2010), using Boolean operations on (parameterized) solids with basic roof shapes (Haala and Brenner, 1999), cell decomposition (Kada and McKinley, 2009), binary space partitioning trees (Sohn et al., 2008) or a combination thereof (Lafarge and Mallet, 2012). Generally, any solid modeling technique can be used to convert the segments based on the semantic information of the multi-scale knowledge graph into a geometric model. In this reconstruction framework, however, half-space modeling is selected because it provides several advantages for the reconstruction of building models such as, among others:

- Segments can be directly used to define the hyperplanes of half-spaces. The fact that these hyperplanes coincide with building roof surfaces makes half-space modeling to a natural choice when it comes to building reconstruction.
- Half-space modeling, as it is applied, guarantees that the resulting building models are always closed solids without unintentional gaps between roof surfaces.
- Occluded parts and holes in the input data are automatically filled. Buildings are, therefore, completely constructed even if they are partly covered by vegetation or high-rise constructions.
- With half-space modeling, buildings can be defined in a compact way. The number of half-spaces needed to construct a complex building is rather low.
- Half-space modeling is generally not limited to any shapes and thus provides a high flexibility.
- The construction based on half-spaces opens up the possibility to test the location of points with regard to building components that might be still incomplete. If points are in front of any single half-space, it is also outside the component despite the fact that more half-spaces could be included in the future. Therefore, there is no need for this purpose to intermediately convert the model into another representation such as B-rep.
- All building types can be represented in the same way, which is the canonical form.
- Building components that are combined to define more complex buildings do not need to be disjoint and may intersect just like the segments obtained from sub-surface segmentation.
- The reuse of hyperplanes in connected building components provides a high degree of flexibility if one component needs to be adjusted.

The use of half-space modeling for the reconstruction of a base roof (a roof without any kind of roof superstructures) is stated in detail in section 6.1. Here, the planar segments, in combination with the derived building knowledge, are utilized to introduce and adjust half-spaces that define a closed building model. For visualization purposes only, these models can afterward be directly converted to B-reps. Most CAD kernels offer such functionality. This guarantees that not only the half-space model but also the B-rep of a building is always closed. The so far presented reconstruction framework is well suited to generate realistic 3D building models for large urban areas because it is also quite robust against partial occlusions, which cause gaps in the input data set, as long as sufficient building features can be recognized.

Generally, the so far presented reconstruction framework is able to recognize and reconstruct not only base roofs but also roof details, such as roof superstructures if the aerial LiDAR input data meets the required quality (e.g., in terms of density, accuracy, and completeness). However, if this is not the case, only an insufficient set of building features might be derived so that coarse and generalized building models are created. Here, roof superstructures are especially affected because they are usually represented by only few point measurements so that they are often discarded during either the segmentation or the building knowledge derivation step. Consequently, the resulting level of detail is reduced in the building models. However, today's requirements for building models are demanding so that building models are expected to include detailed roof superstructures regardless of the input data quality. Current automatic reconstruction approaches quickly reach their limits in this context because measured points from roof superstructures such as chimneys and dormers are, due to their low number of points, usually considered as noise in low-density point clouds and are therefore ignored during the reconstruction process (see section 2.3). Some approaches have been developed that offer users the possibility to add them in a semi-automatic way or that utilize additional data sources like images (Rottensteiner and Briese, 2003; Habib et al., 2010). But most of them are not applicable in practice for a fully automatic large-scale reconstruction process due, for example, to an associated timescale or missing additional data sources.

In order to solve this problem, an advanced automatic recognition and reconstruction approach for small roof superstructures is presented in section 6.2. It makes use of the ICP principles to identify and simultaneously reconstruct regularized roof superstructures of similar shape. Thereby, it is able to handle not only partially occluded roof superstructures but also small roof superstructures in point clouds with low density. Thus, both the robustness of the reconstruction approach and the completeness of the resulting building models are increased. Additionally, because roof superstructures are reconstructed in groups, also those roof areas can be reconstructed in detail where either the segmentation process fails or only an insufficient number of building features are recognized in the building knowledge derivation step. Moreover, symmetries, alignments, and regularities can be enforced in a straight-forward way. Integrated into the so far presented reconstruction approach, the extension is able to reconstruct small roof superstructures in point clouds with less than 5 points/m². The extension is generally independent of the so far presented reconstruction approach so that roof details can be optionally added to a building model according to its purpose.

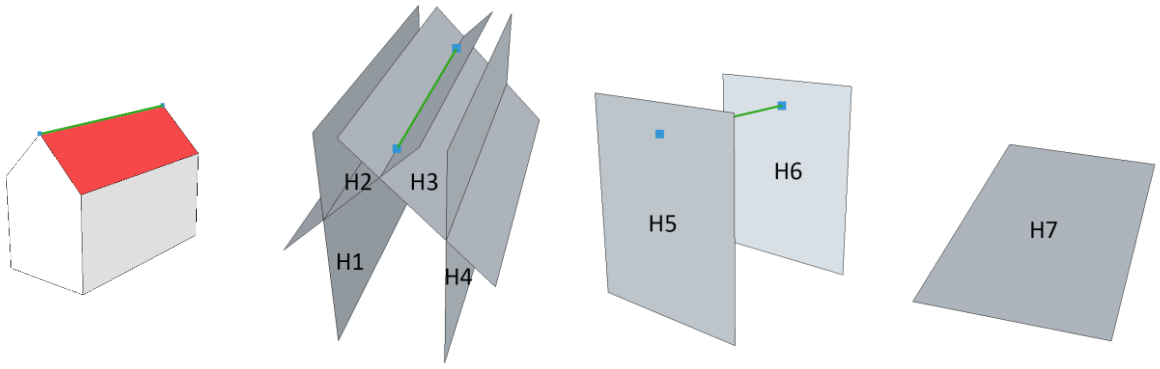
Finally, a 3D adjustment process is presented in section 6.3 that improves both the shape of the reconstructed building models and their accuracy. For this, natural regularities, as they are common in most man-made objects, are further incorporated in the reconstruction framework. Thus far, building regularities are individually defined for each feature of the multi-scale knowledge graph. In this way, semantic information is utilized to emphasize innate building regularities in the models. This approach, however, makes it cumbersome to adapt the rule set of the building derivation process to different data sets. Moreover, the regularities defined by a feature only have a local impact on a building model. In order to increase the flexibility of the reconstruction framework and to emphasize not only local but also global regularities, an adjustment procedure for the half-space representation of building models has been developed based on divisive clustering techniques.

6.1 Base Roof Reconstruction

For the geometric construction of building models, half-space modeling is selected, as it enables a one-to-one relation of planar segments to planar half-spaces and it therefore seems to be a natural choice for the construction of rooftops. This means that, in general, one half-space is generated for every segment. Thereby, the plane equation of a segment is directly used to define the parameters of a half-space. However, to avoid the misuse of segments during the reconstruction process, half-spaces are not directly created based on the presence of segments but they are introduced by certain features of the multi-scale knowledge graph. In this way, building regularities can be also directly enforced in the final building model according to the semantic meaning of a feature to which a segment belongs. Additionally, taking the interconnections of building features into account, further common building regularities can be easily emphasized, for example, by reusing previously defined half-spaces from other features.

Whenever half-space modeling is applied, some care has to be taken so that the generated objects always represent closed point sets. Since rooftop half-spaces originating from planar segments limit the space in the positive z-direction only, additional half-spaces are needed to limit the space in the x-y direction as well. If vertical wall segments of a convex building were available in the data, then both rooftop and wall segments could be mapped one-to-one to half-spaces and their collection, in combination with a ground plane, would be a complete description of a building. Therefore, half-space modeling is a very natural way to translate segments to the 3D model construction. However, vertical wall segments are only rarely present in airborne LiDAR data. To overcome this issue, certain features of the multi-scale knowledge graph are used instead to define half-spaces with vertical hyperplanes, which represent wall surfaces in the final building model. Thereby, special care has to be taken so that the set of wall half-spaces limits the space in any x-y direction. The set of rooftop and wall half-spaces in combination with a half-space that represents the ground plane always defines a closed point set. Here, the ground plane is defined as a half-space with a horizontally oriented hyperplane whose height is determined by the average height value of those non-building points that are close to the building.

By applying the Boolean set operation intersection to a collection of planar half-spaces, convex building models can be directly constructed. In Figure 6.1, an example is given for how a saddleback roof is constructed by seven half-spaces. The construction details of detected convex sub-shapes from the previous building derivation step are explained in subsection 6.1.1.



$$S = H1 \cap H2 \cap H3 \cap H4 \cap H5 \cap H6 \cap H7$$

Figure 6.1. Required hyperplanes to define a saddleback roof building S with half-spaces.

For the construction of concave building sub-shapes, a building generally needs to first be split into convex components that are then united with a Boolean union operation. An example as to how a Dutch gable roof is created based on two convex components is presented in Figure 6.2. In this regard, sub-surface segmentation forms a good basis to find a preferably small number of meaningful convex components as the segments are enlarged as much as possible and therefore opens up many opportunities for the building knowledge derivation step to group them together to form convex shapes. An advantage of half-space modeling is that these convex components do not need to be disjointed and they may intersect like the segments from sub-surface segmentation. Thus, the construction of a building model consists generally of only unions of half-space intersections so that any building can be easily expressed in a canonical form. In this way, many features and complex building shapes can be defined by a rather small number of planar half-spaces while still allowing for a certain flexibility.

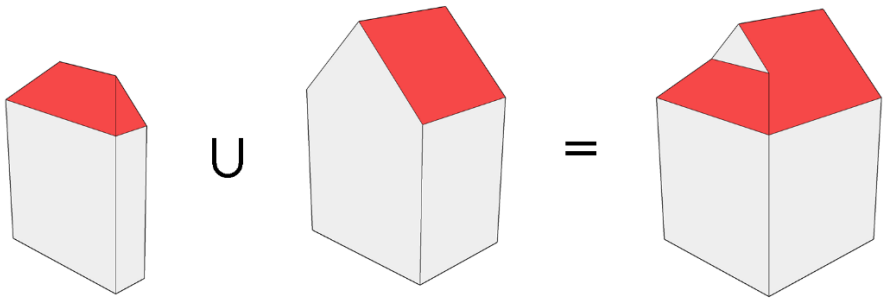


Figure 6.2. General approach for the construction of a building model with concave shape.

In order to reduce the number of half-spaces and Boolean operations, an enhanced version for the construction of some buildings with concave shapes is presented in subsection 6.1.2. For this, the number of decompositions for the construction of concave shapes is reduced by utilizing the Boolean difference operation. As a result, frequently occurring decompositions are avoided, as they are often caused by intrusions and extrusions.

For the geometric construction of detected links from the multi-scale knowledge graph, certain half-spaces from one sub-shape are reused for the construction of the other sub-shape and vice versa. The advantage of reusing half-spaces is not just the reduction of their overall number, but also that adjacent components become connected without any unintentional gaps between them. In addition, unwanted extrusions, which are present if the extent of one sub-shape is modeled so large that one component grows through the other component, are avoided. Moreover, if half-space parameters of one sub-shape are changed during the building adjustment step, then the other sub-shape is automatically changed as well. The details for the construction of connected sub-shapes are explained in subsection 6.1.3.

When the whole building is defined in half-space representation, a B-rep model is directly created for visualization purposes based on the half-space model. This also guarantees that the B-rep of the final building model is always closed. Some reconstructed buildings of the Vaihingen data set and their geometry producing building features are exemplarily shown in Figure 6.3. As demonstrated in the last row of this figure, the proposed reconstruction procedure is not only able to reconstruct base roofs but also roof superstructures if their segments and sufficient building features are recognized.

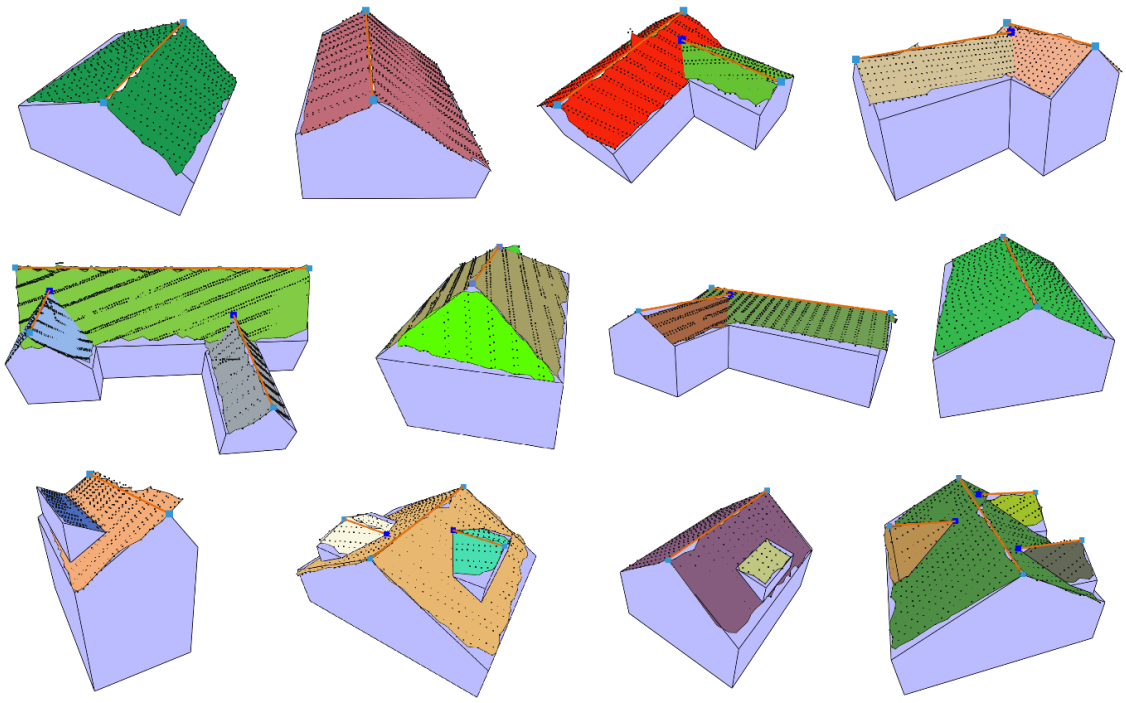


Figure 6.3. 3D point cloud with planar segments, some recognized features, and reconstructed building models.

6.1.1 Construction of Convex Sub-Shapes

To define a convex building shape with half-spaces, a collection of half-spaces needs first to be determined on which the Boolean intersection operation is then applied. As described before, building features in the multi-scale knowledge graph are not isolated entities but are interconnected and form a network. However, only semantically compatible features may be connected, which prevents the misuse of features. Once the semantic connections are established, the information from connected features can be used to determine the collection of half-spaces to define the building. Here, only primitive components, depending on their type and context, are allowed to introduce half-spaces for the construction process. The half-space parameters themselves are computed by the plane equation and extent of segments, and they also consider the geometric and semantic information of connected features. For example, the normal vectors of the two saddleback roof surfaces usually point in the x-y plane into the opposite direction so that their intersection forms a horizontal ridge line. By keeping a set of such constraints for each feature that defines a half-space, regularities are explicitly emphasized in the final building model. The half-space defining features can generally be grouped into geometry producing and geometry refining features. Examples of both are presented in the following paragraphs.

Geometry producing features constitute the basis of the construction stage by providing initial collections of half-spaces defining elementary convex building components. There are mainly three different features for this purpose. Firstly, the flat feature, which represents a locally elevated single surface and therefore defines one half-space. It is used as the basis for both simple flat and shed roofs, but also for more complex roof shapes like the platform roof, which is a horizontal roof with steeply sloped surfaces to its sides. Secondly, the ridge feature, which defines two half-spaces. A ridge feature is used as a common ground for a number of roof shapes with two or more sloped surfaces like the gable, saltbox, hip, gambrel, mansard, gull wing, or saddleback roof. Thirdly, the tip feature, which defines at least four half-spaces. This feature is used for roof shapes with at least four sloped surfaces like pyramidal, tented, helm, or cone-type roofs. The hyperplanes defining half-spaces, which are introduced by these three geometry producing features, are exemplarily shown in Figure 6.4.

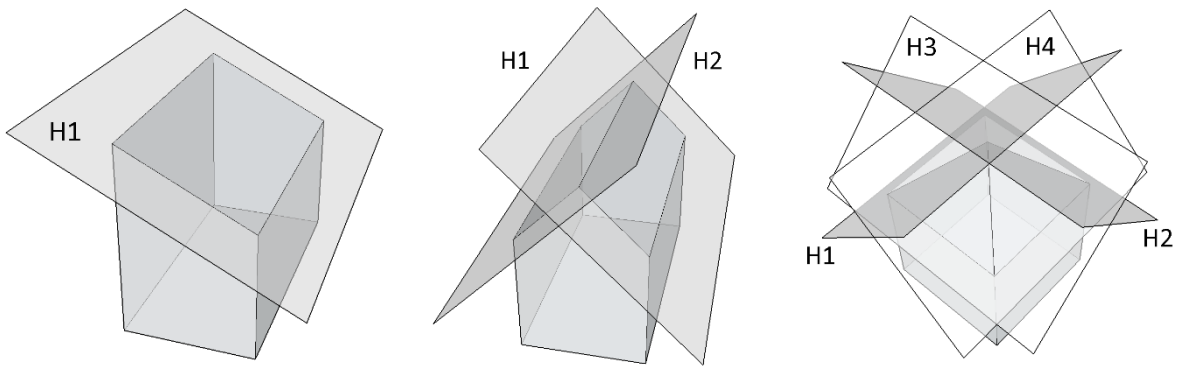


Figure 6.4. Rooftop half-spaces defined by geometry producing features.

Geometry refining features, in contrast, add details to coarse building blocks by inserting one or more half-spaces to the geometry producing feature it is associated with. For example, a gable end feature adds a vertical half-space that delimits the geometric extent of the roof in that direction. Another geometry refining feature for a building with a ridge line is the hip end feature. As illustrated in Figure 6.5, it adds one half-space H1 with the plane parameters of the (half-)hip segment to limit the geometric extent of the ridge feature and one vertical half-space H2 to further limit the extent of the building in x-y direction. Note, that all kinds of hip ends are constructed in the same way. For this, the hyperplane H1 is either moved along its normal vector or its slope is changed. In this regard, the hip end half-spaces are directly adjusted so that their normal vectors point in the same x-y direction as the horizontal intersection of the two half-spaces that are defined by the ridge feature. Consequently, the hip end surface of the final building model is always orthogonal in the x-y plane to both roof surfaces that form the ridge.

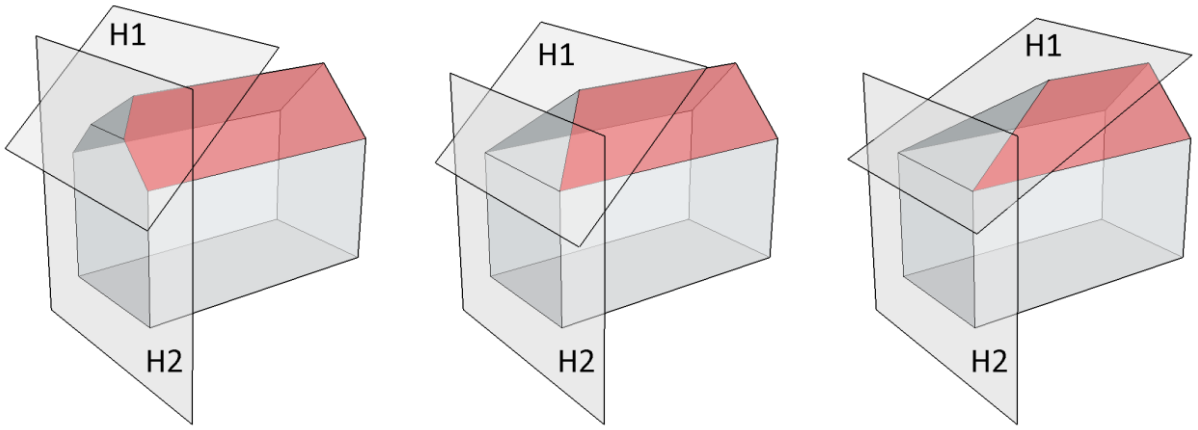


Figure 6.5. Half-spaces defined for different kinds of hip endings.

In a similar way, gambrel and mansard roof sides are constructed. Both building types have on each side a shallower slope above a steeper one so that their surfaces still form a convex object. The two half-spaces defining the shallower sloped roof surfaces are already introduced by the ridge feature. For their refinement, each feature that represents a sharp transition between both slopes introduces as a geometry refining feature one half-space to refine the rooftop and one half-space from an adjacent eave feature to limit the extent of the building in the x-y direction. As shown in Figure 6.6, these additional half-spaces refine the coarse building block that is already defined by the two half-spaces of the geometry producing feature to which they are associated with. Thus, four half-spaces are defined in total to construct the rooftop of a gambrel roof and two further half-spaces for its sides. In this respect, the x-y orientations of the half-space normal vectors are adjusted by taking into account the geometry of the geometry defining feature so that parallel structures are supported in the final building model.

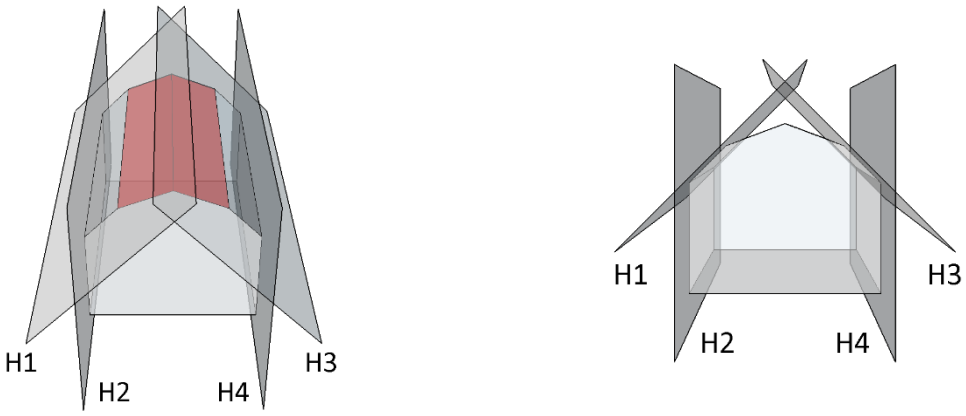


Figure 6.6. The four refining half-spaces of a gambrel roof.

6.1.2 Construction of Concave Sub-Shapes

The construction of buildings using Boolean intersections of planar half-spaces can only result in convex buildings. Therefore, the construction of a building with a concave shape is usually accomplished by partitioning the building into convex components, modeling each component separately, and finally uniting them by a Boolean union operation. According to this procedure, a gable roof building with a simple intrusion as shown on the left side of Figure 6.7 must be partitioned into three components. In particular, the ridge feature and each of its two segments have to be split into three parts. The rooftop half-spaces of these parts often become unstable due to their small number of supporting points. As a result, roof parts originated from the same segment are sometimes modeled with different slopes or shifted ridge lines. Furthermore, due to the fact that each convex component is constructed separately, further effort is required to ensure that parallel and coplanar walls and roof surfaces keep these properties in the final building model. Also, considering the computational time, the number of required half-spaces and Boolean operations to construct the building in Figure 6.7 is, with 21 half-spaces and 20 Boolean operations (each component requires six intersections and two unions are needed to combine all components), rather high. Even considering the option to construct the bottom of all convex components together at the end would decrease their number only by two.

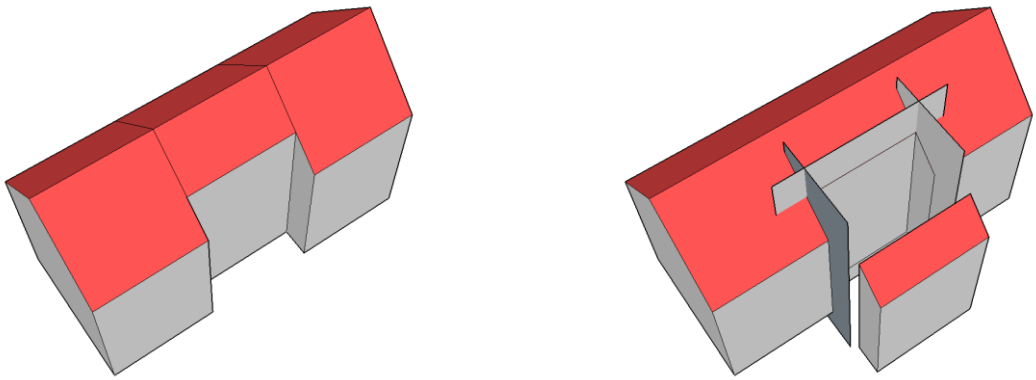


Figure 6.7. Two different approaches to model a gable roof with an intrusion.

In order to avoid such kind of anomalies and to reduce the number of required half-spaces and Boolean operations, another variant is proposed. Here, buildings are first modeled without any intrusions. Then, in a subsequent step, all intrusions are defined and removed by a Boolean difference as exemplarily shown on the right side of Figure 6.7. Note, the effort of a Boolean difference operation is in this case almost the same as a Boolean intersection or Boolean union operation because the Boolean difference of two elements A and B can be defined by Equation (6.1) or Equation (6.2).

$$A - B = A \cap B^C \tag{6.1}$$

$$A - B = (A^C \cup B)^C \tag{6.2}$$

Both equations are, due to De Morgan's laws, obviously equivalent. By considering that the complement of a closed half-space H^+ can be for practical reasons defined as the closed half-space H^- (see subsection 3.3.2), the effort of applying a Boolean difference is almost the same compared to a Boolean intersection or union operation. As a result of making use of a Boolean difference operation, the roof surfaces on each side of the ridge remain coplanar and the number of required half-spaces and Boolean operations is decreased from 21 to ten and from 20 to nine, respectively. For this, some features like the eave feature are allowed to decompose so that an intrusion part can have a vertical half-space parallel to the original vertical half-space of the eave. The intrusion parts for a saddle back building are, for example, determined by utilizing the divisive clustering principle as follows:

1. For all alpha-shape points of a ridge segment, calculate their perpendicular distance to the ridge and sort them accordingly.
2. Find all clusters containing a number of points greater than a predefined threshold.
3. Each of these clusters – except the minimum and maximum cluster – is partitioned into a minimal number of subsets such that the consecutive distance of any two neighboring points in a subset does not exceed a predefined threshold.

Each subset of points determined in this way defines the length and the position of an intrusion so that three half-spaces can be defined per intrusion.

A similar principle is applied when it comes to the construction of a building part with a concave rooftop whose complement forms a convex shape. For example, instead of decomposing a gullwing roof into three convex components as shown on the left side of Figure 6.8, the whole building is constructed at one time, as illustrated on the right side of Figure 6.8. For this, each of the two building features that represent the sharp transitions between the two slopes on the roof sides defines the space in front of their related segments. Thereby, the complement of the half-spaces that have been introduced by the ridge feature are reused. Consequently, only one further rooftop half-space needs to be introduced for each side. Once the convex outside of the gullwing roof is defined on each side, they are removed by a Boolean difference operation. As a result, the total number of Boolean operations used to construct a building with a gullwing roof is decreased from 18 to eight. In this way, simple

concave components that frequently occur are also directly constructed without decomposing them into convex components.

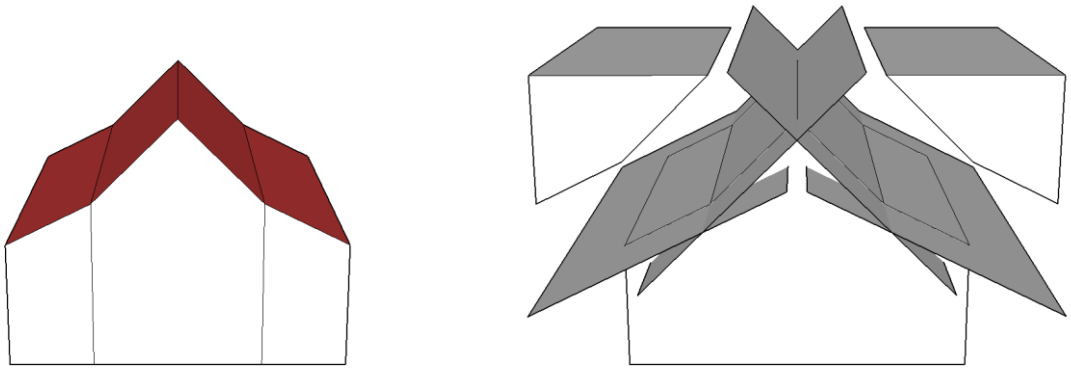


Figure 6.8. Construction of a gullwing roof. Once based on a decomposition (left) and once constructed with Boolean difference (right).

The capabilities to model and to construct buildings in this way become even more apparent for more complex buildings where both presented enhancements are applied together.

6.1.3 Construction of Connected Sub-Shapes

To construct links between sub-shapes in a building model, hip and valley features that are associated with two or more geometry producing features are used as geometry refining features. In contrast to a hip end feature, these features do not introduce new half-spaces but they cross-link already existing half-spaces. In this way, building links are not constructed as separate components but as part of sub-shapes. As, for example illustrated in Figure 6.9, the three distinct cases of a T-shaped link are all constructed in the same way. Only the two half-spaces H1 and H2 from component C2 are added to the intersecting roof part of component C1 while the intersected roof part is not changed. One half-space with the plane equation of the sloped segment of the intersected roof part that is located on the opposite side of the intersecting ridge is needed to model the broken hip, and a vertical half-space that is a vertical half-space of C2 further delimits the intersecting roof part.

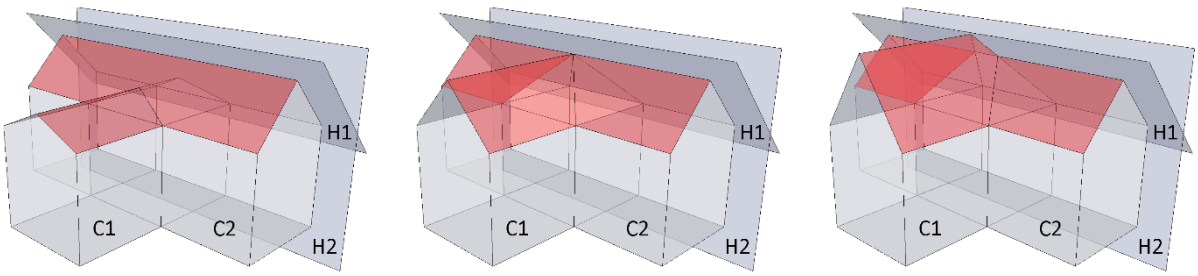


Figure 6.9. Half-spaces H1 and H2 from component C2 are used to delimit component C1.

Analogous, the components of an L-shaped intersection are symmetrically delimited by the half-spaces of the other component. As shown in Figure 6.10, component C1 is delimited by the two half-spaces H3 and H4 of component C2, and component C2 is delimited by the two half-spaces H1 and H2 of component C1. The construction is thus again independent of the two ridge heights.

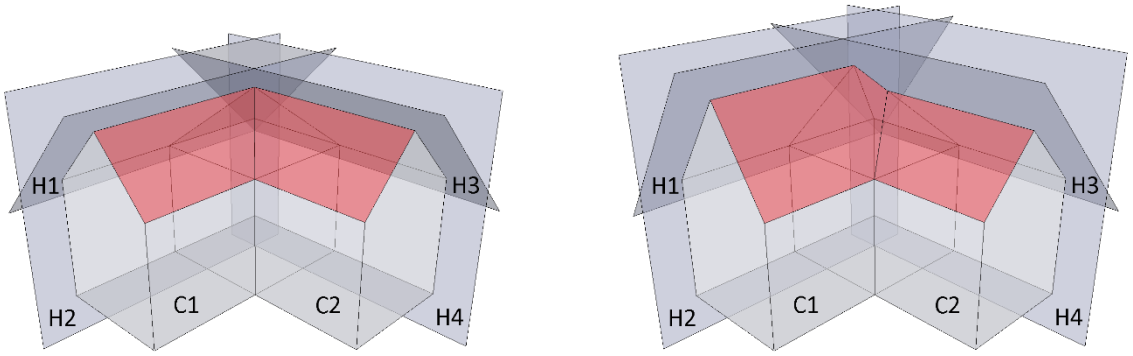


Figure 6.10. Construction of L-shaped intersections performed by a pairwise interconnection of half-spaces of the two components.

Note, since all building components are subsequently combined by Boolean union operations, they do not need to be disjoint to form well-shaped and valid building models. In similar way, further links between sub-shapes are defined.

6.2 Roof Superstructure Reconstruction

For the reliable recognition and reconstruction of small roof superstructures in low-density point clouds, an enhancement of the automatic building reconstruction approach is presented in this section. The basic idea is to take advantage of the fact that often more than one instance of the same superstructure occurs on the same roof as, for example, mentioned in (Oude Elberink, 2008). Instead of reconstructing each superstructure independently from one another, the proposed approach first detects all instances of a superstructure and reconstructs them afterward concurrently. Thus, information of one instance can be used for the construction of another instance so that a more accurate model is generated. The general workflow is illustrated in Figure 6.11 and consists of the following three steps: (i) detection of appropriate roof superstructure candidate points; (ii) grouping of similar roof superstructures; (iii) roof superstructure modeling and construction.

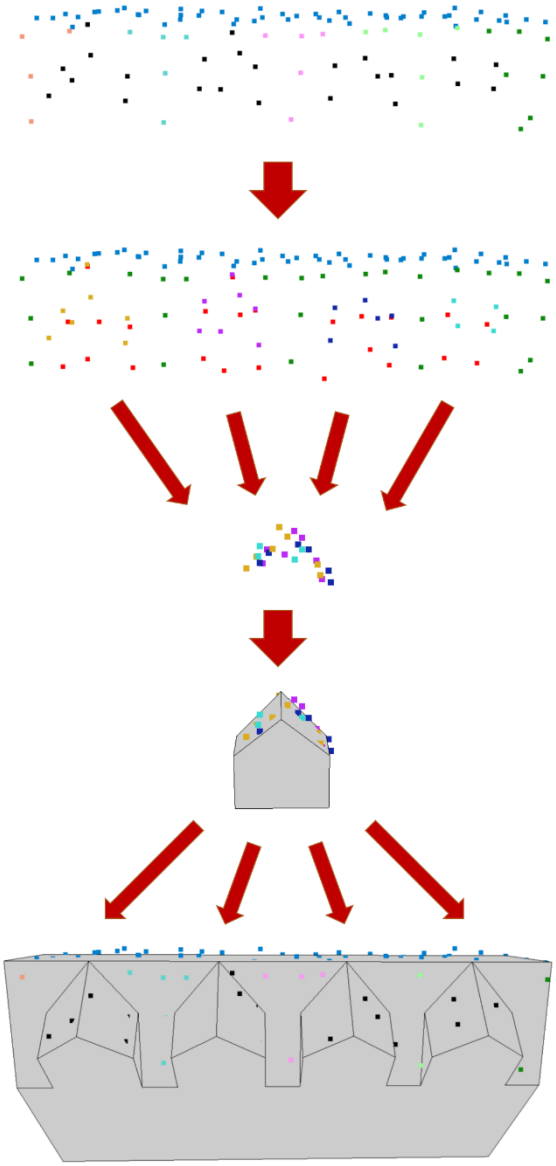


Figure 6.11. Reconstruction workflow of small roof superstructures. From top to bottom: segmentation result (unsegmented points colored in black); segmentation result after adding virtual points (colored in red), and estimation of candidate areas; transformation result of the candidate points; superstructure reconstruction result; reconstruction result of the whole building.

In order to determine appropriate candidate points from the input data, the approach makes use of virtual sub-surface points that are assumed to lie on the base roof faces below the measured points. In conjunction with those surface points that either do not belong to any segment or that do belong to already detected roof superstructure segments, candidate areas of roof superstructures and their surface points are identified. The details of this procedure are explained in subsection 6.2.1. Once all surface points from potential roof superstructures are collected, candidate areas with similar roof superstructures are detected, extracted, grouped together, and registered to one another. For this, a point cloud registration technique based on

the ICP algorithm (see section 3.5), which considers basic building knowledge, is applied as shown in subsection 6.2.2. As an outcome, the joint point density of each detected group is increased whenever more than one instance of the same roof superstructure is present in the real building. The higher point density, in turn, enables the building knowledge derivation process presented in chapter 5 to recognize the shape of the roof superstructure in a second attempt more reliably and in more detail. Afterward, only one roof superstructure model is created for each superstructure type and instances of it are then transformed back to their original position. Finally, the whole building model is checked for overlapping instances and, if necessary, corrected. The recognition and construction details are explained in subsection 6.2.3. The presented method for the reconstruction of roof superstructures is integrated in the automatic building reconstruction workflow as shown in Figure 6.12.

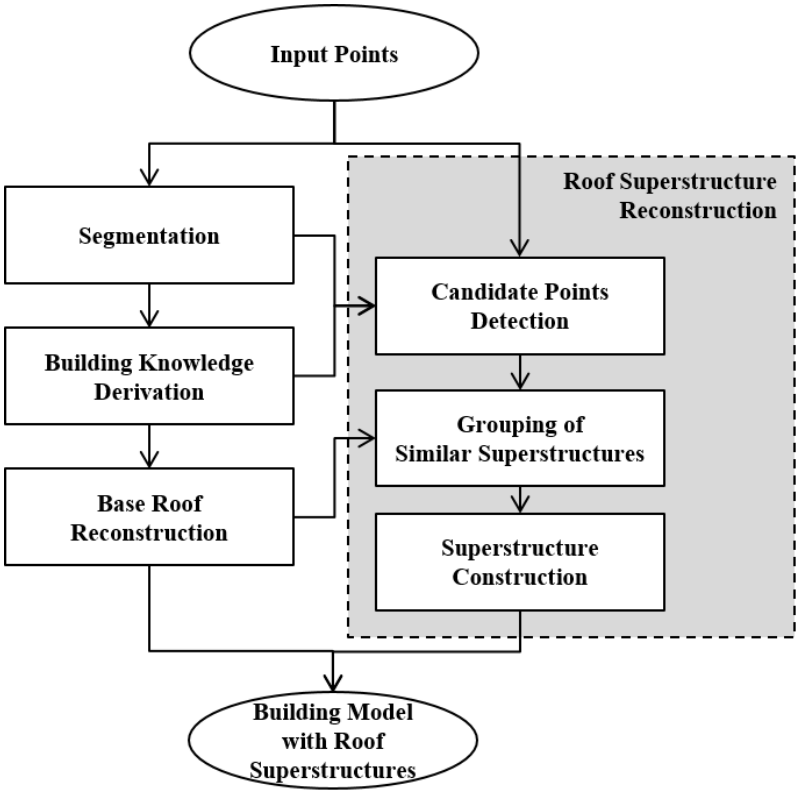


Figure 6.12. Overview of the proposed reconstruction approach of regularized roof superstructures in low-density point clouds that has been integrated in the building reconstruction approach.

For evaluation purposes, the proposed enhancement has been particularly tested both on several buildings located in residential parts of the Vaihingen test data set (4–6 points/m²) and on generated data with a lower density (1.5–4 points/m²) but with similar properties. At the moment, the implementation is limited to those roof superstructures that are located on a single roof plane. As shown in Figure 6.13 and Figure 6.14, for this dominant kind of roof superstructures, the approach presented in this section is generally suitable for their automatic reconstruction in low-density point clouds. In particular, the reconstructed buildings of the generated data feature rooftop details that would otherwise be missed.

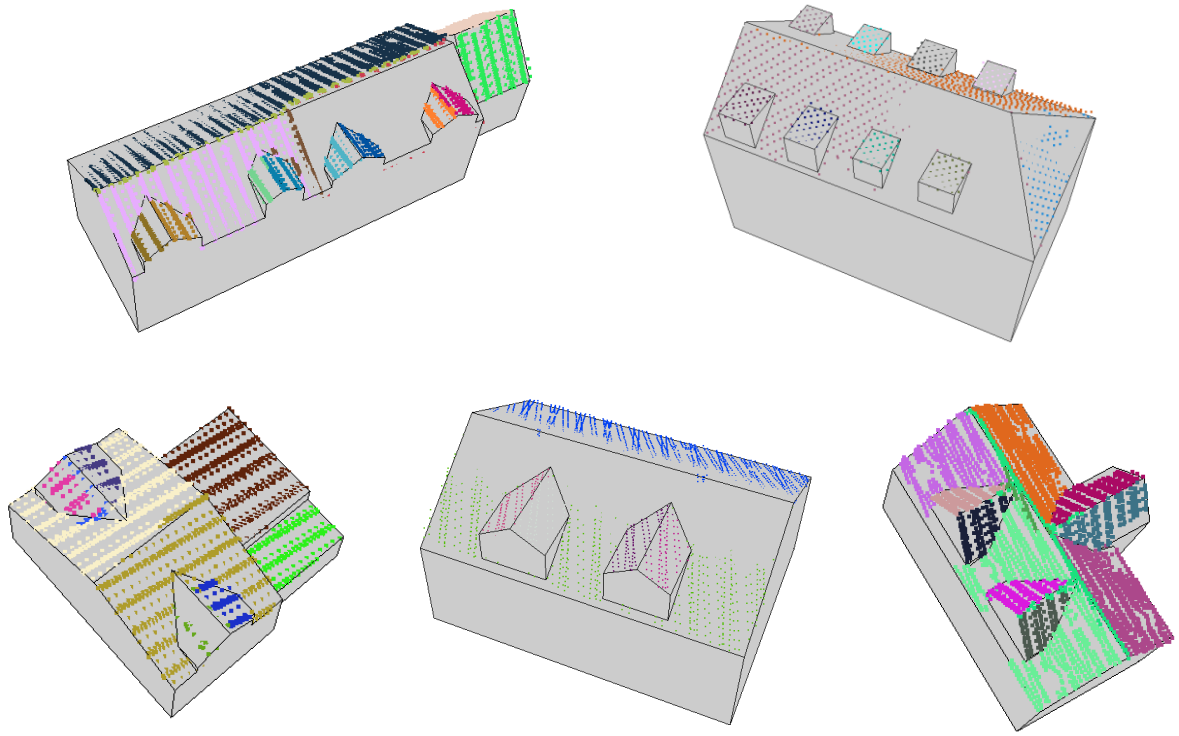


Figure 6.13. Some reconstructed buildings of the Vaihingen test data set.

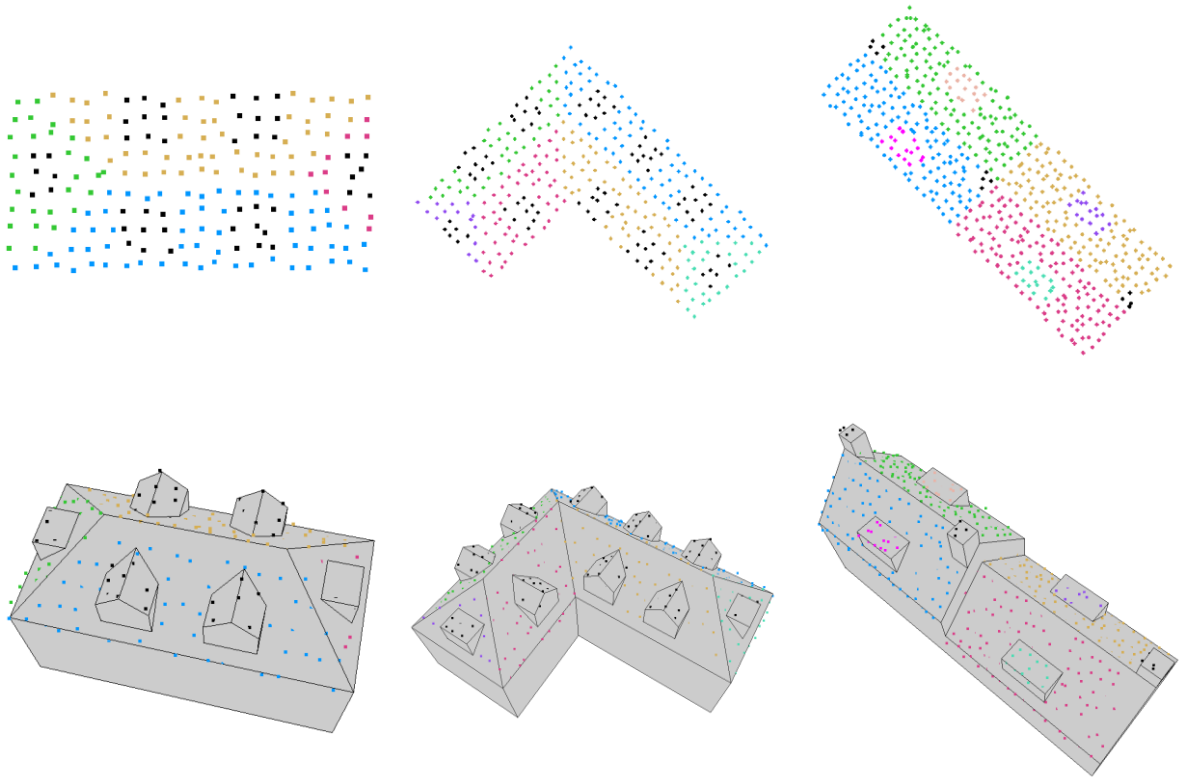


Figure 6.14. Some reconstructed buildings from artificial data. Top: Top view of the segmentation results (unsegmented points colored in black). Bottom: The resulting building models reconstructed as proposed in this section.

The presented approach, in particular, provides accurate models if many instances of the same roof superstructure type are present. In this case, also partially occluded roof superstructures can be reconstructed, as shown in Figure 6.15. These models are furthermore always topologically correct and reflect regularities that are usually present in man-made objects.

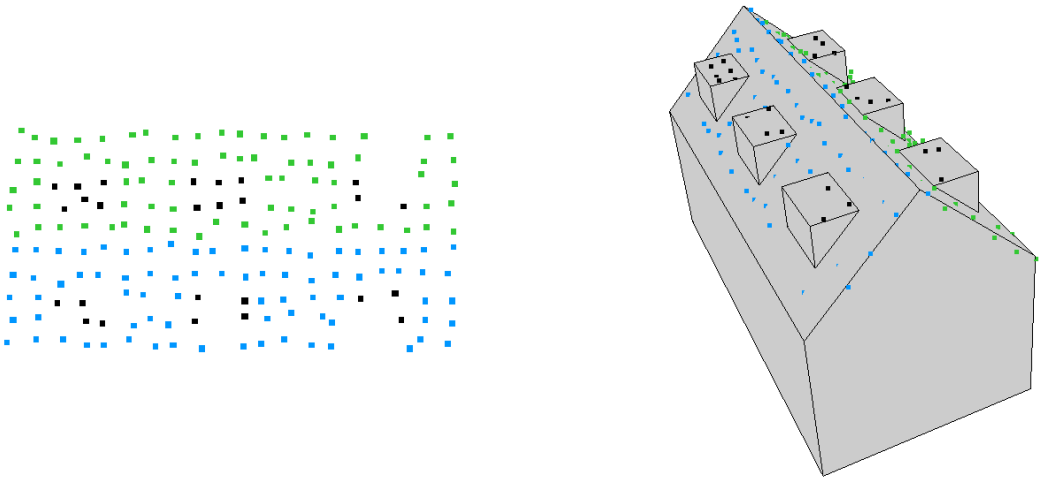


Figure 6.15. Reconstruction result of a building with partly occluded roof superstructures. Left: Top view of the segmentation result (unsegmented points colored in black). Right: The resulting building model reconstructed by the proposed approach.

However, special care has to be taken, so that a roof superstructure model is not placed in an area of point outliers. The size of a candidate area is often a quite sufficient indication but it cannot deal with an accumulation of outliers that might occur in reality.

6.2.1 Candidate Points Detection

A crucial element for the automatic reconstruction of roof superstructures is the detection of those points that belong to a roof superstructure. This task is challenging because it deals with a small number of points so that their surface shape is often not clearly recognizable during the reconstruction process of the base roof. Therefore, many problems might occur if the given point cloud is sparse and noisy. To improve the visualizing capabilities, the presented examples in this subsection are shown in a higher point density than in reality.

For the detection of appropriate candidate points, the proposed approach utilizes virtual points as they are defined in the sub-surface segmentation method presented in chapter 4. Here, not all virtual points are taken into account but only those that are located below unsegmented surface points or that are associated with an already recognized roof superstructure. Since these virtual points are always located below measured surface points, they can be used as an indicator for areas where roof superstructures are present in the x-y plane. Some examples are given in Figure 6.16.

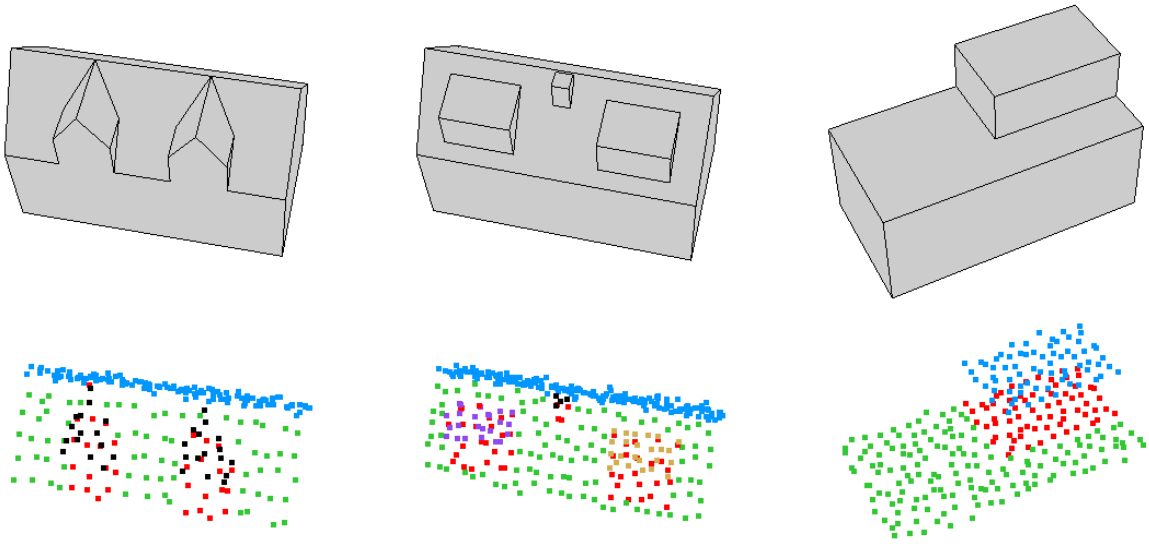


Figure 6.16. Segmented surface points enriched with virtual points (red). Unsegmented surface points are colored in black.

However, virtual points as defined before cannot be used alone as a reliable criterion for the presence of roof superstructures as shown in Figure 6.17. In fact, their related surface points can be considered as a superset of points that are located on a roof superstructure. Therefore, a selection of these points is mandatory.

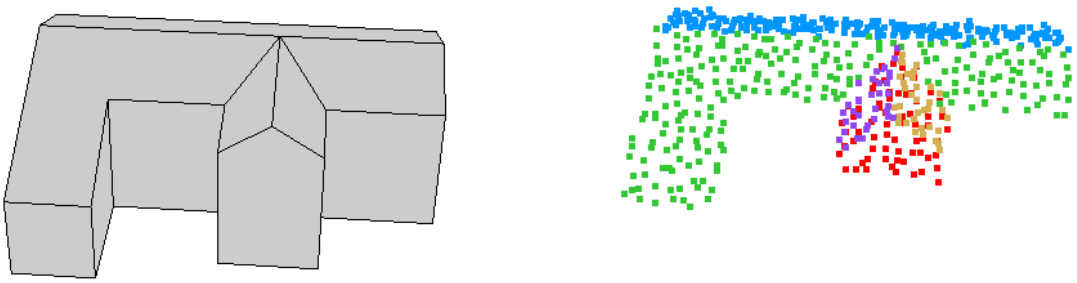


Figure 6.17. The result of sub-surface segmentation. Virtual points are colored in red.

For this purpose, a CCA is carried out to separate the virtual points into different sets so that neighboring virtual points belong to the same point set. The number of point sets can be considered as the maximum number of roof superstructures on a base roof. Afterward, a minimal bounding rectangle is calculated in the x-y plane for each point set. It represents the approximate area and location of the potential roof superstructure, wherefore the term candidate area is used instead in the following. Point sets with a small candidate area are treated as outliers and discarded at this point.

In order to verify whether a candidate area of a point set comprises a roof superstructure, the candidate area is slightly enlarged inversely to the point density of the given input data. Then, all those surface points are selected that are located within the enlarged candidate area. Based on the selected surface points, another minimal bounding rectangle is calculated in the x-y

plane and slightly enlarged. A candidate area is only considered as the location of a roof superstructure if the enlarged bounding rectangle of the surface points encloses all the virtual points of its point set. All other candidate areas and their point sets are discarded. Two examples of the selection workflow are given in Figure 6.18.

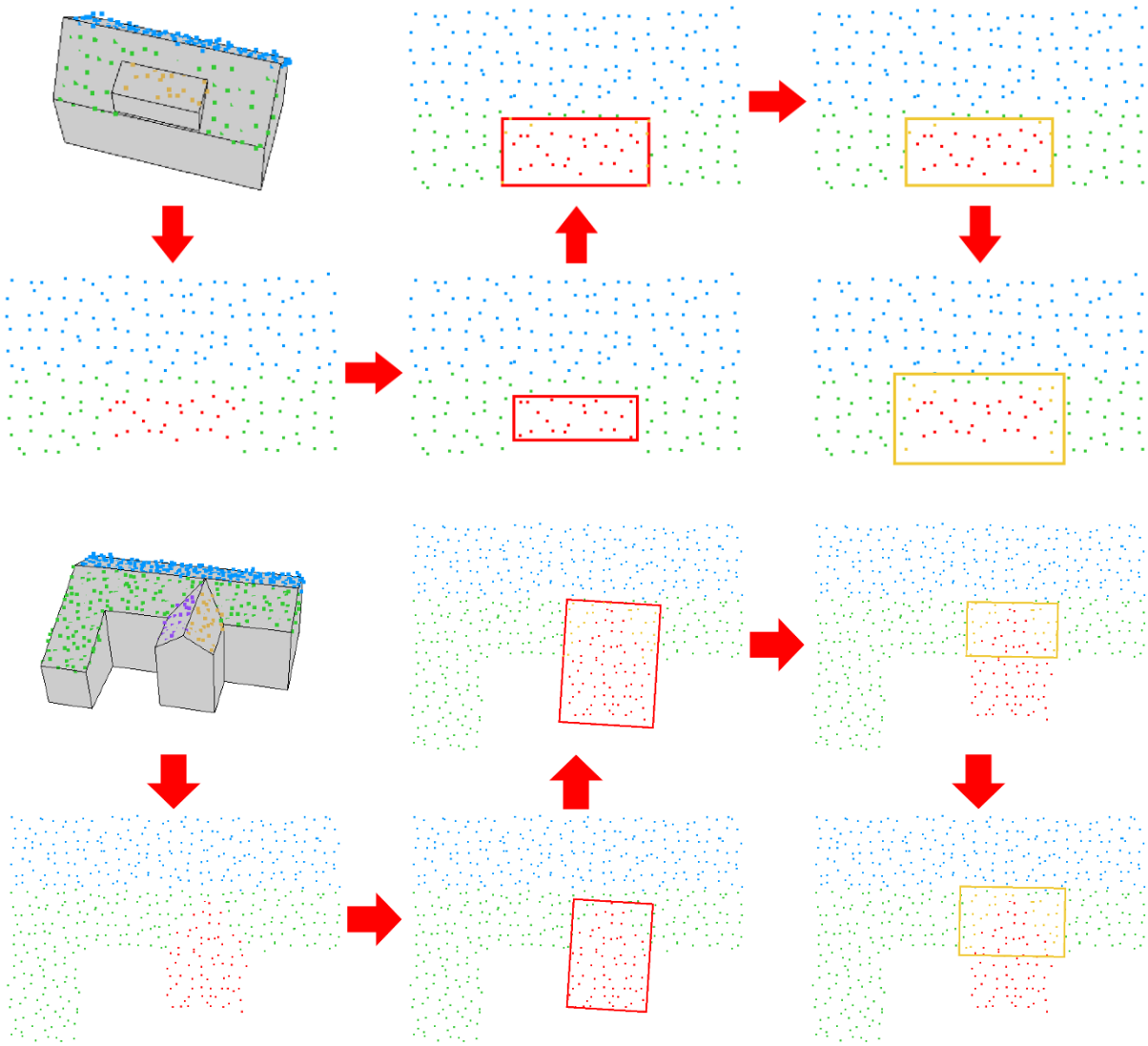


Figure 6.18. Two examples for the selection process of roof superstructure points. From left to right: building overlaid with surface points and the CCA result; bounding rectangles (virtual points); bounding rectangles (surrounding surface points).

In the first example, all virtual points are enclosed by the enlarged minimal bounding rectangle of those surface points that are located within the enlarged minimal bounding rectangle of the virtual points. Therefore, the candidate area of this point set is considered as a location of a superstructure. In contrast, the point set in the second example is discarded because the enlarged bounding rectangle of the selected surface points does not enclose all virtual points in the candidate area.

6.2.2 Grouping of Similar Roof Superstructures

Once all appropriate point sets are detected, all those sets are assigned to the same group that belong to the same type of roof superstructure. For this task, typically keypoints, which are also referred to as interest points, are extracted for each point set. In combination with a local feature descriptor at each keypoint, correspondences between two point sets can be detected. In computer vision, one of the best-known algorithm for this purpose is the SIFT (Scale-Invariant Feature Transform) method (Lowe, 1999). For 3D point clouds, there have also been several methods proposed such as intrinsic shape signatures (ISS) (Zhong, 2009) and normal aligned radial feature (NARF) (Steder et al., 2011). However, due to the low point density and the small size of the candidate areas, an automatic extraction of these keypoints is usually not feasible to complete this task. Also, shape fitting algorithms as commonly used in model-driven reconstruction approaches often fail at automatic extraction.

Therefore, the presented approach utilizes the ICP principles as explained in section 3.5 to increase the point density for each roof superstructure type. For this purpose, all remaining virtual points from the previous subsection are replaced by their related surface points and those candidate areas that feature a similar size are initially assigned to the same group. To start the ICP iterations for a group, an initial rough transformation in form of a rigid transformation is carried out for each of its point sets. The translations ensure that the centroids of the candidate areas of a group meet at one point. Furthermore, the rotations ensure that the x-y normal vectors of the base roof segments, on which the point sets of a group are located, point afterward in the same direction. Thereby certain angles are preferred for two point sets of the same group if they are located on the same base roof; e.g., 0 degree if both point sets belong to the same segment and 180 degree if the point sets belong to opposite segments. Additionally, the direction of the base roof is taken into account if two point sets are not located on the same base roof. An example of the initial rough transformation process is shown in Figure 6.19.

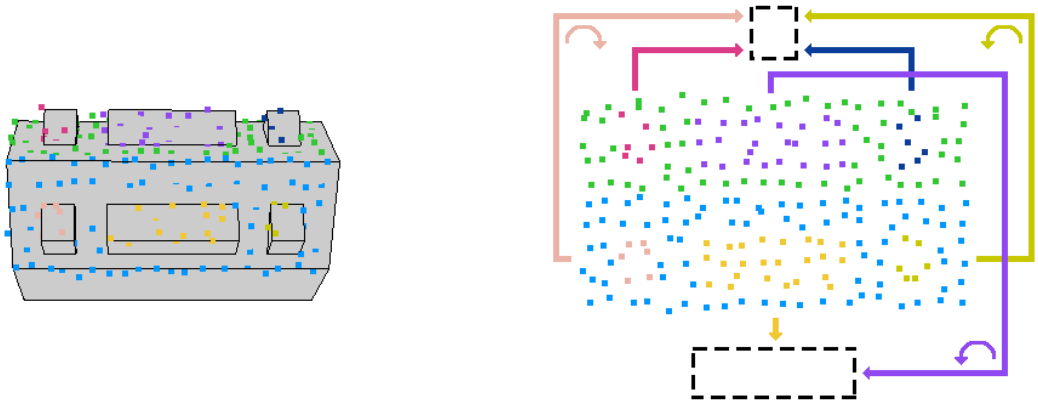


Figure 6.19. An example of the initial grouping and the initial rough transformation process.

The fine registration process starts thereafter in each group to estimate the point correspondences of the point set with the highest and the second highest number of surface points. Then, a transformation that minimizes the MSE of the point correspondences is calculated and applied. But instead of a rigid transformation, the method restricts ICP to use only translations. Rotations are not considered during the fine registration process. The objective function to be minimized in each iteration for two given point sets $A = \{a_i\}$ and $B = \{b_j\}$ with $i = \{1, \dots, n\}$ and $j = \{1, \dots, m\}$ is defined as

$$f(t) = \frac{1}{n}\sum_{i=1}^{n}\left\|a_i - b_j - t\right\|^2 \rightarrow min \tag{6.3}$$

where b_j is the closest point in B to the point $a_i \in A$ and t the translation vector. Both point sets are merged if the final MSE value is lower than a predefined threshold and the fine registration process starts again. Otherwise the group is split and each of the point set is assigned to one group before starting the process again. In the latter case, the remaining point sets of the original group have to be tested for both groups but only the registration result with the lower MSE value is taken into account. Once all point sets of a group are merged, the final groups whose point sets originated from the same group are registered and merged if the MSE value is lower than a predefined value. A reduced threshold is applied if the merging of the group supports the symmetric distribution of the original point sets on the base roof. An example is given in Figure 6.20.

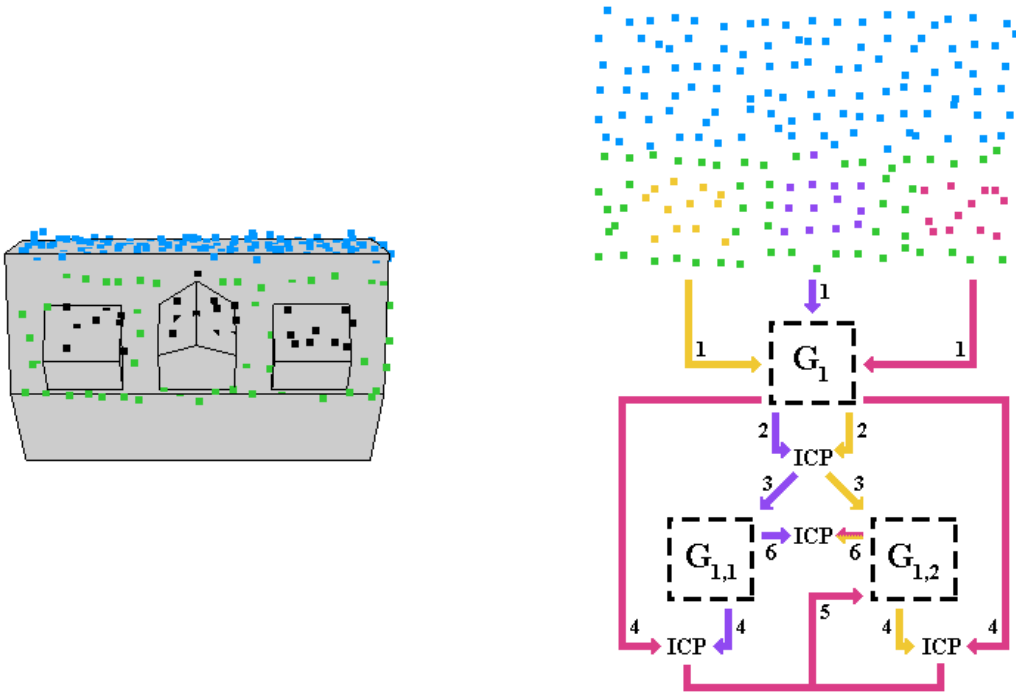


Figure 6.20. Example workflow of the fine registration process.

First, all point sets in the given example are assigned to the same group because their candidate areas feature a similar size. Then, the surface points of the yellow and the purple point sets are tested if they belong to the same roof superstructure type. Due to their final MSE value, they are separated into different groups. Therefore, the remaining pink point set has to be tested for both groups and is finally assigned to the group of the yellow point set. This results in the final grouping because the registration of both groups results in a high MSE value.

6.2.3 Roof Superstructure Modeling and Construction

The resulting groups of the previous subsection are considered during the subsequent modeling process as different roof superstructure types and the number of merged point sets in a group as the number of instances of it. Instead of reconstructing the superstructure of each original point set independently, all instances of a group are constructed at once. Caused by the previous merging process, a higher point density than in the input data is already achieved if more than one roof superstructure instance has been detected. Furthermore, in this case, single gaps in the surface points of an instance, which might occur due to partially occluded areas, are automatically closed by the surface points of other instances.

Therefore, several features like ridge lines can now be directly extracted in each group with less strict thresholds than in chapter 5. Considering the extracted features, points of planar regions are segmented and planes estimated. An adjustment of these planes is carried out to emphasize on the one side rectangularity, if the direction of the normal vector projected onto the x-y plane is close to a certain angle, and on the other side symmetries, which are for example often present in the rooftop slope of a saddleback roof superstructure. Afterward, half-space modeling as described in section 6.1 is carried out for the construction of the roof superstructure model. Then, instances of the so far defined model are created and placed in each candidate area of the original point set of the group. The location of an instance is defined by the inverse rigid transformation of the previous subsection, which consists of the initial rough transformation and the translations during the iterations. In conjunction with the surface of the base roof, the final roof superstructure model is created. Special care has to be taken so that the roof superstructure models do not overlap with each other. In such a case, those half-spaces that represent the vertical facades of the overlapping models are translated accordingly.

6.3 Building Adjustment

As man-made objects, most buildings have strong regularities in their shape and structure. A discussion on the regularity of building structures is, for example, given in (Xiong, 2014). Incorporated in the reconstruction process, this can significantly increase the quality of building models. In practice, its impact on the quality of a model is particularly noticeable if a purely data-driven reconstruction approach is performed that does not incorporate any common regularities. This method often leads to abnormal structures, so that a further adjustment of the building models becomes crucial. Some typical problems with automati-

cally reconstructed building models are shown in Figure 6.21. Here, for example, roof surfaces do not align well with each other, especially if they are comprised of a small number of points. Furthermore, adjacent building components are sometimes modeled with abnormal overlaps. Another frequently occurring issue is that components of buildings and their coplanar facades are often modeled with a small gap between each other resulting in unwanted intrusions and extrusions.

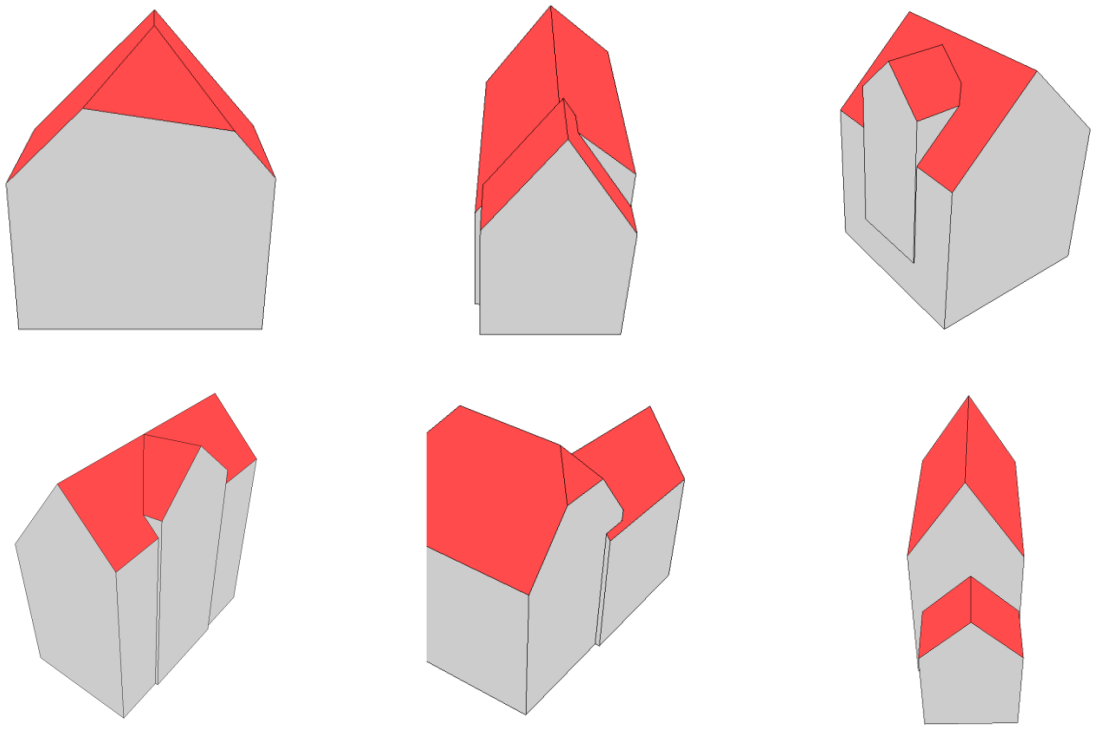


Figure 6.21. Typical problems in the automatic reconstruction of building models from point clouds.

To avoid abnormal structures in building models, some data-driven methods take building characteristics into account during the reconstruction process to reflect building regularities. These, for example, can include the main orientation of a building in order to emphasize parallel and orthogonal structures (Vosselman, 1999; Alharthy and Bethel, 2004). The reconstruction process is thus enhanced by incorporating building knowledge to improve the shape of the building models while still preserving the flexibility of a data-driven approach. However, building regularities are comprised of more than just parallelism and orthogonality, including, but not limited to, co-planarity and repetitive structures. When considering all types of building regularities, a large number of complex constraints guiding the reconstruction process need to be explicitly defined. Thus, their implementation is at risk of becoming protracted and cumbersome. On the other hand, building regularities are automatically ensured in model-driven approaches due to the use of parameterized building templates. Consequently, the resulting models are usually well-formed. The capabilities of a model-driven approach become especially apparent when considering noisy point clouds and segments with a small number of points. However, more complex shapes are usually not defined by a single building template in order to keep the number of templates at a minimum.

Therefore, they are composed of simpler roof shapes on which regularities are individually ensured. Further regularities are, then, only considered between adjacent roof shapes. In this situation, adjustment methods for building models are only performed in a local context. Recent methods, however, have shown that global regularities can also significantly improve the modeling quality in terms of both fitting accuracy and human vision judgment (Zhou and Neumann, 2012).

Thus far, a reconstruction method has been presented where adjustment rules are individually defined for each building feature of the multi-scale knowledge graph in order to emphasize innate regularities in the final building models. For instance, the ridge feature enforces that the ridge-forming roof surfaces, in the final building model, always point in the opposite x-y direction. The resulting building models are therefore, in many cases, aesthetically pleasing. However, the use of purely feature-driven building adjustments to support building regularities has its drawbacks, namely:

- There are many different kinds of features in a building model and each of them requires a well-defined set of adjustment rules.
- Adjustments are performed on every feature individually so that one adjustment might result in subsequent adjustments.
- Adjustments are only performed locally to retain a small number of features while building regularities on the global context are not considered.
- In order to consider global regularities, a large number of dependencies and interdependencies, usually of high complexity, between features need to be considered. Consequently, an easy adaptation of the applied feature set, which confirms to the reconstruction area, is no longer realizable because existing adjustment rules need to be adapted as new features are added or removed.

In conclusion, the feature-driven adjustment is limited to local regularities and requires a large number of complex rules that make building construction laborious and inflexible. In order to overcome these issues, an enhanced 3D building adjustment process is presented in this section, which reduces the number of feature adjustment rules to a minimum. For this, 3D building adjustment rules are defined for a planar half-space representation of a building rather than for individual building features. Thus, once all half-spaces are defined by the recognized features, the adjustment rules can be directly applied. In this context, the adjustment of half-spaces will, from this point forward, be referred to the adjustment of their hyperplanes. The new half-space adjustment rules utilize planar half-space regularities and take into account advanced knowledge of buildings, among others, their local and global building regularities including symmetry, co-planarity, parallelism and orthogonality, thus emphasizing common building regularities in the final building models. This, in turn, improves in most cases the accuracy and shape of the resulting building models.

To ensure that those half-spaces which directly contribute to a building feature have a higher impact on its adjustment than half-spaces that are introduced by other building features, a distinction is made between local and global adjustments. Local adjustments mainly concern

regularities that are within single building components. The details of the local adjustment procedure are explained in subsection 6.3.1. Global adjustments, on the other hand, ensure that building parts are combined accordingly so that they fit well together and do not cause any unintentional ex- or intrusions and overlaps. Here, principles from the local adjustment step are adapted for the use in a global context. Further details of global adjustments are described in subsection 6.3.2. Both, local and global regularization rules employed in this section, are based on methods presented in (Thrun and Wegbreit, 2005) and (Li et al., 2011) and are modified for building reconstruction purposes. Common issues, as shown in Figure 6.21, frequently occur if, for example, half-spaces are directly obtained from segments without adjustment, and are easily solved by applying the proposed half-space adjustment rules on the final building models.

Compared to the feature-driven adjustment process, the proposed half-space adjustment method has the advantages that more than one building feature can be adjusted simultaneously, while special consideration of dependencies between different features is not needed. Furthermore, global regularities are incorporated and the set of features remains flexible. Therefore, when integrated into the fully automatic building reconstruction framework, the half-space-driven adjustment process is also well suited for the automatic reconstruction of large urban areas. A reconstructed segment of the Vaihingen data set is shown in Figure 6.22 as an example, which integrates the half-space adjustment procedure, emphasizing common regularities in the reconstructed building models so that they are aesthetically pleasing and do not possess abnormal structures.

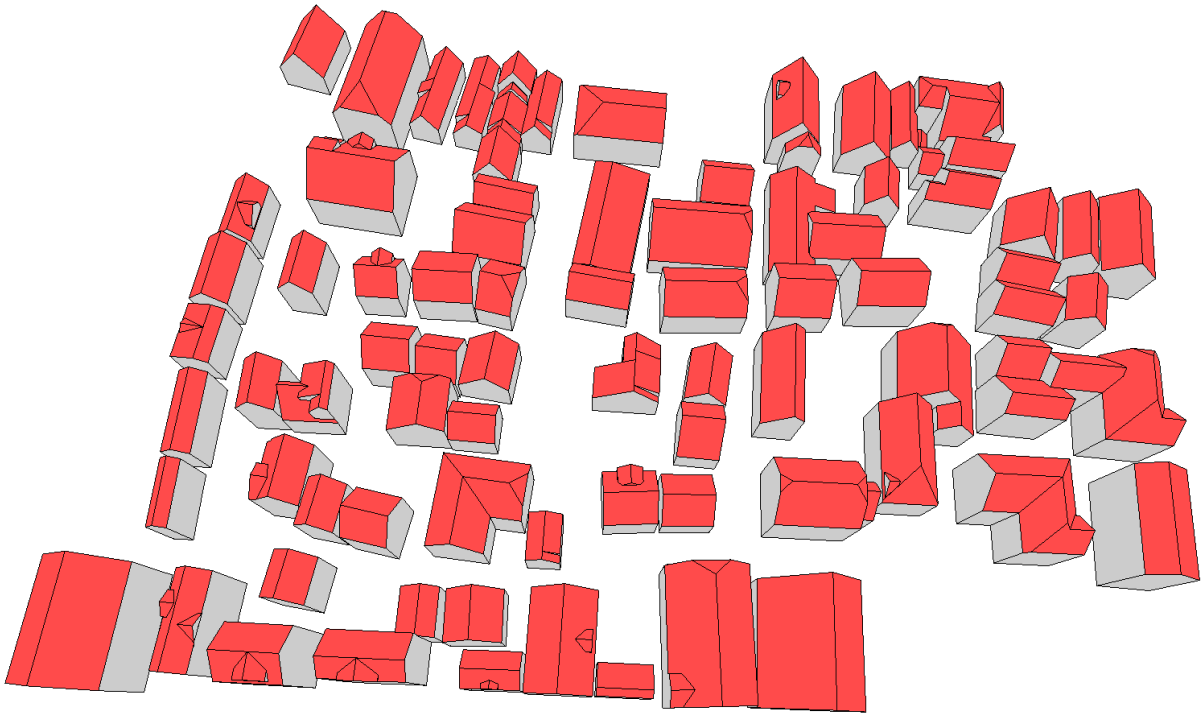
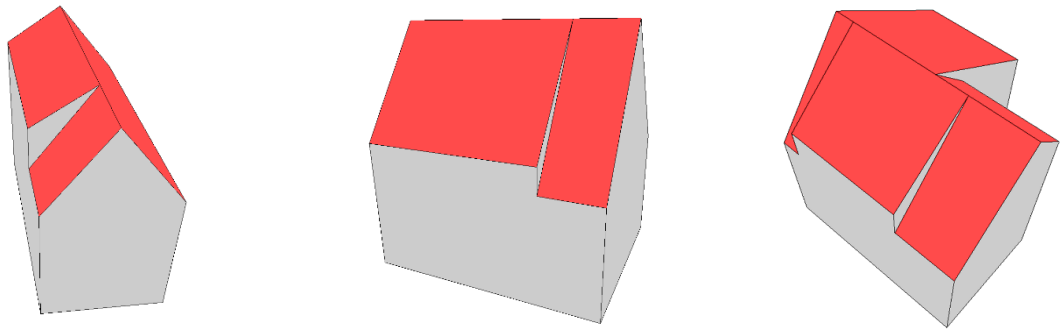


Figure 6.22. Building models reconstructed in accordance with the presented local and global adjustment rules.

A closer look at some adjusted building models reveals that buildings consisting of several components are reconstructed without unwanted in- or extrusions but feature coplanar facades as shown in Figure 6.23. In the resulting models, when only one facade is expected to be present, adjacent building components share that facade. Also, wherever possible, different components are modeled in such a way that they share common building features like a ridge, eave, and gable. It is also interesting to note that, due to global regularities, the first and the last roof surface on the left side of the building in Figure 6.23c have the same slope and eave height.



(a) (b) (c)

Figure 6.23. Reconstructed buildings consisting of adjacent gable roofs and wall dormers.

A difficult challenge for a fully automatic building reconstruction approach is the reconstruction of dormers and other small roof superstructures that belong to other building components. Many reconstruction approaches do not consider the interrelationships between such small components. They reconstruct each component separately and individually whereas the presented half-space adjustment procedure takes these interrelationships into account. Figure 6.24 shows some building models with different roof dormer types obtained as a result of the method presented. From base roof and dormer surfaces as well as between dormer surfaces, global adjustment leads, accordingly, to parallel or coplanar structures. Additionally, because their ridge forming segments are enforced by local position adjustment to indicate an opposition in the x-y direction and global adjustment ensures orthogonality, ridge lines become horizontal and orthogonal to each other. One further result of the combination of the local and global half-space adjustment is that dormer hip ends and the roof plane on which they lay likely share the same x-y orientation.

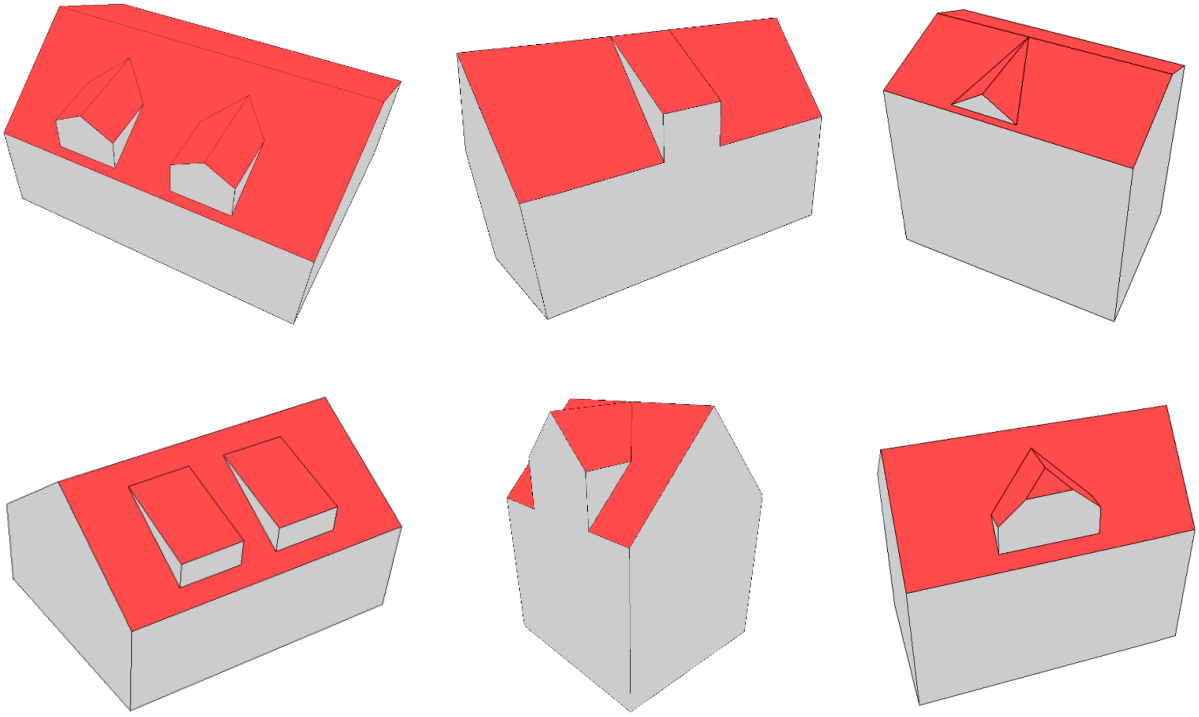


Figure 6.24. Reconstructed buildings with different roof dormer types (gabled, eyebrow, partial hipped, and shed).

6.3.1 Local Adjustments

In this subsection, the central concept for the local adjustment of half-spaces is presented. A half-space adjustment is considered local if the adjustment is only performed on the half-spaces that define a building component, not on other half-spaces. In local half-space adjustment, it is observed that man-made objects, such as buildings, often possess components that are symmetric and regular (Rosen, 1975). To support the occurrence of symmetries and regularities, the half-space adjustment of a building component roughly consists of the following steps:

1) Slope adjustment: a group of half-spaces, whose hyperplanes feature similar slopes, are adjusted to their average value.
2) Orientation adjustment: a group of half-spaces, whose hyperplanes feature similar x-y directions, are adjusted to their average angular value. Here, orientation is regarded as the 2D rotation around the z-axis.
3) Position adjustment: half-spaces with vertical hyperplanes are shifted to further improve symmetries and regularities.

Slope and orientation adjustment are accomplished by similar procedures and are therefore jointly explained. The details of position adjustment follow thereafter. Note that, in this context, the phrase “adjustment of half-spaces” is always in reference to the adjustment of the hyperplanes that define the half-spaces.

The impact of the entire local half-space adjustment procedure is illustrated on a half-hip building in Figure 6.25. The roof surfaces are oriented as strictly orthogonal or opposite to each other, while opposite surfaces have the same slope. The ridge and eave lines are horizontal, where the latter feature the same height on both sides, so there is symmetry with respect to a vertical plane that passes through the ridge.

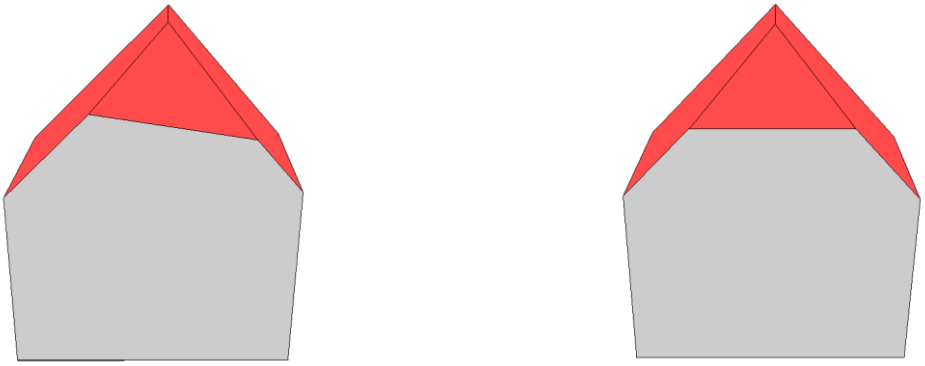


Figure 6.25. Reconstructed building model before (left) and after the local half-space adjustment step (right).

Slope and Orientation Adjustment

A substantial aspect of slope and orientation adjustment is the clustering of half-spaces. The scheme for this procedure is presented in Figure 6.26 and consists of the following four sub-steps: (i) calculation of clustering criterion and sorting of half-spaces accordingly, (ii) cluster determination, (iii) weighted averaging, and (iv) rotation of half-spaces.

The slope and orientation adjustment process each start in the local half-space adjustment step with the sorting of all half-spaces according to the respective clustering criterion. For this, a value for the slope and orientation adjustment is calculated for each half-space based on Equations (6.4) and (6.5) respectively, where h is a given half-space whose hyperplane is used to calculate its slope and orientation. To support the occurrence of symmetry, the absolute values of half-space slopes are used as clustering criterion for the slope adjustment. Thus, for example, it is taken into account that the two rooftop half-spaces of a gable roof often have the same absolute slope value but face in opposite directions. And to support parallel and orthogonal structures, the orientation angle modulo $\pi/2$ is used as clustering criterion for the orientation adjustment. The orientation angle is defined as the counterclock-wise angle between the x-axis and the normal vector of the half-space that is projected onto the x-y plane.

$$cc(h) = |slope(h)| \tag{6.4}$$

$$cc(h) = orientation(h) \ \% \ (\pi \ / \ 2) \tag{6.5}$$

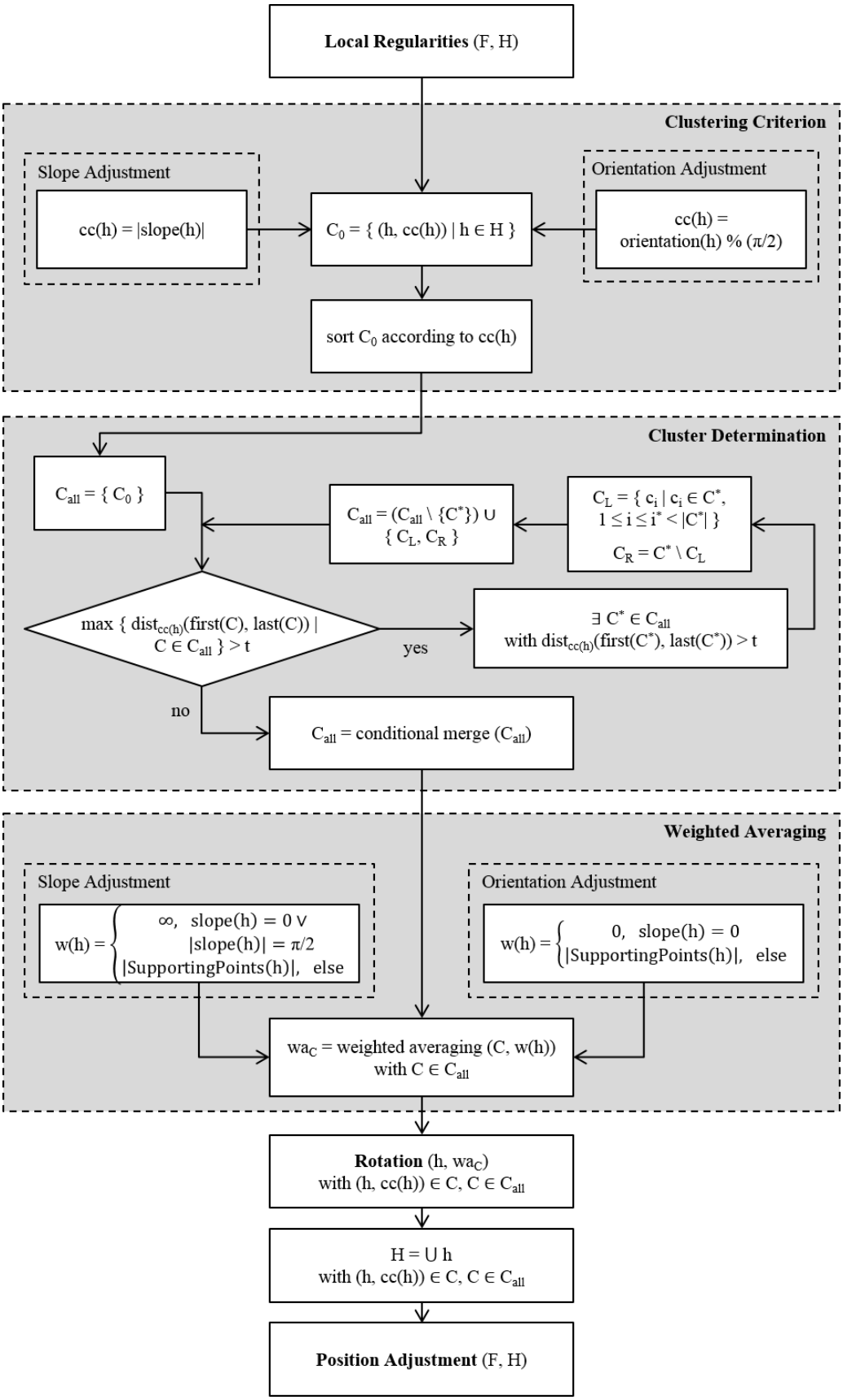


Figure 6.26. Overview of the local half-space adjustment process (F = set of features, H = set of half-spaces, C = cluster of half-spaces, t = threshold for cluster determination).

For the subsequent clustering of half-spaces, there are already several general clustering techniques as shown in section 3.4. For the presented building adjustment, a divisive method is implemented that starts with one sorted cluster, including all half-spaces. If the distance between the first and the last element of a cluster is greater than a predefined threshold, then the cluster is split into the two adjacent elements that have the greatest distance to each other. The algorithm is recursively applied to the two elements until no further split is necessary – this is when the distance between the first and the last element of each cluster is not greater than a predefined threshold. Note, a cluster may only contain a single half-space. In order to reduce the number of clusters caused by noisy data, a conditional merge is performed on the clusters obtained as such. To accomplish this, every isolated cluster is merged with the nearest, adjacent, non-isolated cluster and its number of supporting points is set to zero. A cluster is considered isolated if it only contains one half-space estimated by points, whose average distance to the plane defining half-space is greater than a predefined threshold.

After clusters are determined in this way, the calculation of the weighted average value for each is performed by taking into account the number of points that support a half-space (i.e. those that are close to the hyperplane of the half-space). In order to maintain the slopes of vertical and horizontal half-spaces in the slope adjustment step, an infinite weight is assigned to these half-spaces. This has the effect that half-spaces with a sloped hyperplane are adjusted towards the vertical and horizontal half-spaces in the next sub-step and not vice versa. The overall weighting function for the slope adjustment of half-spaces is presented in Equation (6.6). In contrast, a zero weight is assigned in the orientation adjustment to all vertical half-spaces as shown in Equation (6.7). This is justified by the fact that vertical segments have been discarded during the segmentation process so that they do not have any supporting points from the input data.

$$w(h) = \begin{cases} \infty, \; slope(h) = 0 \vee |slope(h)| = \pi/2 \\ |SupportingPoints(h)|, \; else \end{cases} \tag{6.6}$$

$$w(h) = \begin{cases} 0, \; slope(h) = 0 \\ |SupportingPoints(h)|, \; else \end{cases} \tag{6.7}$$

Because the half-spaces of the slope adjustment are sorted according to the absolute values of their slope, there are, for every cluster, two possible slopes, differing only in their sign. Similarly, because of the use of modulo π/2 in the orientation adjustment, there are, for every cluster, four possible orientation values. In the subsequent rotation of the half-spaces, the most probable value of the slope, respectively orientation, is chosen.

For the rotation of a half-space during the slope and the orientation adjustment, a rotation axis is defined for each step. For the slope adjustment, a horizontal line is chosen whose direction is orthogonal to the normal vector of the half-space. And, for the orientation adjustment, a vertical line is selected. These two lines are chosen so that they have an intersection point that satisfies the following conditions:

- For any non-vertical half-space, the intersection point of the two lines is the center point of the segment, which originally defines the half-space. Therefore, this point remains unaltered by the local half-space adjustment.
- For any vertical half-space, the intersection point of the two lines is the center point of the feature. For example, the intersection point of the two lines for a shed roof is the intersection point of the two lines of the rooftop half-space, and for a gable roof it is the center point of its ridge.

The horizontal and vertical lines chosen this way are the rotation axis in the last sub-step.

Position Adjustment

During position adjustment, vertical half-spaces are translated along their normal vector to improve symmetries and regularities. It is further divided into two sub-steps. Firstly, all vertical half-spaces are sorted by their clustering criterion, which is the shortest distance to the feature (e.g., a ridge line) that originally defines the geometry. Clusters are, then, determined and a weighted average distance is calculated for each cluster. Thereafter, all half-spaces are translated along their normal vector so that the shortest distance to the feature corresponds to the weighted average distance of the cluster to which they belong. If more than one position is possible, the position nearest to the original is used.

Secondly, all pairs of adjacent half-spaces that consist of a vertical and a non-vertical half-space intersecting in a horizontal line (e.g., an eave) are considered. Two half-spaces are considered adjacent if a sufficient number of their points supports the intersection (i.e. those that are close to the intersection line). These half-space pairs are clustered according to the height of their horizontal intersection lines. Now, in order to reduce the number of different intersection heights, the vertical half-spaces are translated in a cluster along their direction within a strict, predefined threshold.

The advantages of implementing these two sub-steps of the position adjustment are as follows:

- The first sub-step ensures that, for example, eaves can be assigned the same height in the second sub-step even if the angle between the two half-spaces of the eave is close to 0.
- The threshold in the second sub-step prevents the translation of a vertical half-space that is far from its original position if the angle between the two adjacent half-spaces of a pair is close to $\pi/2$.

Other approaches often directly adjust eave heights without considering the context of how they were generated. The effects include that heights of eaves, which are actually the same, are not adjusted in the model and that features might get severely shifted.

6.3.2 Global Adjustments

As described in the previous subsection, local half-space adjustment is performed for each building component individually and independently from other components. This allows the use of rules with soft thresholds to support local regularities. In global half-space adjustment, attention is given to global regularities and symmetries, which are usually also present in man-made objects (Kazhdan et al., 2004). For example, two adjacent gable roofs can share the same ridge line as shown in Figure 6.27. Therefore, the aim of global half-space adjustment is to adjust the half-spaces of building components in order to support the occurrence of global regularities and symmetries between them while preferably also maintaining local regularities and symmetries within each component. This step automatically eliminates global asymmetries caused by the previous local half-space adjustment step. The following four steps illustrate the implementation of the global half-space adjustment procedure:

1. Global slope adjustment is performed on the half-spaces of all building components analogous to the local slope adjustment, but with a more stringent threshold.
2. Global orientation adjustment is performed on the half-spaces of all building components analogous to the local orientation adjustment, but with a more stringent threshold.
3. Feature growing translates half-spaces of building components within a strict predefined threshold so that some building features of the same type are merged.
4. Translation of vertical half-spaces is done to support global coplanarity and symmetry of facades in a building model.

To translate vertical half-spaces, the set of all vertical half-spaces is initially determined for each x-y direction. Then, half-spaces of each set are sorted by their distance to a fixed point (e.g., the origin). Afterward, clusters are identified analogous to the divisive clustering method explained in the slope and orientation adjustment of the previous subsection. Finally, for each resulting cluster, a weighted distance average is calculated and each half-space in the cluster is translated along its normal vector according to the calculated average.

Figure 6.27 illustrates a reconstructed 3D model, where the local half-space adjustment of each component has produced several global asymmetries. For example, the half-spaces belonging to the top left of the gable roof have different normal vectors and distances to the origin that were originally the same. In this case, the first and second step of the global half-space adjustment restore the original number of differing normal vectors.

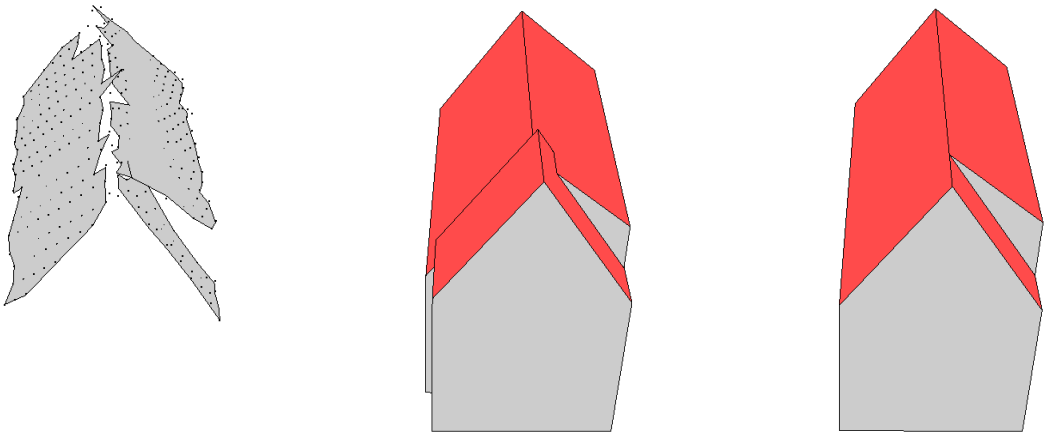


Figure 6.27. The segments of two adjacent gable roofs overlaid with surface points (left), the reconstructed building after local (middle) and after global half-space adjustment (right).

In addition, the second global adjustment step improves the incidence of parallel and orthogonal alignment between building components. As shown in Figure 6.28, the half-spaces of connected building components, such as L-, T- or cross gables, are automatically fitted to the half-spaces of the building components they connect.

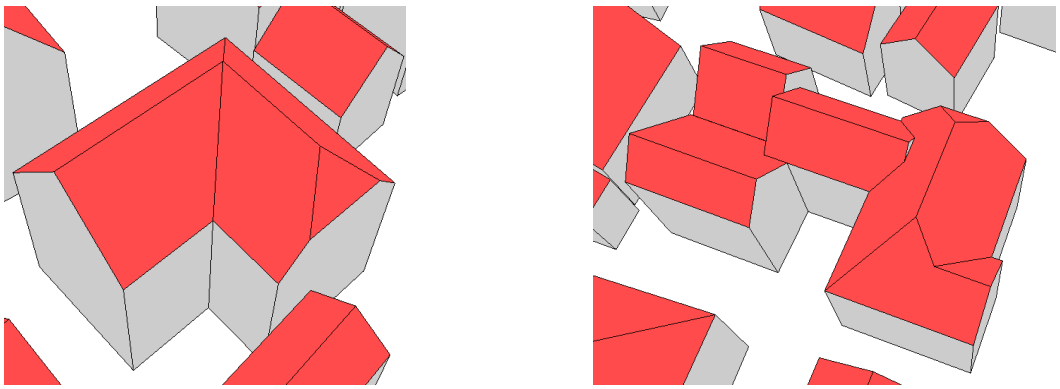
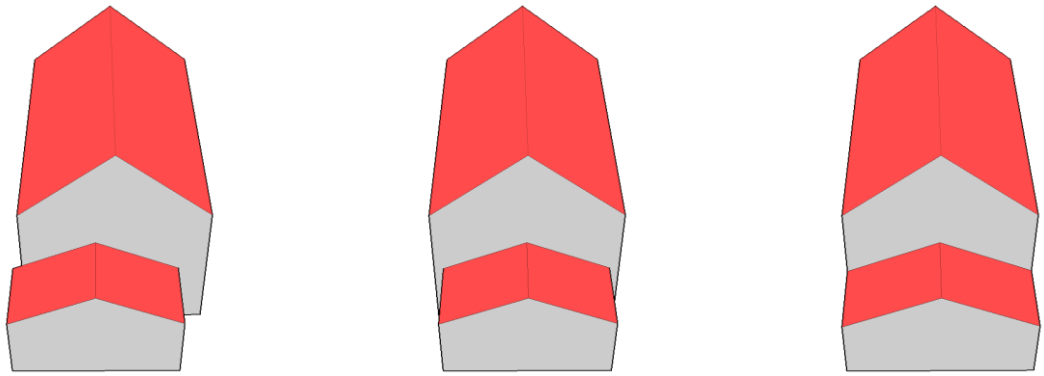


Figure 6.28. With global half-space adjustment reconstructed L-shaped gable roof (left) and a more complex building (right).

In the third step, the half-spaces of some building components are translated in all three directions within a strictly predefined threshold. Therefore, shown in Figure 6.27, the two adjacent gable roofs are translated so that they share the same ridge. Here, feature growing through other building components is allowed. Therefore, features of building components that are not adjacent, but connected to each other by a chain of pairwise adjacent building components, can be merged. For example, all components of a cross gable roof can be adjusted pairwise, even if the two main ridges have different heights. By restricting features to grow only through other building components, misalignments can be reduced.

In addition to growing the 3D feature, a 2D feature growing is performed in the x-y plane. This allows for the alignment of buildings, as shown in Figure 6.29a. Here, the ridge of each of the two building components is adjusted so that their projection on the x-y plane lie on the same line, as demonstrated in Figure 6.29b. Because 2D feature growing is being performed only in the x-y plane, all translations have to be parallel to the x-y plane.



(a) (b) (c)

Figure 6.29. The result of local half-space adjustment (left) and global half-space adjustment with performing 2D feature growing (right).

The fourth step of global half-space adjustment results in, among others, the following effects. It eliminates misalignments between adjacent building components so that the two adjacent gable roofs in Figure 6.29b share the same facade on each side as shown in Figure 6.29c. In addition, it eliminates undesired extrusions and intrusions in a building facade. These are especially pronounced if roof structure components, like dormers, are present, as shown in Figure 6.30.

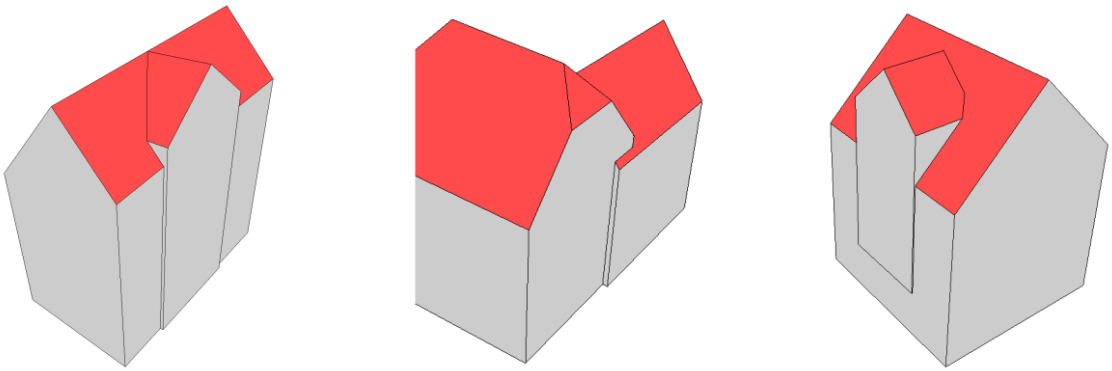


Figure 6.30. Extrusions in different building facades that can be automatically eliminated by global half-space adjustment.

7. Building Knowledge Integration Into a Data-Driven Reconstruction Method

As stated in chapter 2, approaches for 3D building reconstruction can be roughly divided into data-driven and model-driven approaches, depending on the degree of contextual knowledge they integrate about the general shape of buildings. Besides the assumption that almost all roofs consist of planar surfaces, pure data-driven approaches often do not integrate any other building knowledge at all. The shapes of the resulting 3D models are, therefore, not limited by any restrictions and they resemble very closely the input data. Thereby, not only areas with common roof structures can be constructed but also areas that include buildings with individual shapes. Without regularization, however, the building models or parts thereof can easily end up distorted and exhibit small irregularities if the constructed planar surfaces do not precisely meet in common points or lines. In contrast, model-driven approaches are more restricted towards the shapes that they are able to reconstruct. They often use a library of parameterized templates that can be combined to generate more complex shapes. The inherently strong regularization of shape templates is implicitly passed on to the reconstructed building models. Because not all buildings in the real world can be described by a finite set of shape templates, some buildings can only be crudely approximated by model-driven approaches.

Lately, data-driven and model-driven approaches have been merged towards hybrid reconstruction approaches that try to exploit the advantages of both worlds: the shape flexibility of a data-driven approach with the regularization capabilities of a model-driven approach. For this, they integrate prior knowledge of roof shapes with the intention to improve the regularization of the resulting models without lessening the flexibility to generate all real-world occurring roof shapes. In order to exemplarily demonstrate that the developed graph grammar in combination with the proposed multi-scale knowledge graph can also serve as a good basis for this purpose, a method has been developed to integrate building knowledge from the multi-scale knowledge graph into an existing data-driven reconstruction approach.

The automatic 3D building reconstruction approach presented in (Sohn et al., 2008) is in its core purely data-driven. It uses the concept of binary space partitioning to decompose the horizontal space according to the planar regions resulting from 3D point cloud segmentation. By assigning each resulting 2D region with its respective plane equation, a 3D building model is specified; see subsection 7.1 for specifics. The BSP based building reconstruction approach is generally suitable for the reconstruction of planar building roofs. It handles the missing data problem and is not limited to certain roof types. The resulting building models, however, do not follow any shape regularities and exhibit the same characteristics as models from data-driven reconstruction approaches. In order to improve their shape, a retrospective regularization of polygons that emerge from the BSP tree can be performed. But its imple-

mentation is not without difficulty because it has to deal with the 2D BSP subdivision itself and the plane definitions of the resulting partition regions to ensure topological correctness. This is aggravated by the use of hyperplanes during the binary subdivision that often splits planar roof regions into several parts that are stored in different subtrees of the BSP tree. In the proposed approach, building knowledge is, therefore, instead already incorporated during the construction of the BSP tree so that a regularized partitioning of the space is accomplished. The quality of a polyhedral building model obtained by such a partitioned space depends mainly on the quality of the hyperlines, the sequence in which the hyperlines are applied (partitioning score), and the merging conditions for adjacent partitions. These three aspects should thus particularly be taken into account in a BSP based reconstruction process. Therefore, the proposed method to integrate building knowledge into binary space partitioning specifically aims to

- improve the geometric accuracy of hyperlines,
- enhance the partitioning score,
- reduce the number of merge operations.

In order to achieve these objectives, an enhanced BSP-driven reconstruction approach has been developed that additionally takes building knowledge during the BSP construction into account. Analogous to data-driven approaches, it is not limited to a predefined set of shapes but their recognition supports the regularization of the partitioned space and thus the quality of the resulting models. Its general workflow is illustrated in Figure 7.1.

As proposed in (Sohn et al., 2008), the workflow starts with a height clustering in which a given set of building points is decomposed into clusters to reduce the shape complexity. The height clustering procedure is implemented as a connected component analysis based on the Delaunay triangulation. Thereby, it is ensured that the height discrepancy between a point and its neighboring points in each height cluster is less than a predefined threshold. Based on the partitioning, the 2D boundary is detected for each height cluster. In cases where only one point set can be obtained the boundary represents the initial building outline. For the subsequent segmentation of planar areas, sub-surface segmentation, as described in chapter 4, is performed for each height cluster. Thereby, smaller segment patches, which are usually disconnected due to superstructures, are implicitly merged to larger segments so that the subsequent extraction of building knowledge is improved. Based on the result of sub-surface segmentation, building knowledge is derived by applying the proposed GG in combination with the multi-scale knowledge graph as described in chapter 5. Afterward, the derived building knowledge is used for adjustment purposes. Here, those segments are first adjusted that are part of the extracted semantic information. After this all other segments which are not part of a recognized building feature are adjusted according to the properties of the extracted building knowledge. These two adjustment steps emphasize the natural structure in the geometry of a building so that a regularized space partitioning can be performed in the next step. The details of the adjustment step are further explained in section 7.2. After adjustments have been carried out, intersection and step lines are extracted. But instead of formulating

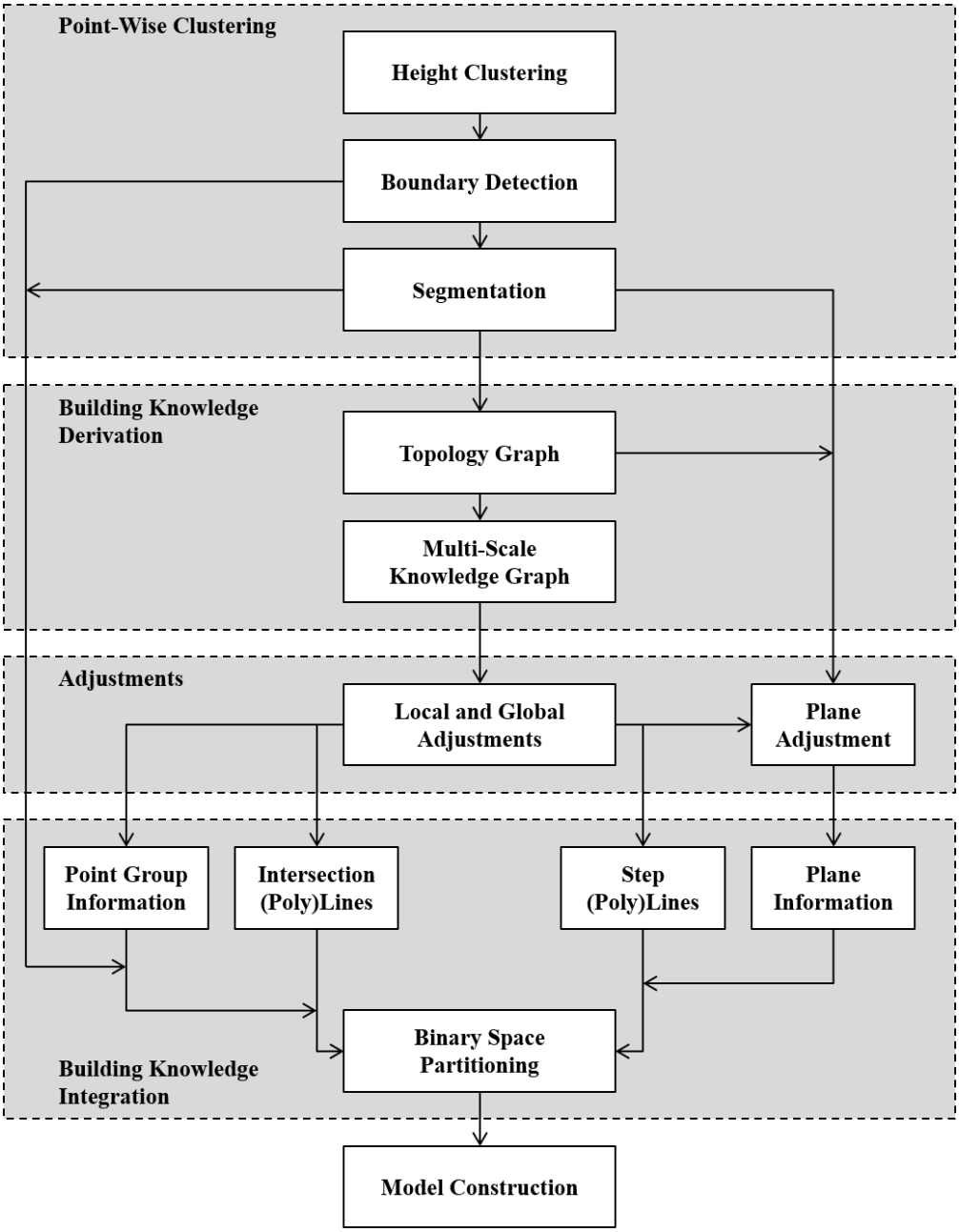


Figure 7.1. Overview of the extended BSP based reconstruction approach which integrates building knowledge for the construction of regularized models.

these lines directly to hyperlines and applying them to partition the space, the concept of hyperpolylines for the binary partitioning of the space in the context of building reconstruction is introduced. It reduces the number of partitions and supports the unambiguity during the merging process of adjacent partitions. For example, the number of partitions for a hip roof without any superstructures can be reduced from six to four so that no further merging is needed. Since the merging process is a possible failure source, the reduction of the partition number has a big impact especially on buildings with small superstructures such as dormers which usually cause a large number of small partitions. Also, concave point sets can be directly expressed by the use of valid hyperpolylines. This allows a more natural partitioning of the space if such a partition can be determined. The details of the line extraction and the

space partitioning are explained in subsection 7.3. In the last step of the reconstruction process a regularized 3D polyhedral building model is directly obtained from the BSP as explained in (Sohn et al., 2008).

The enhancement has been tested on several selected buildings of the Vaihingen test data set. In general, the approach presented in this section is suitable for the reconstruction of buildings with individual shapes. For complex buildings with many sub- and superstructures in particular, a big improvement in the resulting polyhedral building model is identified due to fewer partitions and the more regularized structure of the BSP. The quality of a resulting building model depends mainly on the building knowledge extraction. The more knowledge that can be derived from the input data, the more regularized the BSP and its resulting model will be. But the method is neither limited to the number of building features nor to the extractable building knowledge. Analogous to data-driven approaches, the amount of available building knowledge affects only the quality of the resulting model. Also, it handles the missing data problem, and always produces closed and regularized building models which can be directly obtained from the BSP. A further advantage of the integration of building knowledge into BSP is that the merging criteria for two adjacent partitions are not limited anymore to the information of their own points. For example, similar nonadjacent sub-parts of a building with the same semantic information can be taken into account. The final result of the BSP based reconstruction approach for the two segmented point clouds in Figure 7.6 and Figure 7.9 is shown in Figure 7.2 once separately in magnified form, and once as part of the reconstructed test area 1 of the Vaihingen data set. Note, in order to demonstrate the capabilities of the proposed extension for their reconstruction, only a minimum set of production rules has been used during the building knowledge derivation step.

An important advantage as well as limitation of the reconstruction method presented in the previous chapters is the recognition and incorporation of building features during the automatic reconstruction process. By taking into account the most common building features, abnormal geometric shapes in the final building models are avoided. Furthermore, certain details with regard to their semantic meaning can be switched off for further analyses. This advantage, however, automatically comes with the drawback that only those models can be created whose features are well-defined in the GG. Consequently, the set of production rules becomes quickly large in size if every building feature needs to be defined. Particularly in areas that include buildings with many individual shapes, the production rule set is at risk to become cumbersome although the presence of some building features might be only rarely detected. To keep the size of the production rule set low and to increase the flexibility of the presented automatic 3D building reconstruction framework so that also unknown roof shapes can be constructed, the enhanced BSP-driven reconstruction approach presented in this chapter can be applied afterward. For this, only a limited number of frequently occurring building features is defined in the GG. As a result, some planar segments are sometimes not meaningfully assigned to a building feature so that they are ignored during the reconstruction process. This results, as shown in Figure 7.3, in buildings that are still valid but not complete.

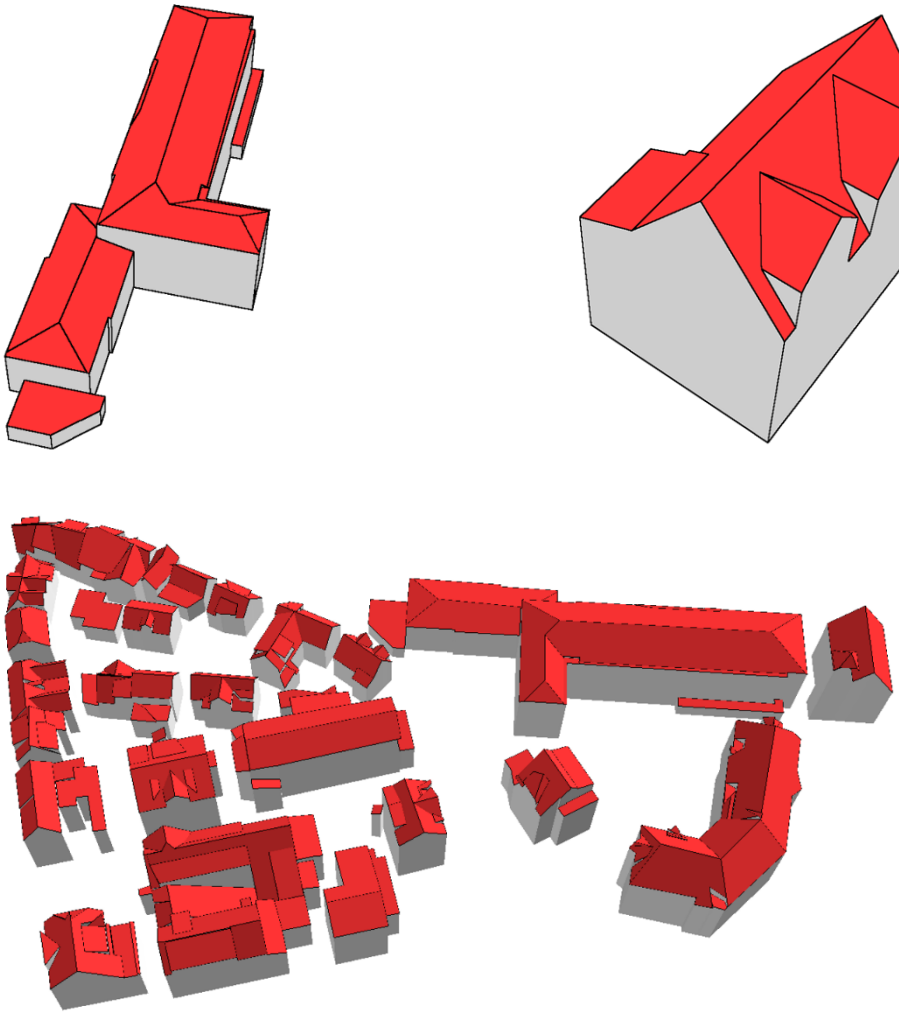


Figure 7.2. Resulting building models reconstructed by the BSP based reconstruction approach.

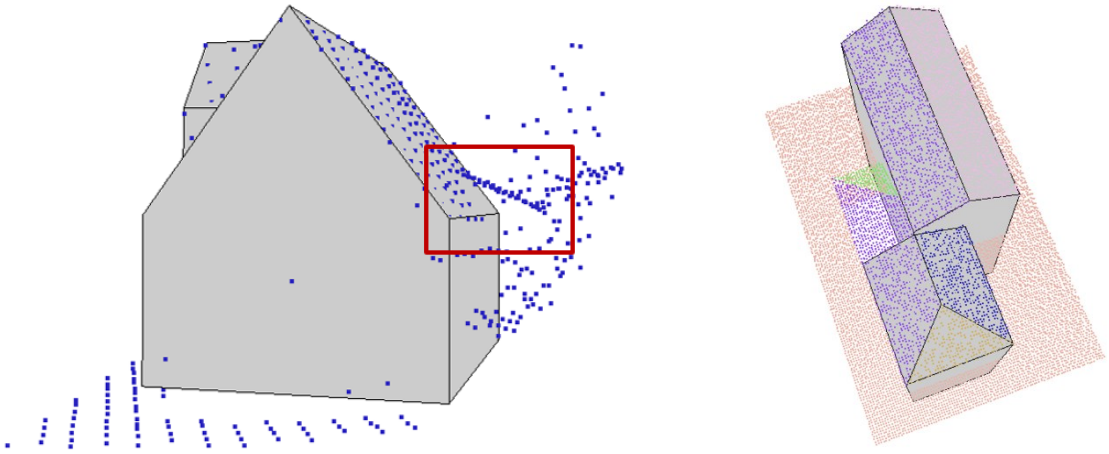


Figure 7.3. Two examples of unrecognized roof parts.

The presented enhanced BSP-driven reconstruction approach in this chapter can be then afterward applied to incorporate roof areas that were not reconstructed by the grammar based reconstruction approach. The focus is here in particular on larger roof areas whose shape do not frequently appear in the data set but that would be, due to their size, obviously missed in the resulting model. With the intention to improve the regularization of the resulting models without lessening the flexibility to generate all real-world occurring roof shapes, building knowledge of already reconstructed roof shapes is integrated in the partitioning process of the missing space as explained in this chapter. In this way, the number of production rules in the GG can be reduced and those parts of a building where only an insufficient number of building features were recognized can be reconstructed as well. Thus, the presented reconstruction approach can be considered as an optional extension to increase the geometric completeness of the resulting building models from the previous chapters.

7.1 Data-Driven Reconstruction Method

In the context of data-driven building reconstruction a BSP based approach has been introduced in (Sohn et al., 2008). It generates a polyhedral building model by applying the following three steps: point-wise clustering, building cue extraction, and BSP-driven rooftop topology construction. The main aspects of each step are roughly summarized in the next paragraph.

The first step starts with a height clustering of all previously identified building points. It decomposes the initial set of points into clusters to reduce the shape complexity. Thereby, each cluster has the property that the height discrepancy between a point and its neighboring points, as defined by the Delaunay triangulation, is less than a predefined threshold. Then a plane clustering algorithm is independently applied to each height cluster in order to estimate segments. In the second step, intersection lines and step lines are extracted. The intersection lines are obtained for each height cluster by calculating the intersections of all pairs of adjacent segments in a height cluster. The extraction of the step lines is based on the boundaries of adjacent segments and a CLF (Sohn et al., 2008). For the rooftop topology reconstruction in the third step a 2D binary space partitioning is performed. Therefore, the extracted step lines and intersection lines from the previous step are formulated as hyperlines. Due to the recursive nature of the BSP, the resulting space partitioning depends on the order in which the hyperlines are applied. For this reason a partitioning score, which takes into account the plane homogeneity, the geometric regularity, and the edge correspondence for each hyperline, is calculated in every recursion. Finally, a merging process is performed based on the BSP tree that merges all those adjacent partitions whose planar equations have similar normal vectors.

This method is able to produce polyhedral building models even in complex urban settings where buildings are comprised of a number of sub-shapes. Additionally, it handles the presence of the missing data problem. However, due to the data-driven nature of this approach the quality of such a polyhedral model depends mainly on the extraction quality of the intersection and step lines. Even the devised geometric regularization of the CLF, which quantizes line slopes in a limited number of angular ranges, cannot always avoid irregular and

sharp corners. Therefore, in (Sohn et al., 2012) a method is introduced to rectify errors in a polyhedral building model. It considers the resulting vectors from a BSP as noisy model boundaries and progressively rectifies them based on MDL. The presented method is, on the one hand, able to produce building models which consider certain 2D regularizations. However, on the other hand, the solidness of the models is not guaranteed anymore, which means that an x-y coordinate can have more than one z-value due to unintentional gaps between adjacent roof planes. Some typical examples of this phenomenon are shown in Figure 7.4.

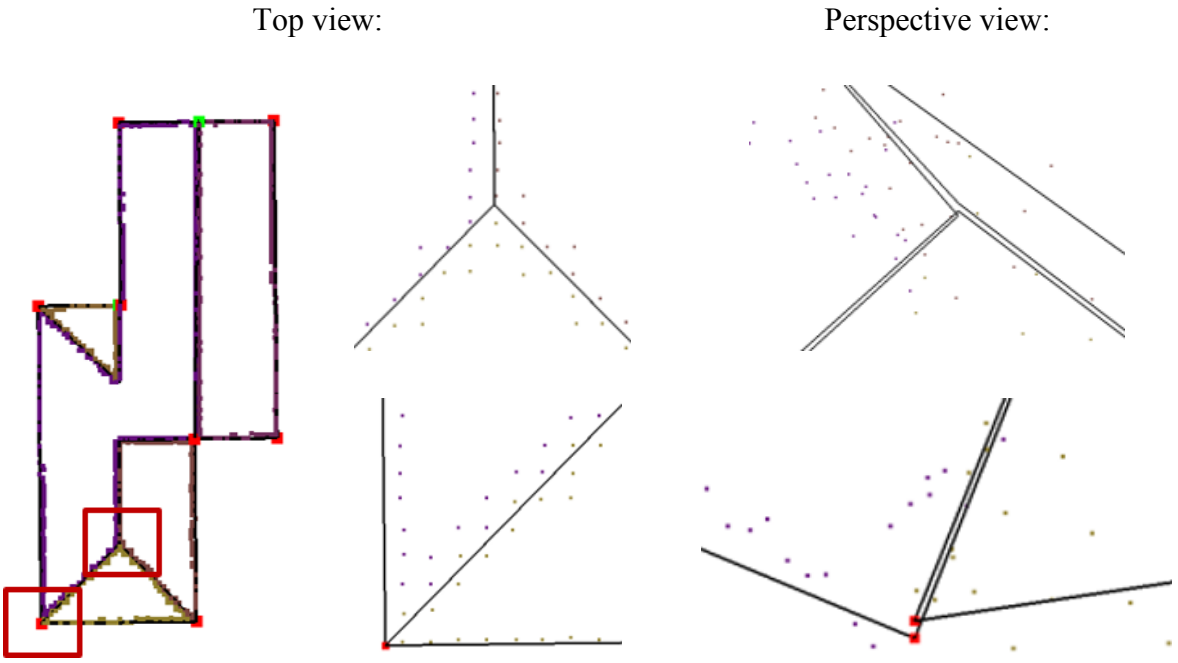


Figure 7.4. Two examples of unintentional gaps between roof planes after applying the MDL based optimization method presented in (Sohn et al., 2012).

A further optimization is presented in (Sohn et al., 2013). It takes additional hyperlines into account which are extracted from a single image. To connect these hyperlines with the already extracted hyperlines, different hypotheses based on CLF are generated and evaluated. An overview of the whole BSP based building reconstruction process is given in Figure 7.5.

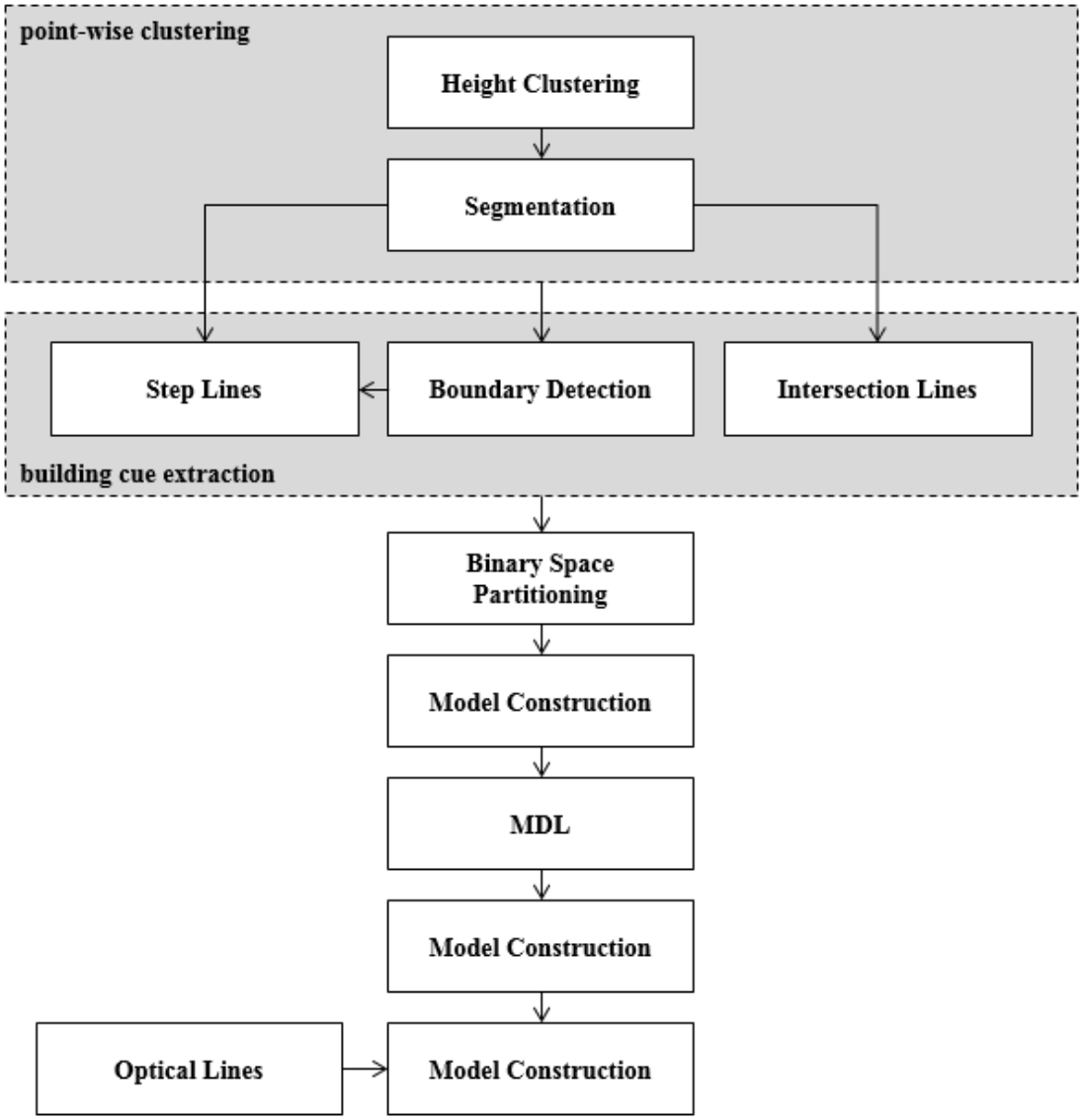


Figure 7.5. Overview of the reconstruction process based on BSP presented in (Sohn et al., 2008; Sohn et al., 2012; Sohn et al., 2013).

7.2 Adjustment of Segments

Based on the intrinsic information about a building gained from the derived multi-scale knowledge graph, segments are adjusted to emphasize regularities in the resulting building model. The proposed adjustment procedure is divided into two separate sub-steps. In the first sub-step, only those segments of building parts that are represented by the multi-scale knowledge graph are adjusted where meaningful building knowledge could be extracted. Then, in the second sub-step all other parts are adjusted according to the results of the first sub-step. In the following, both sub-steps are further explained.

In analogy to the adjustment methods presented in section 6.3, a distinction is made in the first sub-step between local and global adjustments. Instead of half-spaces, the adjustment rules are applied to segments and their plane equations. A local adjustment is performed on a

set of adjacent segments that are aggregated in the multi-scale knowledge graph to a higher-level building feature. Thus, local adjustments are individually applied for every building feature and independent from other building features that are not part of it. In the subsequent global adjustment special care is taken of the relation between the recognized building features, even if they are not adjacent to each other. The two-step adjustment ensures that those segments which are part of the same building feature have a higher impact on its segments than the segments of other building features.

The adjustments in this approach change mainly the slope and the orientation of segments in a local and global context. By using a divisive clustering method, as explained in section 6.3, those segments which feature similar slopes or x-y directions are adjusted to their average value. Additionally, all segments of a recognized building component are translated within a strict predefined threshold during the global adjustment so that building features in the multi-scale knowledge graph can be merged together. It is important to stress that the topological information of each segment as well as its semantic information is considered during the local and global adjustment process. For example, the adjustment of the segments of a subgraph, which is surrounded by a segment of another subgraph, is restricted to remain in their respective segment. Therefore, roof superstructures such as dormers always remain on the same roof plane so that a possible degeneration of the building model is reduced. Furthermore, by taking building knowledge into account, it is ensured that the x-y direction of a subgraph that represents a dormer, is mainly influenced by the main direction of its connected substructure.

Subsequently, more divisive clustering algorithms are carried out in the second sub-step to adjust also those segments in the multi-scale knowledge graph that are not part of a recognized building feature. The adjustment of these segments is performed according to the adjustment result of all those segments that are already part of a building component. In this way, information from already recognized building parts are incorporated in the reconstruction of the remaining parts. For this purpose the global adjustments are repeated but this time including all segments and by considering the following three conditions:

- If more than one adjusted segment of an already recognized building part occurs in a cluster with differing cluster criterion values, the cluster is split again.
- If exactly one adjusted segment of an already recognized building part occurs in a cluster, all unadjusted segments are adapted to its cluster criterion value.
- If no adjusted segment of an already recognized building part occurs in a cluster, all segments are adjusted to their average cluster criterion value.

The combination of all adjustment steps exposes the natural structure in the geometry of a building so that a regularized space partitioning can be generated.

7.3 Regularized Space Decomposition

In this section, the integration of building knowledge into BSP for the generation of regularized building models is explained. For demonstration purposes, the construction of the multi-scale knowledge graph has been limited to the effect that the feature recognition ends with a basic set of primitives so that more complex primitives and links between primitives are not detected. Without considering any building knowledge, binary space partitioning can be performed based solely on the segmentation and boundary detection results, from which intersection and step lines are then formulated as hyperlines. The quality of such a partitioning is especially low for low density point clouds because the real orientations of the hyperlines are difficult to determine. Furthermore, this approach usually produces numerous small partitions as shown in Figure 7.6 which leads to ambiguities during the merging process.

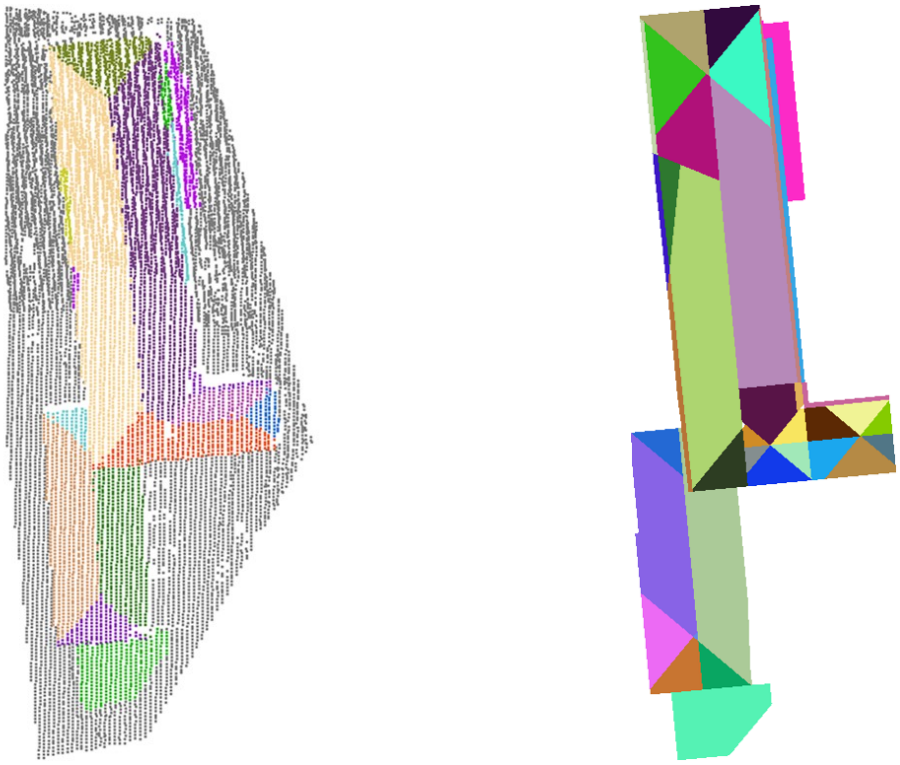


Figure 7.6. Left: The result of the segmentation. Right: The result of the binary space partitioning without considering building knowledge.

In order to overcome this issue, intersection and step lines are determined in the following four steps: First all inner intersection and step lines are collected that are part of a recognized primitive that compose the building. Due to the performed adjustments, these lines are regularized and support the occurrence of symmetrical, parallel and orthogonal structures. Then, all building features in the multi-scale knowledge graph that were not considered in the previous step are used for the estimation of additional lines. Afterward, also those parts of the building where no knowledge has been extracted are considered as follows: The segments are first categorized into different groups. Two groups are merged together if they are part of the

same substructure or if they belong to two different groups which are connected by a building feature. Then, the intersection and step lines between two groups are added. Finally, the lines of the boundary are also added.

An example of the line generation can be found on the left side of Figure 7.7 where the input points that belong to the building are separated into four height clusters represented by different colors. The building feature recognition step detects a total of three one-sided hip roofs whose inner lines are colored in red. Two of them are in the same height cluster and lower-level building features in the multi-scale knowledge graph indicate an L-connection relationship between them. Due to the different ridge line heights of the two connected one-sided hip roofs, the three green intersection lines are extracted based on the plane equations of the segments. Additional lines between two segments which belong to different groups are colored in blue. The black lines are derived from the boundary detection.

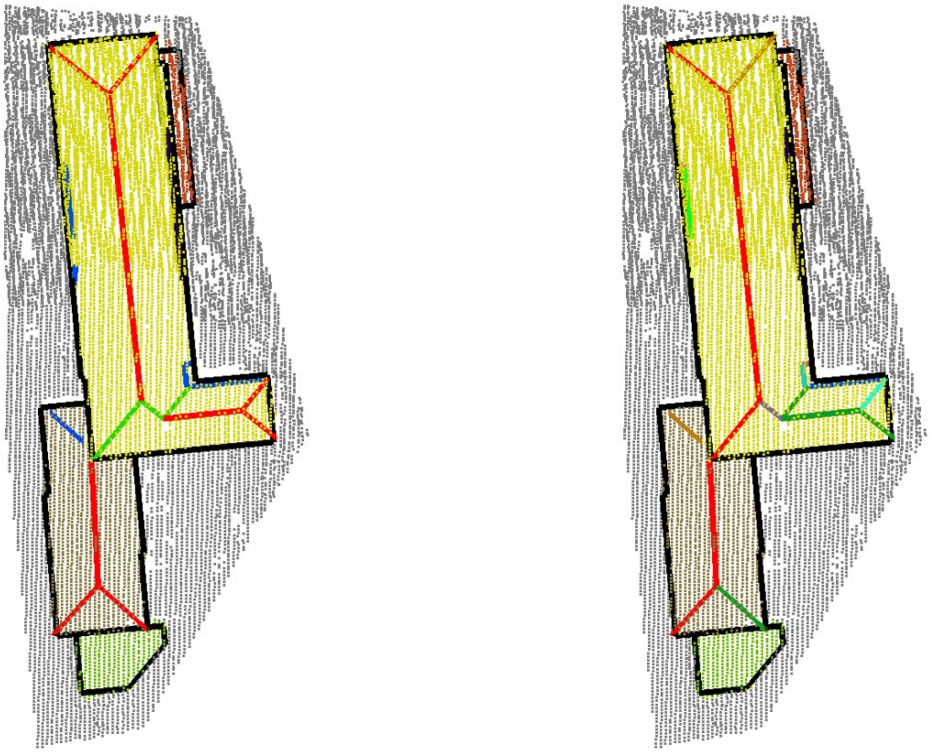


Figure 7.7. Left: Point cloud overlaid with extracted line segments originating from substructures (red), additional building knowledge (green), and segment groups (blue). Right: Point cloud overlaid with accumulated polylines colored for each height cluster according to their priority.

To reduce the number of partitions in the BSP tree, the use of hyperpolylines in the generation of the BSP tree is proposed to avoid unnecessary spatial subdivisions, so that the spatial integrity of planar roof regions is better maintained. For this, all intersection and step lines are treated as line segments and they are connected to form polylines. The start and end points for line segments that are inside a recognized primitive are already well defined through their

building features. The two end points for other lines are determined based on segment intersections and the outmost points on them. The polylines are estimated for each height cluster separately as follows: Choose the longest line segment that is not yet part of any polyline and repeatedly add the longest line segment (that is also not yet part of any polyline) that connects to either end point of the polyline to this new polyline until no more line segments can be added. Repeat until no more polylines can be generated in this way. The result of the polyline estimation is shown on the right side of Figure 7.7. The first extracted polyline of each height cluster in this figure is highlighted in red, the second in dark green, the third in brown, and so on. As shown, because of the polyline formulation as the partitioning element, the binary partitioning can now also be realized with concave borders which often occur in rooftops.

For the binary partitioning of a height cluster, all polylines are used and formulated as hyperpolylines. The order in which they are recursively applied has an impact on the result. Therefore, a partitioning score PS is calculated for each hyperpolyline hpl and its corresponding set of polylines pl as defined in Equation (7.1), where $\alpha, \beta, \gamma,$ and δ are weight factors. The partitioning score is updated during every iteration of the partitioning process after the hyperline with the highest score has been selected and applied to partition the space.

$$PS(pl, hpl) = \alpha \cdot length(pl) + \beta \cdot line\_number(pl) + \gamma \cdot \frac{length(pl)}{length(hpl)} + \delta \cdot PH\left(pl_{segments}\right) \tag{7.1}$$

The partitioning score prioritizes hyperpolylines that are long, have a large number of line segments, and where the ratio of the polyline length and the length of the corresponding hyperline is close to 1. As the points at the two sides of a partition should preferably be homogenous, the score also incorporates a plane homogeneity factor PH that considers the number of points with similar planar properties in each partition. The result of the binary space partitioning with incorporated building knowledge is shown on the left side of Figure 7.8. Compared to the original partitioning in Figure 7.6, the number of partitions is now less than half.

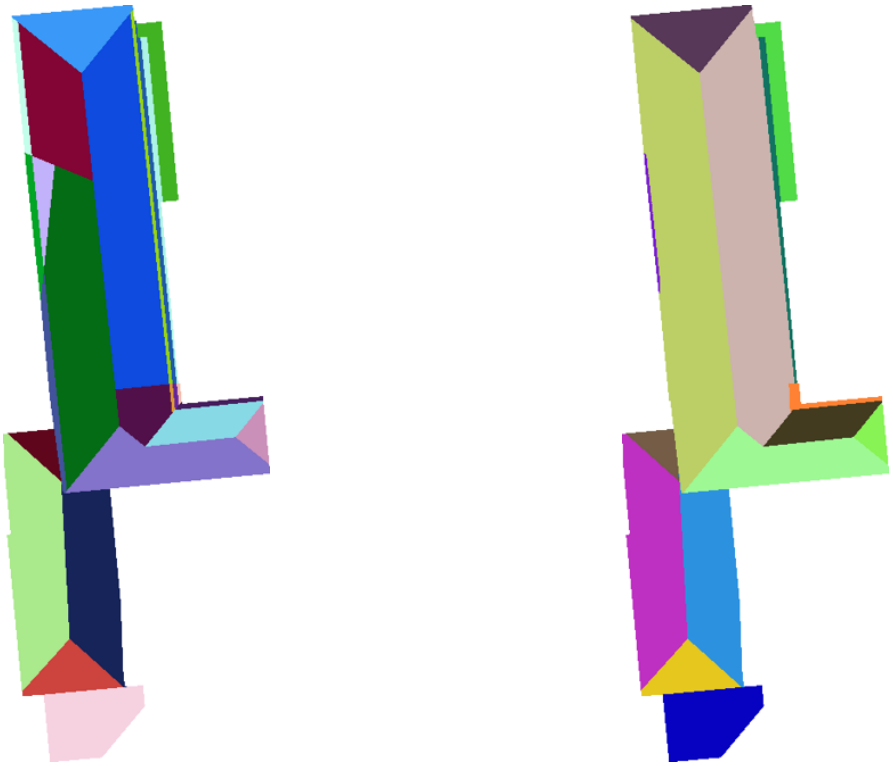


Figure 7.8. The result of the binary space partitioning which takes building knowledge into account before (left) and after (right) the merging process.

After space partitioning, adjacent partitions with similar properties are merged and the plane equations of the new partitions are re-estimated. Note, due to the adjustment step, the resulting plane equation of a partition depends not only on the points in it. Once all partitions with similar normal vectors are merged together, the outline of the regularized building can be extracted directly. During the conversion of the BSP tree into a boundary representation, the outline will result in facade polygons.

For a better comparison with the data-driven reconstruction method presented in (Sohn et al., 2008), the RANSAC plane extraction method that was originally used has been applied in the following examples instead of sub-surface segmentation. In Figure 7.9, the planar segmentation result of a building is presented. Based on this, the BSP is constructed once with (as shown in the bottom row) and once without (as shown in the top row) the integrated building knowledge in the BSP. As shown in the second column of Figure 7.9, the number of partitions before applying the merging process is now reduced from 23 partitions to only 14 partitions. The main difference after the merging process is that the left roof plane on which the two dormers are located could not be merged together in the first case. This also has an impact on the quality of the bottom dormer because the black partition implies a longer ridge line than the orange one. Therefore, the resulting model of the BSP without considering any building knowledge has a gap. In contrast to this, the solidness of the reconstructed building model of the presented approach is shown in Figure 7.10.

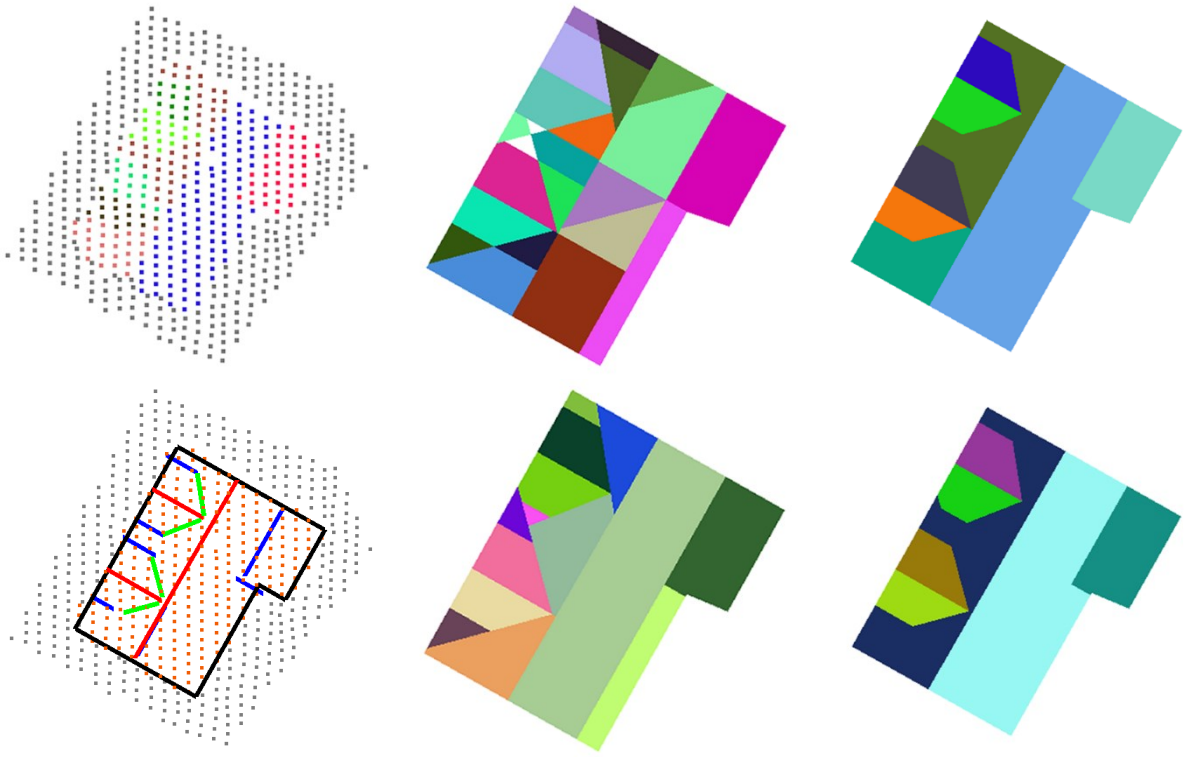


Figure 7.9. Top: The result of the segmentation using a RANSAC plane extraction method and the BSP without considering building knowledge before and after the merging process. Bottom: The result of the line extraction based on building knowledge and the building knowledge considering BSP before and after the merging process.

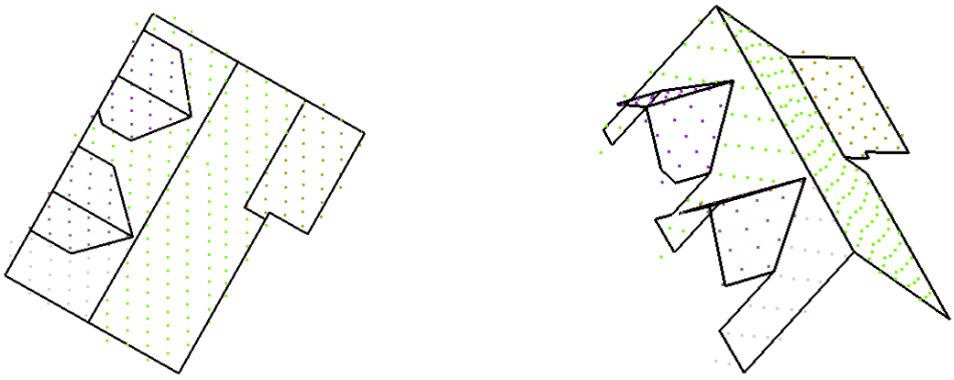


Figure 7.10. The resulting regularized building outline of the automatic reconstruction approach which is directly extracted from the BSP in top and perspective view.

An unsolved issue which still occurs in the resulting outline is shown in Figure 7.10 and in its partitions in Figure 7.9. The dormer in the back misses a small part that is close to the building outline. It is incorporated in the roof plane because the hyperline from the other dormer cuts this part away. Similar problems also occur without the integration of building knowledge, as can be seen in the dark segment in the top right image of Figure 7.9, and is an unavoidable glitch in the greedy strategy of the partitioning process.

8. Results and Discussion

The proposed reconstruction approach has been applied to a benchmark data set to evaluate the resulting building models. The benchmark consists of several test areas that are located at two different continents and of which each features different building characteristics. The benchmark is thus well suited to evaluate the reconstruction result and to verify how well the research objectives stated in section 1.3 are met. A description of the test areas and their characteristics is given in section 8.1. Afterward, the applied main quality metrics are described in section 8.2. Finally, the result of the evaluation is presented and discussed in section 8.3.

8.1 Data Sets

In this thesis, the freely available benchmark data set for urban classification and 3D reconstruction provided by the International Society of Photogrammetry and Remote Sensing (ISPRS), German Society for Photogrammetry, Remote Sensing and Geoinformation (DGPF), and Teledyne-Optech was used to evaluate the results of the proposed reconstruction method. This benchmark data set has been chosen to make the results of the proposed reconstruction approach comparable to other existing reconstruction approaches. As shown in Figure 8.1, the benchmark consists of two different test sites with five test areas (= areas of interest (AOIs)) in total. The first test site was captured in Vaihingen (Germany) and has three test areas consisting of different types of buildings. The second test site was acquired over the downtown of Toronto (Canada) and features two different test areas. The details of both test sites and their data sets are described in subsection 8.1.1 and 8.1.2, respectively, and essentially summarizes the information presented in (Rottensteiner et al., 2013).

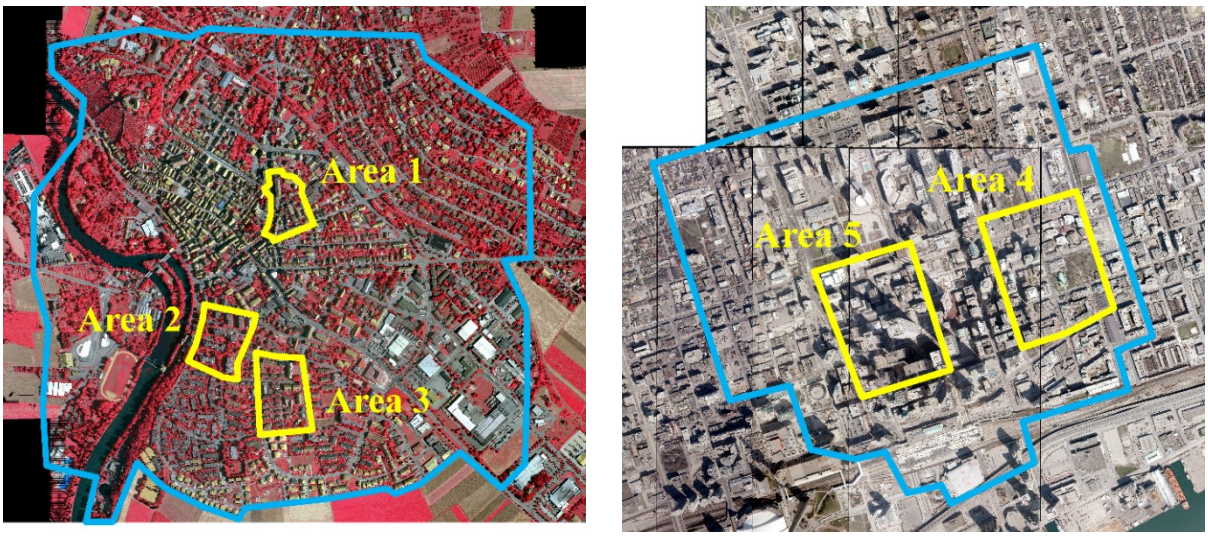


Figure 8.1. The Vaihingen (left) and Toronto (right) test sites (blue) and their AOIs (yellow).

8.1.1 Vaihingen Test Site

The ALS data of the Vaihingen test site was provided by DGPF. It was acquired on 21 August 2008 and consists of 10 strips of which 5 overlap with the test site. An overview of the overlapping ALS strips is given in Figure 8.2. The data set was captured by a Leica ALS50 system with 45° field of view and a mean flying height above ground of 500 m. The average point density of each strip is approximately 4 points/m². Due to an average strip overlap of 30 %, the average point density is 6.7 points/m² (i.e., ~ 0.39 m point spacing). In order to correct systematic errors in georeferencing, the original point clouds were post-processed by strip adjustment.

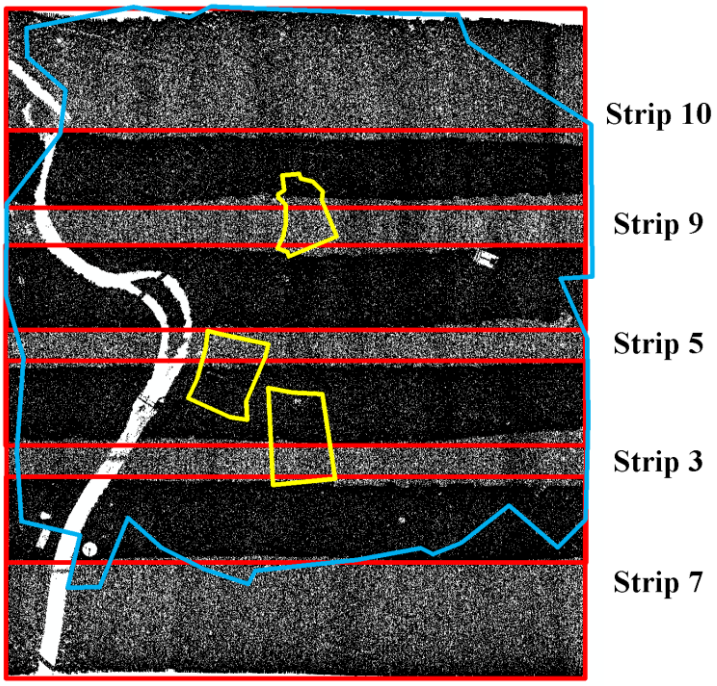


Figure 8.2. ALS data of the Vaihingen test site with a reduced point density.

For the evaluation of the automatically reconstructed building models, three different test areas were selected whose buildings were manually modeled by stereo plotting based on aerial high-resolution pan-sharpened color images. The images were captured on 24 July and 6 August 2008 with 65 % forward lap and 60 % side lap using an Intergraph/ZI DMC (Digital Mapping Camera) with a GSD (Ground Sampling Distance) of 8 cm and a radiometric resolution of 11 bits. The resulting 3D building models correspond to the level of detail LoD 2 according to the CityGML standard (Gröger et al., 2008) and have an accuracy in planimetry and height of about 10 cm. The three test areas are magnified shown in Figure 8.3 and their characteristics are summarized in Table 8.1. For further details, see (Cramer, 2010), (Haala et al., 2010), and (Rottensteiner et al., 2012).

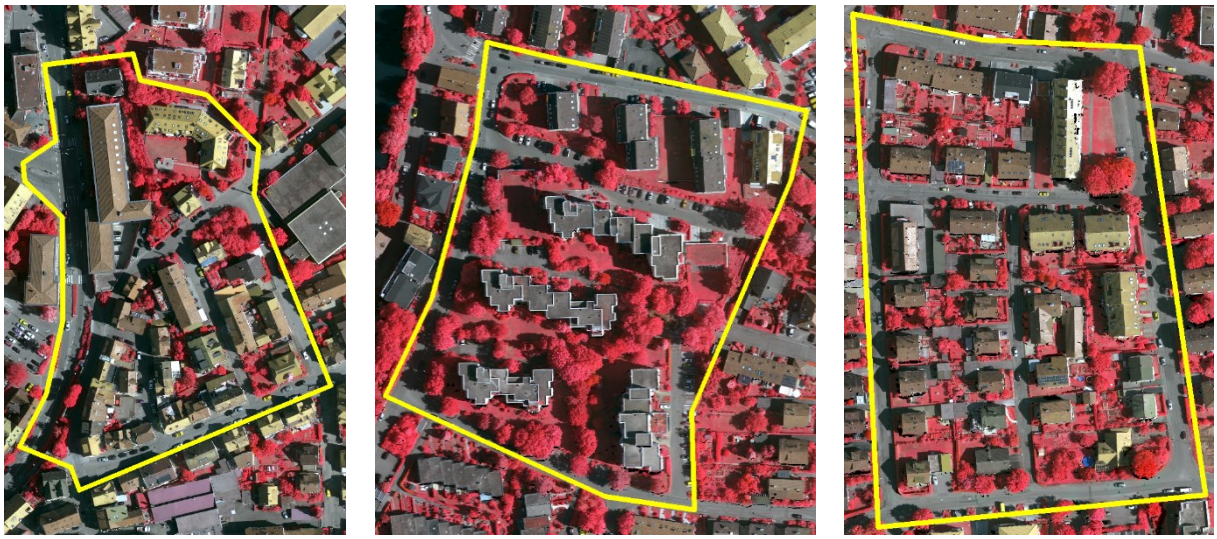


Figure 8.3. The three test areas 1 (left), 2 (middle), and 3 (right) of the Vaihingen test site.

Table 8.1. Characteristics of the three test areas in the Vaihingen test site according to (Rottensteiner et al., 2014).

	Description	**Size**	**IDs of Overlapping Strips**	**Number of buildings**
Area 1: "Inner City"	Historic buildings with complex shapes in the center of the city.	125 m x 200 m	9 and 10	37
Area 2: "High Riser"	Few high-rising residential buildings with mixed flat and sloped roof types.	170 m x 190 m	3 and 5	14
Area 3: "Residential Area"	Purely residential area with small detached houses.	150 m x 220 m	3 and 5	56

8.1.2 Toronto Test Site

The ALS data of the Toronto test site was provided by Teledyne-Optech, First Base Solutions, GeoICT Lab at York University, and ISPRS WG III/4. It was acquired in February 2009 and consists of 6 strips. An overview of the strips is given in Figure 8.4. The data set was captured with Optech's ALTM-Orion M, which operates at a wavelength of 1064 nm and scans the underlying topography with a scan frequency of 50 Hz and a scan width of 20 degrees. A sampling rate of 100 kHz was used to digitize the reflected echoes. The mean flying height above ground was 650 m. The average point density with the overlap is approximately 6 points/m² (i.e., ~ 0.41 m point spacing).

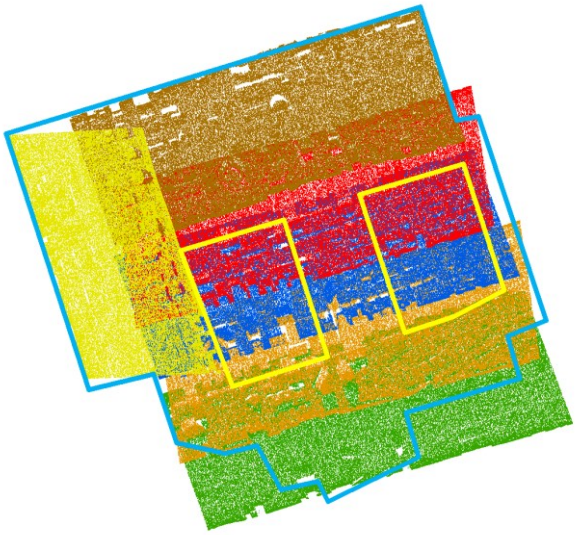


Figure 8.4. ALS data of the Toronto test site with a reduced point density.

For the evaluation of the automatically reconstructed building models, two different test areas were selected whose buildings were manually modeled by stereo plotting based on aerial high-resolution pan-sharpened color images. The images were captured with 60 % forward lab and 30 % side lap using an UltraCam-D with a GSD of 15 cm and a radiometric resolution of 8 bits. The resulting 3D building models correspond to LoD 2 according to the CityGML standard (Gröger et al., 2008). The planimetry and height accuracy of well-defined points is about 20 cm and 15 cm, respectively. The two test areas are magnified shown in Figure 8.5 and their characteristics are summarized in Table 8.2. For further details, see (Rottensteiner et al., 2012).

Figure 8.5. The two test areas 4 (left) and 5 (right) of the Toronto test site.

Table 8.2. Characteristics of the two test areas in the Toronto test site according to (Rottensteiner et al., 2014).

	Description	Size	IDs of Overlapping Strips	# of buildings
Area 4: "Inner City"	Mixture of low and high-storey buildings	530 m x 600 m	1, 2, 3, and 4	58
Area 5: "High Riser"	Cluster of high-rise buildings in a typical central business district of North Amerika.	530 m x 600 m	2, 3, and 4	38

8.2 Evaluation Methods

For a better comparison with other building reconstruction methods that were applied to the same benchmark data set, the same quality metrics were used to evaluate the resulting 3D building models of the developed reconstruction approach. The metrics are therefore equivalent to the evaluation methods of the ISPRS benchmark project (Rottensteiner et al., 2012) and incorporate metrics described in more detail in (Rutzinger et al., 2009). In the following, a brief description of the main quality metrics is given:

- Completeness (Comp, Comp_{10}), which is also referred to as producer's accuracy, represents the percentage of roof planes in the reference that were detected. It is defined as

$$Completeness = \frac{\|TP\|}{\|TP\| + \|FN\|} \tag{8.1}$$

  where TP and FN are, from the point of view of the reference, the numbers of true positives and false negatives, respectively. In contrast to Comp, Comp_{10} considers only roof planes larger than 10 m².
- Correctness (Corr, Corr_{10}), which is also referred to as user's accuracy, indicates how well the detected roof planes correspond to the reference. It is defined as

$$Correctness = \frac{\|TP\|}{\|TP\| + \|FP\|} \tag{8.2}$$

  where TP and FP are, from the point of view of the result, the numbers of true positives and false positives, respectively. In contrast to Corr, Corr_{10} considers only roof planes larger than 10 m².

- Quality (Q, Q_{10}) represents a compound performance metric that balances completeness and correctness. It is defined as

$$Quality = \frac{1}{\frac{1}{Completeness} + \frac{1}{Correctness} - 1} \tag{8.3}$$

  where $Completeness$ and $Correctness$ are defined as in Equation (8.1) and Equation (8.2), respectively. In contrast to Q, Q_{10} considers only roof planes larger than 10 m².

For a detailed analysis of the automatically reconstructed building models, completeness, correctness, and quality were determined on a per-area level as well as on a per-roof-plane level. As stated in (Rutzinger et al., 2009), both evaluation methods have advantages and disadvantage. The area-based evaluation has, for example, the advantage that it can be easily applied by comparing the labels of homologous pixels of a raster. The decision whether an area is classified as TP is thus unambiguous. However, differences between the reference and the result data may be caused not only by errors in the result but also by quantization and sampling effects, differences in the semantic definition of which objects belong to a building, inaccuracies in the reference, and others (Foody, 2002). Note, in this thesis, a pixel size of 0.2 m was chosen. The object-based evaluation, on the other hand, is generally less sensible to errors in the outline of a roof plane because a roof plane is classified as TP if it substantially overlaps with a roof plane in the reference. For this, however, it introduces a threshold to decide whether or not a substantial overlap exists. The object-based evaluation has therefore the disadvantage that a subjective aspect is introduced into the evaluation (Shufelt, 1999). A further disadvantage of the object-based evaluation is that the relation to the covered area is lost. In this thesis, an overlap of at least 50 % was required for a reference and a result roof plane to be counted as a true positive.

In addition to completeness, correctness, and quality, the following quality metrics were incorporated in this thesis:

- RMS (root mean square) (RMS_{XY}, RMS_Z) represents a value for geometrical errors. The errors of planimetric distances of extracted roof plane boundary points to their nearest neighbors on the corresponding reference boundaries are represented in RMS_{XY}, which is defined as

$$RMS_{XY} = \sqrt{\frac{\sum d^2}{N}} \tag{8.4}$$

  where d is the Euclidian distance in x-y direction between corresponding points and N the number of detected correspondences. Note, distances between corresponding points larger than 3 m were discarded. Furthermore, a corresponding point on a

reference roof plane boundary does not necessarily correspond to a vertex of that polygon. In contrast, the errors of height differences that are represented in RMS_Z are not calculated from the boundaries but by comparing two synthetic DSMs generated from the extracted building models and the reference. For this, the vertical distances between corresponding planes at all pixels that are assigned to them are considered. It is thus defined as

$$RMS_Z = \sqrt{\frac{\sum \left(z_{ref_i} - z_{res_j} \right)^2}{N}} \tag{8.5}$$

where z_{ref_i} and z_{res_j} are two height values with the same planimetric position but located in the reference plane ref_i and in the result plane res_j.

- Topology represents the number of 1:1, 1:N, N,1, and N:M relations between roof planes in the reference and in the automatically reconstructed building models.

8.3 Evaluation of the Semantic 3D Building Models

The proposed grammar-guided reconstruction approach has been applied on the ISPRS benchmark data set to evaluate its reconstruction capabilities. The resulting building models of the three test areas in Vaihingen and the two test areas in Toronto are shown in Figure 8.6. All reconstructed buildings are represented as valid 3D models that can be used for different applications. Since these models are semantically labeled, certain building details or building types can be optionally hidden if they are not required for a specific application. The average processing time for the reconstruction of a single building with current hardware of an ordinary computer was less than one second. A summary of the evaluation result is given in the following. A more detailed discussion and a comparison to other reconstruction approaches is presented for each test area separately in the subsequent subsections. Note, for a better comparison with the evaluation results of other building reconstruction approaches, the choice of colors and their descriptions were essentially adopted in this section from (ISPRS, 2017). Furthermore, although the evaluation revealed that some building parts are missing in the reference, the reference building models have not been changed so that the comparability with other semi- or fully automatic reconstruction approaches was maintained.

Figure 8.6. Reconstruction result of the three test areas in Vaihingen (top row) and the reconstruction result of the two test areas in Toronto (bottom row).

The result of the area-based evaluation with a pixel size of 0.2 m is summarized in Table 8.3. As can be seen, the proposed reconstruction approach achieved for all test areas an average completeness, correctness, and quality of 96.0 %, 94.8 %, and 91.1 %, respectively. Furthermore, the completeness, correctness, and quality within each test site varied only slightly (less than 3 %) so that they are for each test area at least 94.3 %, 89.6 %, and 89.1 %, respectively. It can be thus stated that the area of the resulting building models properly overlaps with the area of the reference building models. The result of the area-based evaluation further shows that, in relative terms, less area of buildings was reconstructed on average in the Vaihingen test site than in the Toronto test site (94.3 % compared to 98.7 %) but that the reference correspond better to the reconstructed areas of the Vaihingen test site than to the reconstructed areas of the Toronto test site (97.2 % compared to 91.1 %). For further investigations, it is interesting to note that, on the one hand, the highest completeness was achieved in AOI 5 (99.4 %) while it features with 89.6 % at the same time the lowest correctness of all recon-

structed AOIs. On the other hand, the highest correctness was achieved in AOI 1 (97.7 %) while it features, together with the other AOIs in Vaihingen, with 94.3 % the lowest completeness.

Table 8.3. Evaluation result on a per-area level (pixel size: 0.2 m).

Data set	Area	Comp [%]	Corr [%]	Q [%]
Vaihingen	1	94.3	97.7	92.3
	2	94.3	97.5	92.1
	3	94.3	96.5	91.2
	Sub-total	94.3	97.2	91.9
Toronto	4	97.9	92.5	90.7
	5	99.4	89.6	89.1
	Sub-total	98.7	91.1	89.9
Total		96.0	94.8	91.1

In the object-based evaluation, the proposed reconstruction approach achieved for all test areas an average completeness, correctness, and quality of 83.9 %, 93.1 %, and 78.8 %, respectively (see Table 8.4). The completeness and the quality of the object-based evaluation is thus significantly lower than the average completeness and quality of the area-based evaluation. This indicates that particularly small roof planes are missing in the result. This assumption is supported by the significant increase of the average completeness and quality of all test areas that considers only roof planes larger than 10 m² (from 83.9 % to 93.5 % and from 78.8 % to 89.2 %, respectively). It becomes especially apparent in AOI 2 for which the completeness is raised from 72.5 % to 91.5 %. But the impact of ignoring small roof planes on the completeness is also apparent in the other AOIs, for which the completeness still increases between 3.4 % and 11.4 %. In contrast, the average correctness of all test areas is essentially stable against different roof plane sizes and increases for all AOIs only slightly from 93.1 % to 95.3 % and for each AOI only between 0.2 % and 3.8 %. The high correctness values and the small increase indicate that there is in general a high correspondence between the reference and the result roof planes regardless of their area. Only AOI 5 features with 82.2 % (or 86.0 % if only roof planes larger than 10 m² are considered) a significantly lower correctness than the other AOIs. Furthermore, the result of the object-based evaluation shows analogous to the area-based evaluation result that, in relative terms, less roof planes were reconstructed on average in the Vaihingen test site than in the Toronto test site (82.3 % compared to 86.4 %) but that the reference roof planes correspond better to the reconstructed roof planes of the Vaihingen test site than to the reconstructed roof planes of the Toronto test site (96.6 % compared to 87.8 %).

Table 8.4. Evaluation result on a per-roof plane level (threshold for classification as a true positive is 50%).

Data set	Area	Comp [%]	Corr [%]	Q [%]	$Comp_{10}$ [%]	$Corr_{10}$ [%]	Q_{10} [%]	Comp balanced by area [%]	Corr balanced by area [%]	Q balanced by area [%]
Vaihingen	1	89.2	95.9	86.0	92.6	98.6	91.3	94.2	99.0	93.4
	2	72.5	97.1	70.9	91.5	100	91.5	94.9	99.7	94.7
	3	85.1	96.7	82.7	92.7	98.3	91.3	95.9	99.6	95.6
	Sub-total	82.3	96.6	79.9	92.3	99.0	91.4	95.0	99.4	94.6
Toronto	4	88.1	93.4	83.0	94.7	93.6	88.9	98.2	95.3	93.7
	5	84.7	82.2	71.6	96.1	86.0	83.0	99.4	91.5	90.9
	Sub-total	86.4	87.8	77.3	95.4	89.8	86.0	98.8	93.4	92.3
Total		83.9	93.1	78.8	93.5	95.3	89.2	96.5	97.0	93.7

As mentioned in section 8.2, the relation to the covered area is basically lost in object-based evaluations. In order to overcome this disadvantage, completeness, correctness and quality were also balanced by area in the sense that each roof plane was weighted by its total area. For many applications, these quality metrics are therefore most meaningful. As shown in Table 8.4, the completeness, correctness, and quality balanced by area are for all test areas on average 96.5 %, 97.0 %, and 93.7 %. Additionally, it can be stated that the reconstructed test areas feature a higher average completeness in Toronto than in Vaihingen (98.8 % compared to 95.0 %). The same still applies to each test area within Toronto and Vaihingen. But, analogous to the previously described pixel- and object-based evaluation results, the average correctness of the automatically reconstructed building models is in Vaihingen higher than in Toronto (99.4 % compared to 93.4 %). Since the completeness, correctness and quality are also within each AOI respectively at least 94.2 %, 91.5 %, and 90.9 %, it can be stated that all reconstructed test areas can serve as a good basis for many different applications.

The evaluation result of the geometric accuracy is summarized in Table 8.5 and has to be seen in relation to the ALS point spacing (Vaihingen: ~ 0.39 m, Toronto: ~ 0.41 m) and to the accuracy of the reference models (Vaihingen (planimetry and height accuracy): ~ 10 cm, Toronto (planimetry and height accuracy): ~ 20 cm and ~ 15 cm, respectively). It shows for the extracted roof plane boundaries of the test areas in Vaihingen and Toronto an average RMS_{XY} of 0.60 m and 0.95 m, respectively. The geometric accuracy in planimetry is thus lower for the automatically reconstructed building models in Vaihingen than in Toronto. This still applies if the RMS_{XY} of each test area is separately compared with each other because they differ from the average RMS_{XY} values only by up to 0.07 m and 0.02 m, respectively. The same applies to the RMS_{XY} of the reference boundaries, which is on average for Vaihingen 0.58 m and for Toronto 1 m. Additionally, the deviation of each test area is from

these average values only less or equal than 0.10 m and 0.05 m, respectively. Since the RMS_{XY} of the extracted boundaries and the RMS_{XY} of the reference boundaries are for all test areas on average with 0.74 m and 0.75 m less than twice the ALS point spacing, they confirm a good geometric accuracy of the reconstructed building models.

Table 8.5. Evaluation result of geometric accuracy.

Data set	Area	RMS_{XY} of extracted boundaries [m]	RMS of centers of gravity of extracted objects (X / Y) [m]	RMS_{XY} of reference boundaries [m]	RMS of centers of gravity of reference objects (X / Y) [m]	RMS_Z of planes [m]	RMS_Z of planes found to correspond [m]
Vaihingen	Area 1	0.54	0.52 / 0.59	0.68	0.56 / 0.63	0.78	0.74
	Area 2	0.67	0.39 / 0.55	0.55	0.46 / 0.47	0.77	0.36
	Area 3	0.58	0.36 / 0.47	0.51	0.58 / 0.65	0.36	0.21
	Sub-total	0.60	0.42 / 0.54	0.58	0.53 / 0.58	0.64	0.44
Toronto	Area 4	0.93	0.78 / 0.81	0.95	0.62 / 0.69	3.71	3.04
	Area 5	0.97	0.62 / 0.78	1.04	0.84 / 0.84	8.33	0.94
	Sub-total	0.95	0.70 / 0.80	1.0	0.73 / 0.77	6.02	1.99
Total		0.74	0.53 / 0.64	0.75	0.61 / 0.66	2.79	1.06

In order to weaken the strong influence of those roof planes that feature a large number of vertices in their boundary, RMS_X and RMS_Y of the centers of gravity were calculated as well for each roof plane. The average RMS_X and RMS_Y of the extracted and the reference boundaries support for Vaihingen (0.42 m / 0.54 m and 0.53 m / 0.58 m) as well as for Toronto (0.70 m / 0.80 m and 0.73 m / 0.77 m) the statements that the reconstructed building models have a good geometric accuracy in relation to the point spacing of the input data. Furthermore, the geometric accuracy of the reconstructed buildings is generally higher in Vaihingen than in Toronto. This also becomes apparent when their average RMS_Z values are compared with each other. If all planes are considered, the average RMS_Z is significantly lower for the test areas in Vaihingen than in Toronto (0.64 m compared to 6.02 m). Particularly AOI 5 features with 8.33 m a very high RMS_Z that is, however, significantly reduced to 0.94 m if only pairs of corresponding planes are considered. The same applies to the other test area in Toronto for which the average RMS_Z of Toronto is decreased from 6.02 m to 1.99 m. Compared to this, the decrease of the RMS_Z is significantly lower in Vaihingen (from 0.64 m

to 0.44 m) than in Toronto. Consequently, there is a better roof plane correspondence for the test areas in Vaihingen than in Toronto. However, it has to be considered that the height variance and the geometric inaccuracy of the reference models are significantly higher in Toronto than in Vaihingen. Further reasons are explained in the subsequent subsections.

A similar pattern can be observed in the result of the topological evaluation in which 1:1, 1:N, N:1, and N:M relations between roof planes in the reference and the result were detected. As shown in Table 8.6, the percentage of 1:1 relations is for the test areas in Vaihingen with 73.6 % on average clearly dominant whereas this is less discernable for AOI 4 and AOI 5 in Toronto (64.5 % and 54.3 %, respectively). A general trend that shows if either under- or over-segmentation is the dominant type of topologic error is, however, not clearly recognizable. In AOI 2 and AOI 4, for example, over-segmentation might be the dominant type of topologic error, whereas under-segmentation might be a major issue of AOI 3 and AOI 5. Note, some topologic errors are due to different approximations or due to missing building parts in the reference and do not necessarily represent errors in the automatically reconstructed building models. Therefore, a perfect match between the result and the reference, in which only 1:1 relations are present, is only barely possible and not really wanted.

Table 8.6. Evaluation result of the roof plane topology (reference : result).

Data set	Area	Topo 1:1	Topo 1:N	Topo N:1	Topo N:M
Vaihingen	**1**	163	24	22	14
	2	29	14	3	1
	3	128	6	23	8
Toronto	**4**	401	104	52	65
	5	134	17	61	35

In the following subsections, a more detailed evaluation of the automatically reconstructed building models is presented.

8.3.1 Vaihingen: AOI 1

The 37 buildings of AOI 1 consist of historic buildings with complex shapes in the center of the city Vaihingen. In order to determine the areas in which buildings were reconstructed, a pixel-based building detection was carried out once for the manually reconstructed building models of the reference and once for the automatically reconstructed building models of the result. As shown in Figure 8.7, all buildings of the reference greater than a certain size have been automatically reconstructed in AOI 1. Furthermore, their area and their outer 2D shape closely resemble to each other. However, there are also some buildings with a rather small area in the reference that are not represented in the result. This is, for example, the case for the two buildings in the southwest and the building in the courtyard of the dark brown building in the east of the test area. Furthermore, some building components that are in the reference

attached to large buildings are missing in the result. This is, for example, the case for the large L-shaped building in the northwest and the blue building in the east.

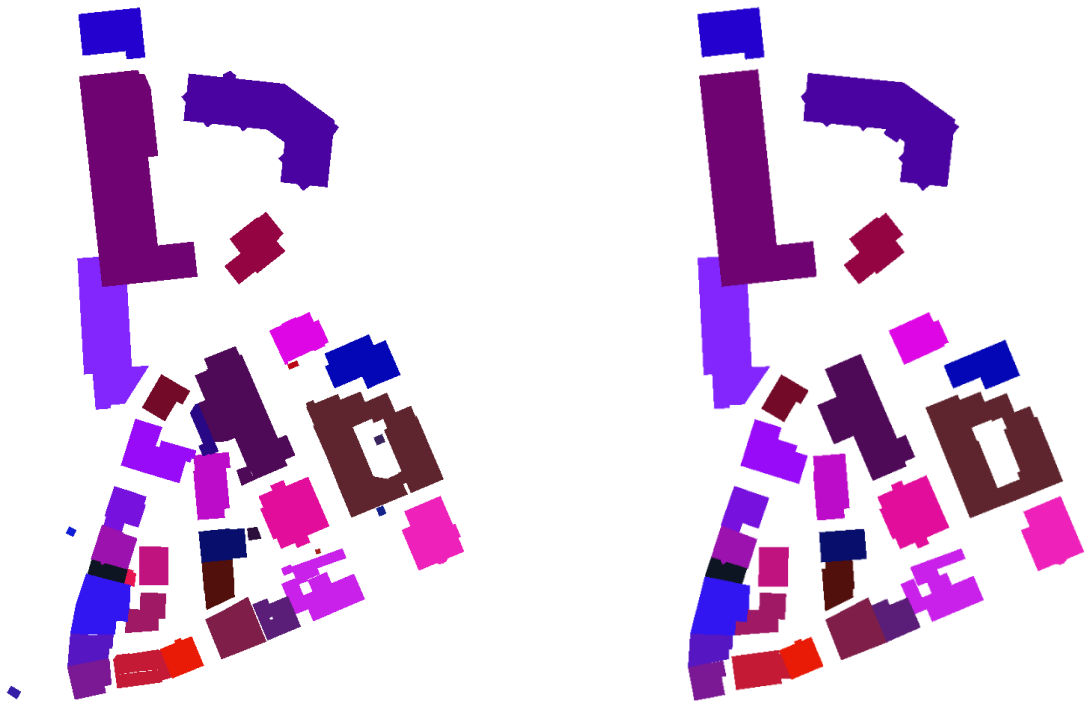


Figure 8.7. Pixel-based building detection result of the building models in the reference (left) and in the result (right).

To highlight the differences between the areas of reconstructed buildings in the reference and the result, TP, FN, and FP were calculated pixel by pixel based on the building detection result. The resulting image is shown in Figure 8.8. As can clearly be seen, the area of true positive pixels is dominant which means that most manually reconstructed areas in the reference were also automatically reconstructed. However, there are several pixels classified as false negative of which most of them comprise the aforementioned small buildings and attached building components that are missing in the result. The reason for this is that most of their points have been wrongly classified as non-building points during the classification procedure so that they were not considered as input of the proposed reconstruction approach (e.g., the two buildings in the southwest). Conversely, also several points were assigned to the class of building points during the classification that are not part of the reference. These points primarily belong to facades which explains why the majority of false positive pixels are located close to building outlines. As a consequence, facades in the result are compared to the reference sometimes slightly shifted or rotated. However, there are also some false positive pixels that represent small intrusions in the reference. These missing intrusions are often due to the proposed adjustment step which was initially introduced to support natural regularities and to avoid small jagged fractures. Thereby, small intrusions were in some cases discarded so that the resulting building shapes represent a generalization of the reality and thus become in many cases more pleasant to the human eye. Consequently, the impact of the proposed

adjustment procedure has to be weaken if the shape of the resulting 3D building models needs to closer resemble the reality.

Figure 8.8. Evaluation result of the pixel-based building detection result (yellow: true positive pixels, blue: false negative pixels, and red: false positive pixels).

For a more detailed analysis, building detection was carried out on a roof plane level as well. As can be seen in Figure 8.9, there is a high correspondence between detected roof planes in the reference and in the result. Particularly large roof planes were reliably reconstructed. The areas of corresponding roof planes and those in which differences occur are highlighted in Figure 8.10. Here, most areas that are classified as false negative in Figure 8.8 were assigned to the dark blue colored class. This indicates that planes in the reference are missing in the result and that it would be insufficient to change only the shape of existing planes in the result. In contrast, many pixels that are classified as false positive in Figure 8.8 were not assigned to the dark red colored class. This supports the previously made statement that these false positive pixels basically belong to facades and thus do not introduce additional roof planes. Additionally, some roof planes can be still identified in the result that are outside all reference buildings. This is, for example, the case for the curved L-shaped building in the north. A closer look on this building reveals that some details of this building were not manually reconstructed in the reference (see Figure 8.11). Furthermore, some pixels in Figure 8.10 are colored in yellow which means that they are not part of a building in the reference and in the result although a sufficient overlap between a roof plane in the reference and in the result was found. These areas mainly represent either automatically reconstructed intrusions that are not present in the reference or intrusions that are discarded in the result due to generalization purposes; particularly the latter one prevails. Further differences between the

roof planes in the reference and in the result are colored in Figure 8.10 in bright blue and bright red. In the building ensemble in the southwest, for example, it can be seen that a further dormer was reconstructed in the result and that the overlap of the underlying roof plane becomes thereby insufficient.

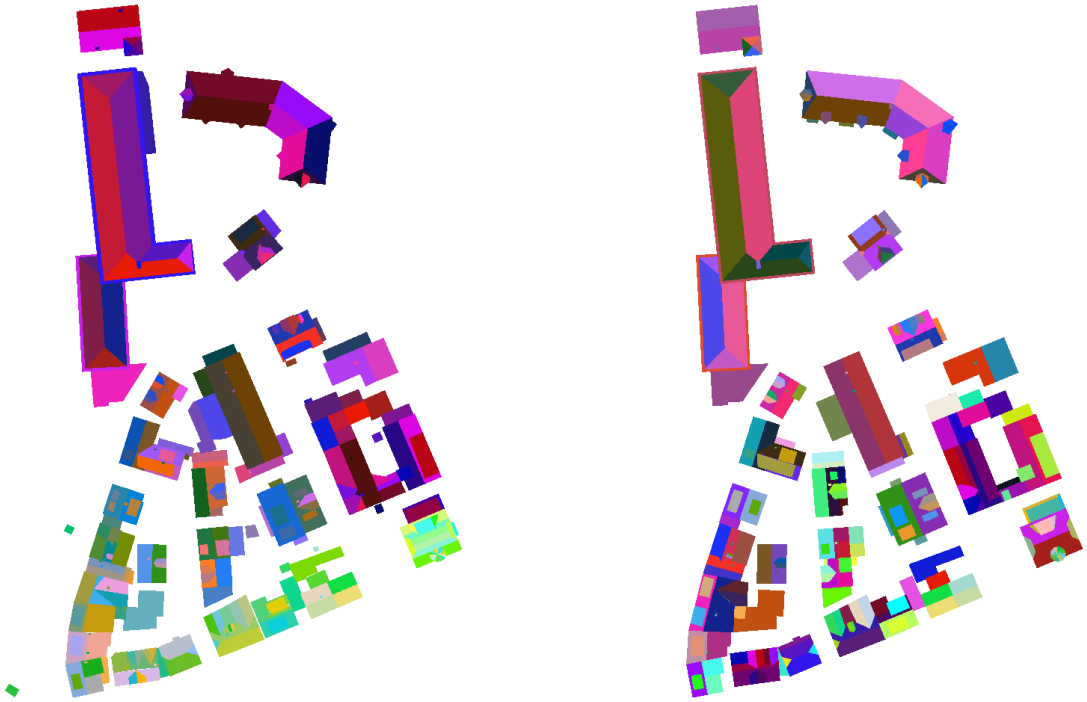
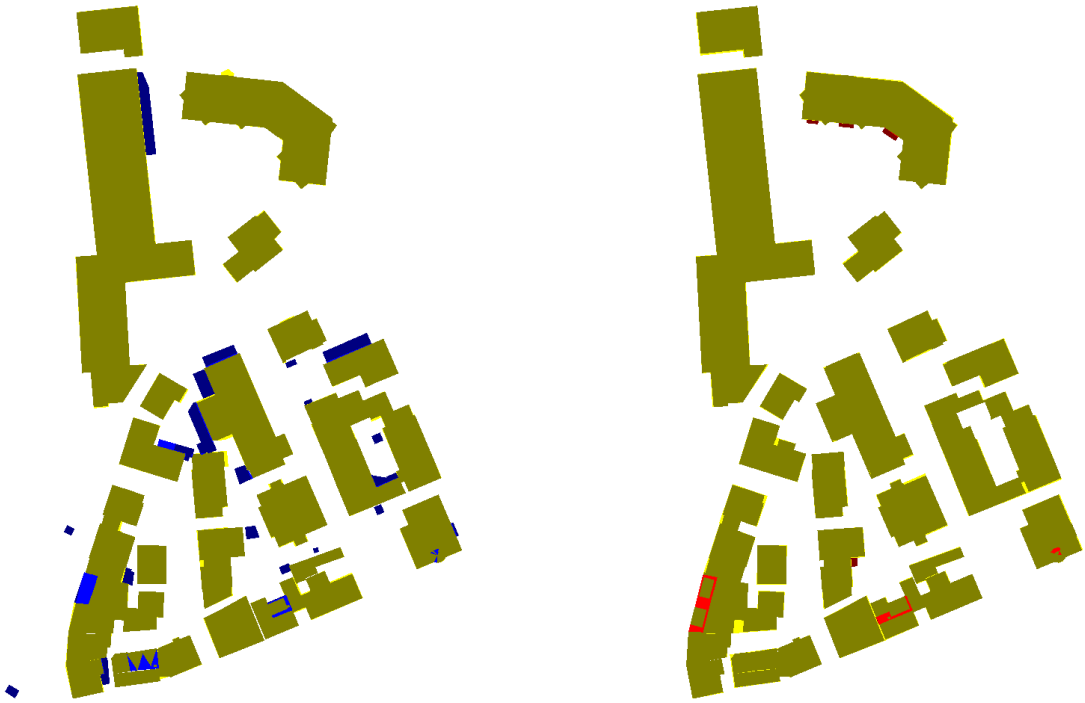


Figure 8.9. Roof plane detection result of the building models in the reference (left) and in the result (right).



Ochre: Pixels available in both the reconstruction results and the reference and belonging to a reference plane found to have sufficient overlap with planes in the roof reconstruction results.

Yellow: Pixels that belong to a reference plane found to have sufficient overlap with planes in the roof reconstruction results but that are not inside a building in the reconstruction results.

Dark blue: Pixels in reference roof planes not having sufficient overlap with detected roof planes and not being inside a building in the reconstruction results.

Bright blue: Pixels in reference roof planes not having sufficient overlap with detected roof planes, but being inside a building in the reconstruction results.

Ochre: Pixels available in both the reconstruction results and the reference and belonging to a detected roof plane found to have sufficient overlap with planes in the reference.

Yellow: Pixels that belong to a detected roof plane found to have sufficient overlap with planes in the reference but that are not inside a building in the reference.

Dark red: Pixels in detected roof planes not having sufficient overlap with roof planes in the reference and not being inside a building in the reference.

Bright red: Pixels in detected roof planes not having sufficient overlap with reference planes, but being inside a building in the reference.

Figure 8.10. Evaluation result of the roof plane detection result once from the point of view of the reference (left) and once from the point of view of the result (right).

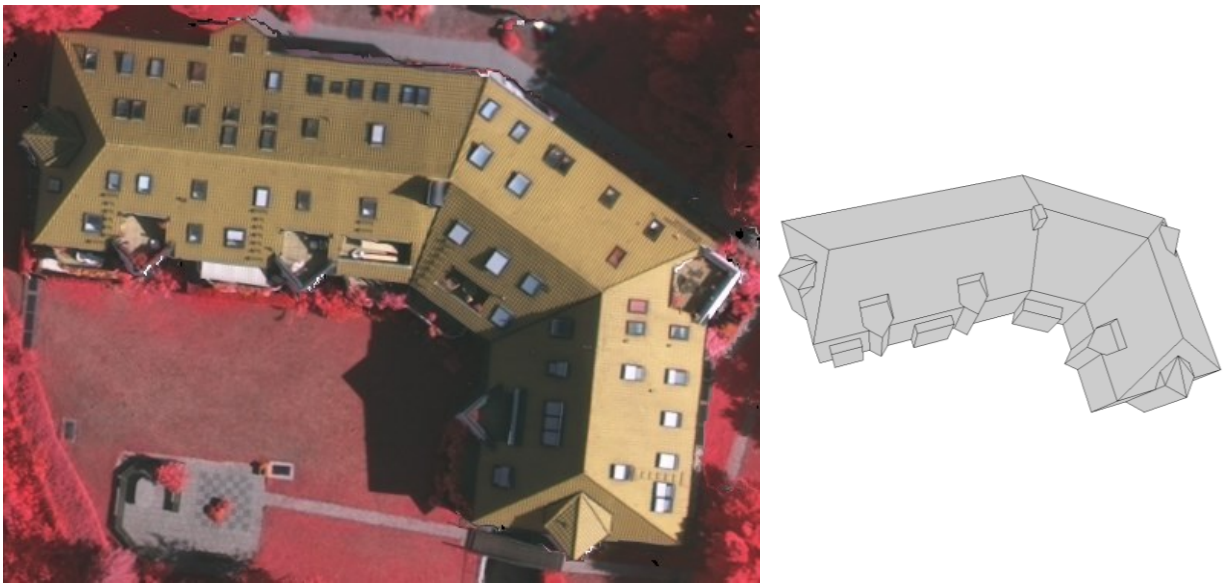
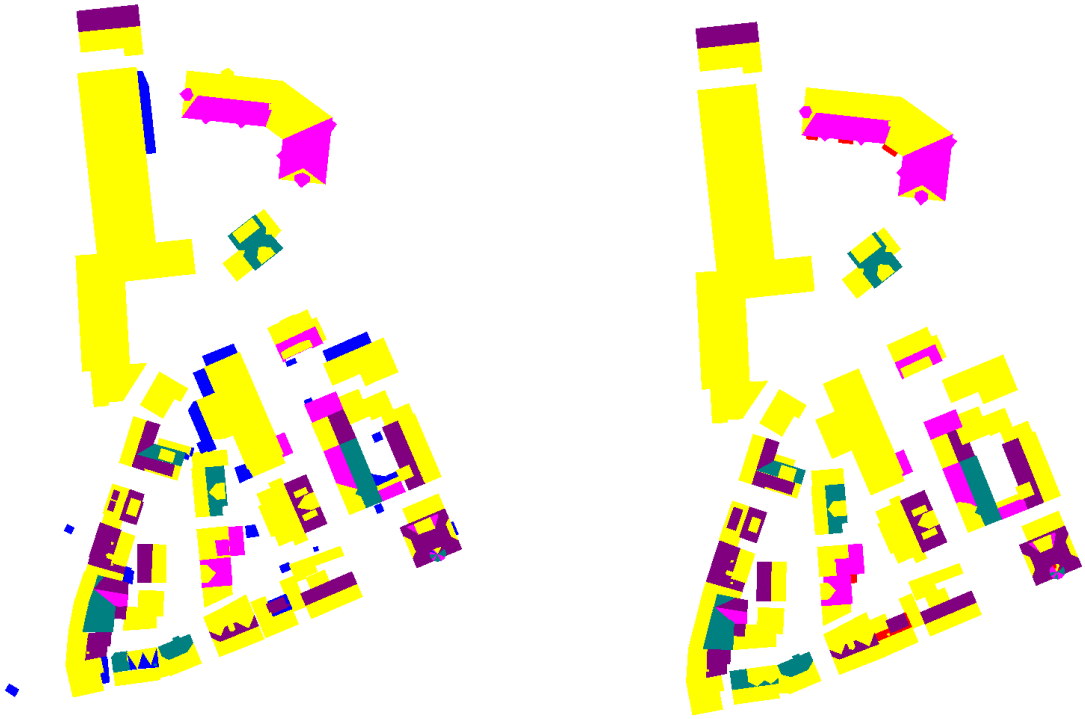


Figure 8.11. Magnified image of a building where details are missing in the reference (left) but not in the result (right).

In order to determine further differences between the roof planes of the reference and the result, topological differences between them were detected. As can be seen in Figure 8.12, most of the roof plane pixels feature a 1:1 relation. However, there are also some roof planes that feature a 1:N, N:1, or N:M relation. 1:N and N:1 relations often indicate that dormers, chimneys, or other small roof superstructures are either missing in the reference or in the result. In contrast, N:M relations often indicate that a building component features a different shape in the result than in the reference so that a sufficient overlap between their roof planes does no longer exist. The reason for most areas that are colored in dark magenta is that some roof superstructures have been considered as outliers either during the segmentation or during the building knowledge derivation procedure and they were thus discarded from the subsequent reconstruction process. Even the proposed ICP based roof superstructure reconstruction approach was not able to detect them due to their low number of points or instances in a single building. The three large roof planes of the curved L-shaped building in the north that are colored in bright magenta indicate that roof planes in the reference were split in the result. As pointed out above, this is caused by the missing balconies in the reference. The fact that also the reference roof planes of the two dormers on top of its hip endings correspond to several roof planes in the result indicates that they were modeled as simple shed dormers in the reference ignoring the fact that each of them is composed of four roof planes with different slopes as in the reconstruction result (see Figure 8.11). An issue of the topologic evaluation that becomes apparent in AOI 1 is the evaluation of non-planar roof surfaces that are approximated by planar roof segments. An example is given for the conical shape of the building part in the southeast. Although it has been correctly detected, its approximation in the result is different from the reference and thus causes several 1:N, N:1, and N:M relations.



Yellow: Pixels inside reference planes having a 1:1 relation to planes in the reconstruction results.

Blue: Pixels inside reference planes that have no corresponding plane in the extraction results.

Dark magenta: Pixels in reference roof planes that are merged with other planes in the reconstruction results (N:1 relation).

Bright magenta: Pixels in reference roof planes split into two or more planes in the reconstruction results (1:N relation).

Dark cyan: Pixels in reference roof planes being part of a plane cluster having N:M relations between planes in the reference and planes in the reconstruction results.

Yellow: Pixels inside reconstructed planes having a 1:1 relation to planes in the reference.

Red: Pixels inside reconstructed planes that have no corresponding plane in the reference.

Dark magenta: Pixels in reconstructed roof planes that correspond to several planes of the reference (1:N relation).

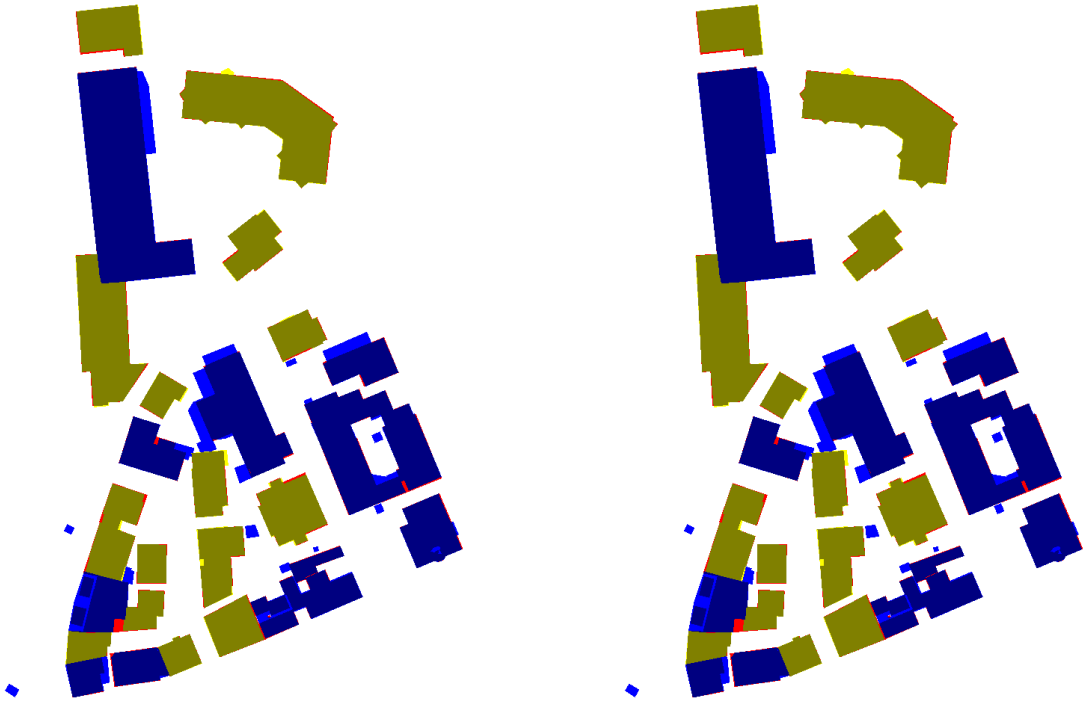
Bright magenta: Pixels in reconstructed roof planes that are the result of a split of a plane in the reconstruction results (N:1 relation).

Dark cyan: Pixels in reconstructed roof planes being part of a plane cluster having N:M relations between planes in the reconstruction results and planes in the reference.

Figure 8.12. Assessment of the topological differences between the reference and the reconstruction result once from the point of view of the reference data (left) and once from the point of view of the detected roof planes (right).

In Figure 8.13, the evaluation result of the completeness is shown. The assessment was carried out once for all roof planes and once for all roof planes larger than 2.5 m². Thereby, it is taken into account that larger building structures are usually easier to reconstruct and that small building structures cause often many differences although they might be less important for many applications. Those buildings that are colored in ochre indicate that all considered reference planes were reconstructed in the result. In contrast, blue colored buildings indicate that some reference roof planes are missing in the result. Since both assessments feature the same result, it can be stated that the differences between the building models in the reference

and the result are not only caused by building structures that are composed of small roof planes.



Ochre: Pixels in buildings for which all reference roof planes were found to have correspondences of sufficient overlap in the reconstruction results, but only if the pixel is inside a building both in the reference and in the reconstruction results.

Yellow: Pixels in buildings for which all reference roof planes were found to have correspondences of sufficient overlap in the reconstruction results; however, these pixels are not inside a building in the reconstruction results.

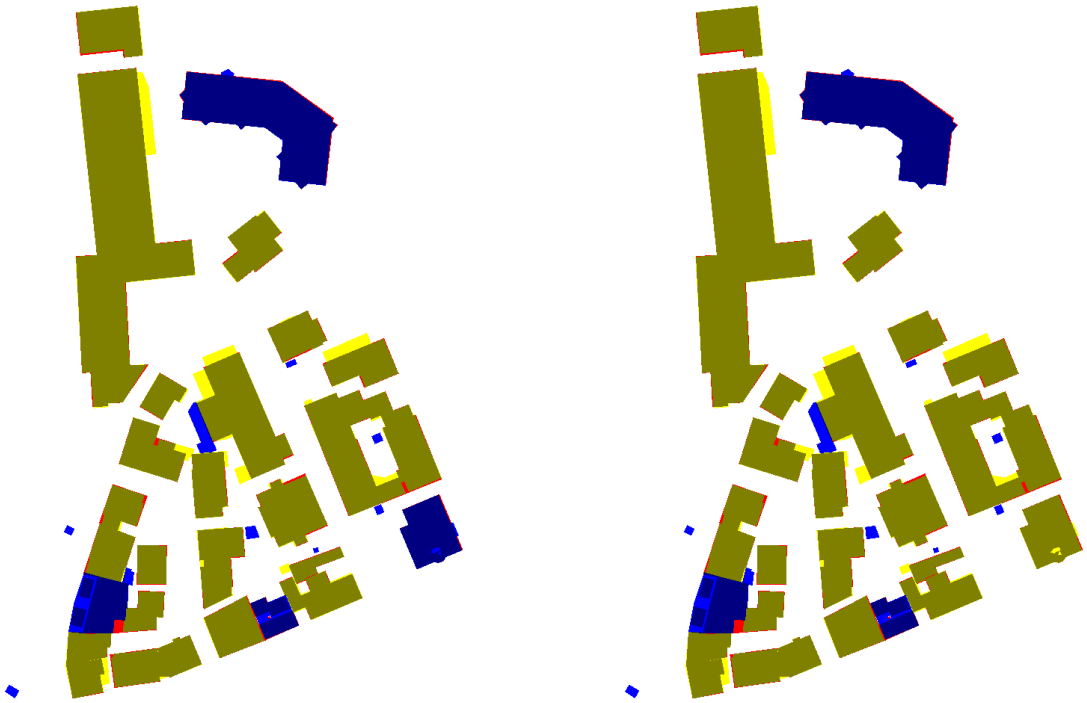
Bright red: Pixels in correctly detected roof planes that are not inside a building in the reference.

Dark blue: Pixels in buildings for which at least one reference roof plane was not detected in the reconstruction results and which are inside a roof plane in the reconstruction results.

Bright blue: Pixels in buildings for which at least one reference roof plane was not detected in the reconstruction results and which are not inside a building in the reconstruction results.

Figure 8.13. Assessment of the completeness once for all detected roof planes (left) and once for roof planes covering an area larger than 2.5 m² (right).

Analogous, two assessments were carried out to evaluate the correctness once for all roof planes and once for all roof planes larger than 2.5 m². The result is presented in Figure 8.14 and shows those buildings in ochre whose automatically reconstructed roof planes are all correct. As can be seen, only three buildings that were reconstructed feature roof planes larger than 2.5 m² that are incorrect with regard to the reference. The other blue colored buildings represent buildings whose points have not been used as input of the automatic reconstruction approach and which are thus not present in the result.



Ochre: Pixels in buildings for which all automatically reconstructed roof planes were found to have correspondences of sufficient overlap in the reference, but only if the pixel is inside a building both in the reference and in the reconstruction results.

Yellow: Pixels in buildings for which all automatically reconstructed roof planes were found to have correspondences of sufficient overlap in the reference; however, these pixels are not inside a building in the reconstruction results.

Bright red: Pixels in correctly detected roof planes that are not inside a building in the reference.

Dark blue: Pixels in buildings for which at least one automatically reconstructed roof plane was not correct and which are inside a roof plane in the reconstruction results.

Bright blue: Pixels in buildings for which at least one automatically reconstructed roof plane was not correct and which are not inside a building in the reconstruction results.

Figure 8.14. Assessment of the correctness once for all detected roof planes (left) and once for roof planes covering an area larger than 2.5 m² (right).

For a further investigation of the completeness, correctness, and quality, the roof plane evaluation was also carried out as a function of the roof plane size. As illustrated in Figure 8.15, correctness is already greater than 95 % if all roof planes are taken into account while completeness and quality become greater than 95 % if only roof planes larger than 55 m² are considered. Thus, large roof planes tend to be well represented in the result while small roof planes cause some problems. Furthermore, the lowest completeness was achieved for roof planes of an area between 55 and 65 m² (80 %), although this would be usually the case for smaller roof planes because they are often more difficult to detect. This, however, is due to the low overall number of roof planes that feature such an area so that the absence of a small number of roof planes has already a great impact. As a consequence, the completeness in the cumulative histogram is only slightly increased if planes of such an area are ignored. In

general, it can be stated that the completeness, correctness, and quality become greater if more small and medium-sized roof planes are ignored in the roof plane evaluation of AOI 1.

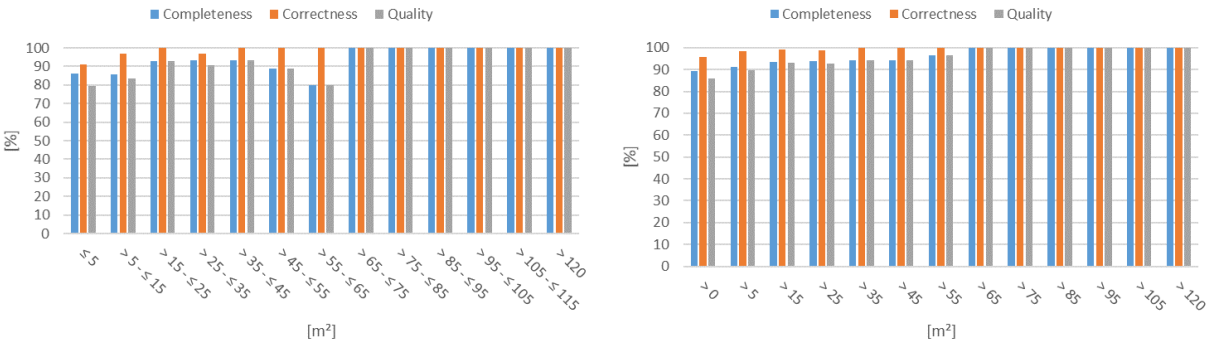


Figure 8.15. Histograms of the roof plane evaluation result as a function of the roof plane size once for area intervals (left) and once for roof planes larger than a certain area (cumulative) (right).

With regard to the geometric accuracy, histograms were calculated once along the reference and once along the result to visualize the RMS error distribution of the roof planes. As shown in Figure 8.16, more than 50 % of the roof planes feature along the reference boundaries and along the result boundaries an RMS error less than 0.3 m. For more than 75 % of the reference and the result roof planes, the RMS error is still less than 0.6 m and 0.8 m, respectively. The median RMS error is thus much lower than the above stated average RMS error (see Table 8.5). The reason for this is that there are some roof planes that have a rather great RMS error. For example, there are four roof planes that feature an RMS error greater than 2 m along the reference and three roof planes that feature an RMS error greater than 2.5 m along the result boundaries. Consequently, it can be stated that most reconstructed roof planes feature a lower RMS error than the aforementioned average RMS error but that there are a few outliers with a great RMS error.

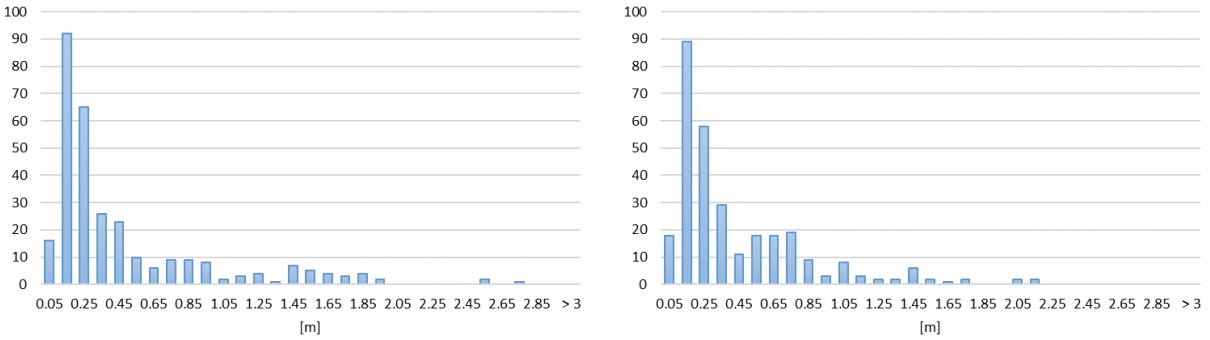


Figure 8.16. Histograms of RMS errors once along reference boundaries (left) and once along result boundaries (right).

In order to illustrate the height differences between the 3D building models of the reference and the result, digital surface models and their differences were calculated as shown in Figure 8.17. Most of the areas feature a height difference less than 0.5 m and thus represent

the building heights very well. However, those building details that were not reconstructed cause height differences larger than 1.2 m which explains the above stated RMS_Z value of around 0.75 m.

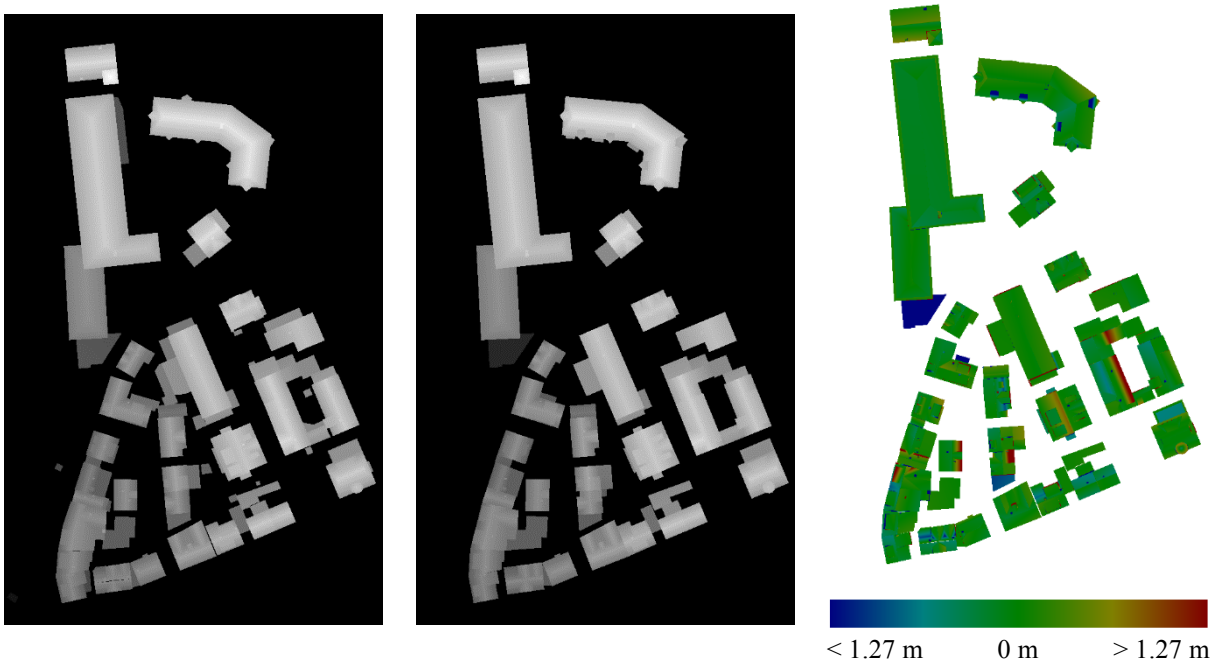


Figure 8.17. Digital surface models derived from the roof planes of the reference (left), from the roof planes of the result (middle), and the difference of these models where planes were found in both data sets (right).

A comparison of the evaluation result to other fully and semi-automatic reconstruction approaches is presented in Table 8.7. As can be seen, the proposed grammar-guided reconstruction approach TUB provides with 89.2 % the highest completeness. The highest correctness was achieved with 100 % by ITCE2 and TUD2. Their high correctness, however, is to the detriment of their completeness, which is only 65.3 % and 73.3 %, respectively. Additionally, their lead in the correctness is further reduced from 4.1 % to 1.4 % if only roof planes larger than 10 m² are considered. Furthermore, roof planes of most other reconstruction approaches feature considerably more frequently N:1 relations than 1:N relations which indicates that under-segmentation might be their major problem. In contrast, 1:N and N:1 relations are balanced in TUB and it features with 22 N:1 relations the lowest value compared to the other approaches so that it can be stated that neither under- nor over-segmentation is a general problem of TUB. A comparison of the RMS_{XY} accuracy reveals that TUB features with 0.6 m the lowest RMS_{XY} error while the average RMS_{XY} error of the other reconstruction approaches is around 0.86 m.

Table 8.7. Evaluation result (TUB) of AOI 1 in comparison to other evaluated (fully/ semi-automatic) reconstruction approaches presented in (ISPRS, 2017).

Abbrev.	Comp [%]	Corr [%]	$Comp_{10}$ [%]	$Corr_{10}$ [%]	Topo 1:N	Topo N:1	Topo N:M	RMS_{XY} [m]
CKU	86.8	98.9	88.4	99.2	10	36	3	0.9
ITCE1	60.8	96.6	64.6	97.2	16	26	17	1.0
ITCE2	65.3	**100.0**	68.0	**100.0**	**0**	38	3	0.9
ITCX1	76.0	99.2	73.5	99.0	2	40	2	0.9
ITCX2	84.7	96.2	87.8	99.2	13	41	6	0.8
ITCX3	**89.2**	96.4	93.2	97.7	5	39	6	0.8
TUD	67.4	96.2	68.0	97.8	1	33	**1**	0.8
VSK	72.2	96.7	80.3	95.9	7	42	6	0.9
YOR	88.2	98.5	94.6	99.2	5	36	14	0.8
MON	76.4	83.3	84.4	84.9	6	42	7	1.0
MON_mod	75.0	95.3	84.4	98.2	9	32	9	1.0
MON2	66.0	91.7	85.7	97.5	17	**22**	11	0.9
TUD2	73.3	**100.0**	70.7	**100.0**	1	36	3	0.8
MEL_HE	88.2	99.5	**95.2**	99.2	3	40	2	0.7
BNU2	84.7	99.3	89.8	99.2	2	42	3	0.8
MON5	74.3	98.7	89.8	**100.0**	15	32	10	0.8
TUB	**89.2**	95.9	92.6	98.6	24	**22**	14	**0.6**

In summary, the proposed grammar-guided reconstruction approach is generally suitable to reconstruct historic buildings with complex shapes. However, the automatic reconstruction of small roof structures is still challenging, especially if their number of instances within a building is low.

8.3.2 Vaihingen: AOI 2

AOI 2 comprises 14 high-rising residential buildings with mixed flat and sloped roof types. The pixel-based building detection result shown in Figure 8.18 reveals that all large buildings are represented in the result. There are only four small buildings that were not automatically reconstructed. The reason for this is that their points were assigned to the class of non-building points during the classification. Consequently, these points were not used as input points of the proposed reconstruction approach why they are not represented in the result. Further differences regarding the automatically reconstructed building models and their references are highlighted in Figure 8.19. As can be seen, there is an attached building part missing in the south. This area was also not classified as building because the height to the terrain is rather low for a building and by taking the images into account, it seems to be a garden surrounded by a wall. Since the overall number of buildings in this AOI is with 14

rather low compared to the other AOIs, these missing areas have a strong impact on the evaluation result. Further false negative or false positive pixels are analogous to AOI 1 located close to the building outlines. Those that feature small rectangular shapes essentially represent canopies whose points were not assigned to the building class during the classification procedure. The small rectangular ensemble of red colored pixels in the northeast are due to a reconstructed balcony in the result that is missing in the reference. Other false negative and false positive essentially represent slight shifts or rotations of building facades.

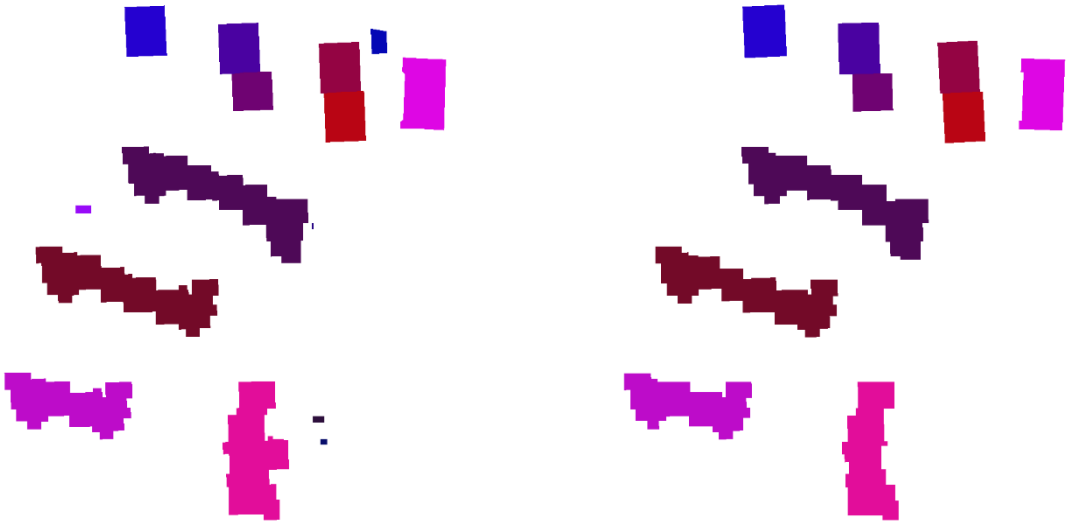


Figure 8.18. Pixel-based building detection result of the building models in the reference (left) and in the result (right).

Figure 8.19. Evaluation result of the pixel-based building detection result (yellow: true positive pixels, blue: false negative pixels, and red: false positive pixels).

The roof plane detection result of the reference and the result reveals that the roof planes of high-rising residential buildings with mixed flat and sloped roof types are generally well represented in the result (see Figure 8.20). This applies even for the chimney and the small dormers on the building in the northeast. Additionally, the evaluation result of the roof plane detection result in Figure 8.21 shows that there is only one area from the point of view of the reference that is located within a building of the result and that features an insufficient overlap. This area consists of a horizontal and a sloped roof plane which were differently reconstructed in the result so that they feature an insufficient overlap with the reference roof planes. The reconstruction of this kind of roof structure is, however, not a general problem of the proposed reconstruction approach. Similar roof structures are present in the two buildings that are located in the southwest and they were for both buildings successfully reconstructed with sufficient overlap to the reference roof planes.

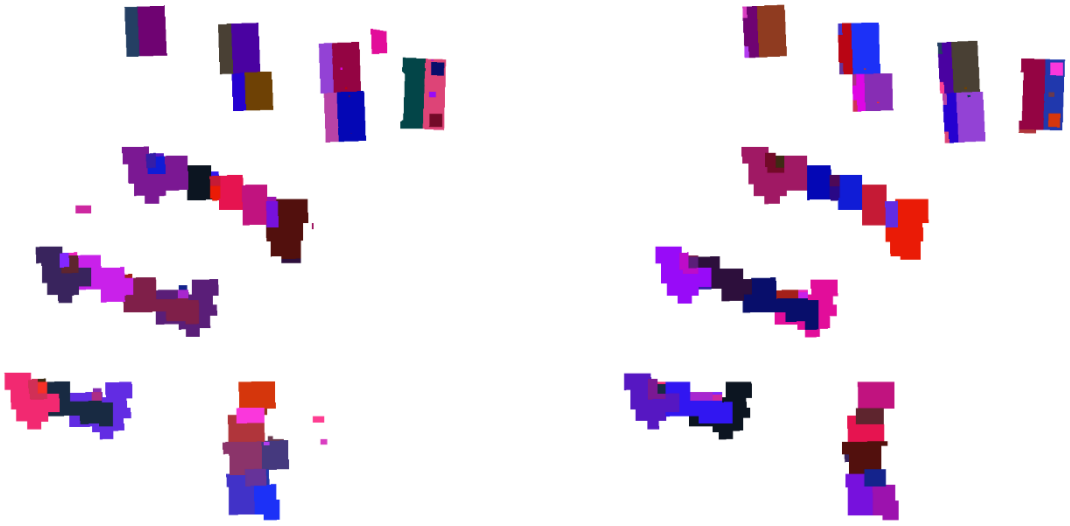


Figure 8.20. Roof plane detection result of the building models in the reference (left) and in the result (right).

Ochre: Pixels available in both the reconstruction results and the reference and belonging to a reference plane found to have sufficient overlap with planes in the roof reconstruction results.

Yellow: Pixels that belong to a reference plane found to have sufficient overlap with planes in the roof reconstruction results but that are not inside a building in the reconstruction results.

Dark blue: Pixels in reference roof planes not having sufficient overlap with detected roof planes and not being inside a building in the reconstruction results.

Bright blue: Pixels in reference roof planes not having sufficient overlap with detected roof planes, but being inside a building in the reconstruction results.

Ochre: Pixels available in both the reconstruction results and the reference and belonging to a detected roof plane found to have sufficient overlap with planes in the reference.

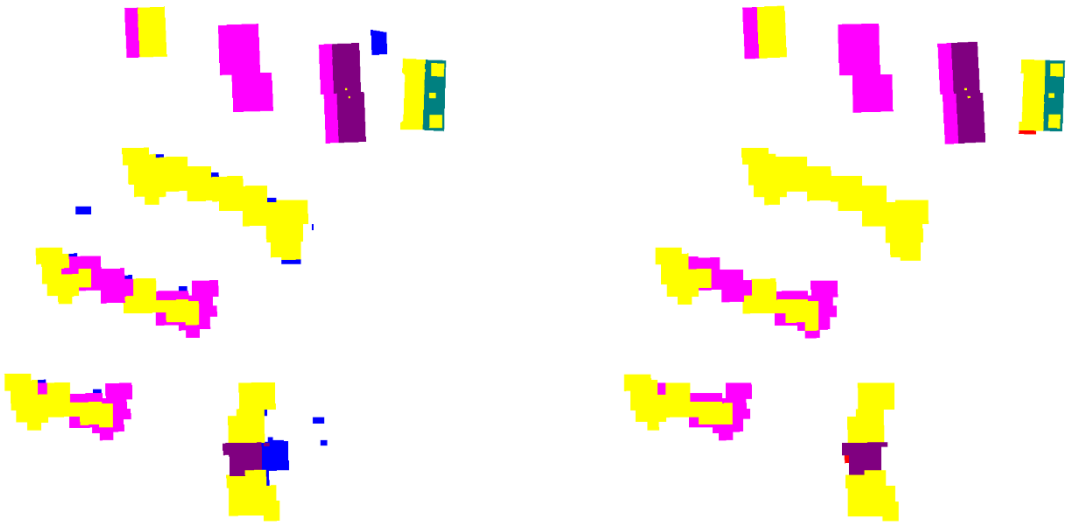
Yellow: Pixels that belong to a detected roof plane found to have sufficient overlap with planes in the reference but that are not inside a building in the reference.

Dark red: Pixels in detected roof planes not having sufficient overlap with roof planes in the reference and not being inside a building in the reference.

Bright red: Pixels in detected roof planes not having sufficient overlap with reference planes, but being inside a building in the reference.

Figure 8.21. Evaluation result of the roof plane detection result once from the point of view of the reference (left) and once from the point of view of the result (right).

Although the evaluation result of the roof plane detection result indicates a high correspondence between the roof planes in the reference and the result, the assessment of their topological differences shows several discrepancies (see Figure 8.22). Generally, it can be stated that the number of 1:N relations is from the point of view of the reference dominant in AOI 2. For example, the five roof planes on the left of the saddleback buildings in the north feature a 1:N relation. The reason that the number of roof planes is larger in the result than in the reference is that roof terraces were reconstructed in the result as vertical intrusions of the roofs but they are missing in the reference (see Figure 8.23). The same reason, however, does not apply on the buildings with complex flat rooftops in the southwest. These 1:N relations are essentially caused by over-segmentation. Contrary, under-segmentation it the reason why the roof plane of the building in the south is colored in dark magenta. As a result, a chimney is missing in the automatically reconstructed building model.



Yellow: Pixels inside reference planes having a 1:1 relation to planes in the reconstruction results.

Blue: Pixels inside reference planes that have no corresponding plane in the extraction results.

Dark magenta: Pixels in reference roof planes that are merged with other planes in the reconstruction results (N:1 relation).

Bright magenta: Pixels in reference roof planes split into two or more planes in the reconstruction results (1:N relation).

Dark cyan: Pixels in reference roof planes being part of a plane cluster having N:M relations between planes in the reference and planes in the reconstruction results.

Yellow: Pixels inside reconstructed planes having a 1:1 relation to planes in the reference.

Red: Pixels inside reconstructed planes that have no corresponding plane in the reference.

Dark magenta: Pixels in reconstructed roof planes that correspond to several planes of the reference (1:N relation).

Bright magenta: Pixels in reconstructed roof planes that are the result of a split of a plane in the reconstruction results (N:1 relation).

Dark cyan: Pixels in reconstructed roof planes being part of a plane cluster having N:M relations between planes in the reconstruction results and planes in the reference.

Figure 8.22. Assessment of the topological differences between the reference and the reconstruction result once from the point of view of the reference data (left) and once from the point of view of the detected roof planes (right).

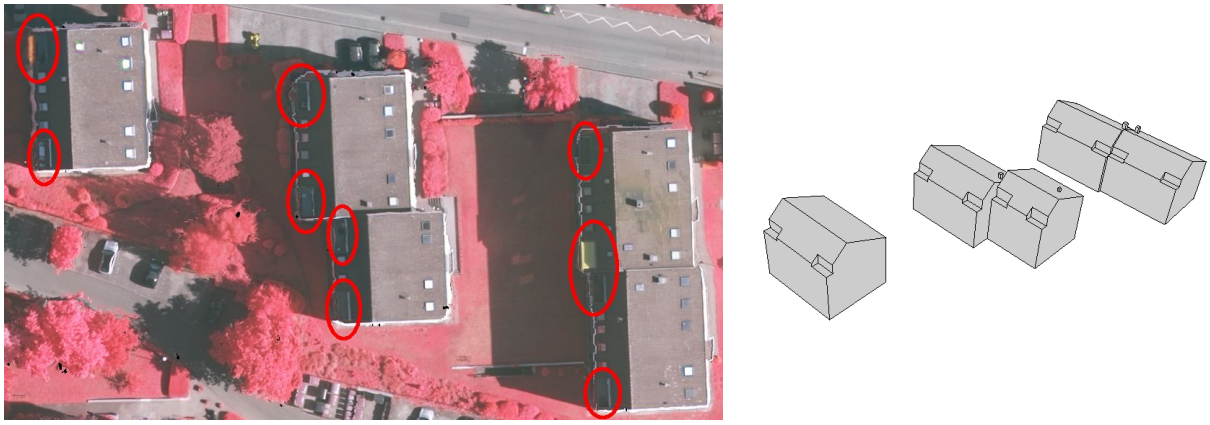


Figure 8.23. Magnified image of buildings where roof terraces are missing in the reference (left) but not in the result (right).

The assessment of the completeness is presented in Figure 8.24. As can be seen, all saddle-back roofs in the north were completely reconstructed whereas the other buildings, which feature complex flat rooftop structures, are missing some building parts. This is essentially due to the aforementioned points that were classified as non-building or due to missing chimneys in the result. Particularly the former reason explains why these buildings are still incomplete if only roof planes larger than 2.5 m² are considered. The great number of incomplete buildings explains furthermore the low completeness value of 72.5 % for AOI 2 (see Table 8.5).

Ochre: Pixels in buildings for which all reference roof planes were found to have correspondences of sufficient overlap in the reconstruction results, but only if the pixel is inside a building both in the reference and in the reconstruction results.

Yellow: Pixels in buildings for which all reference roof planes were found to have correspondences of sufficient overlap in the reconstruction results; however, these pixels are not inside a building in the reconstruction results.

Bright red: Pixels in correctly detected roof planes that are not inside a building in the reference.

Dark blue: Pixels in buildings for which at least one reference roof plane was not detected in the reconstruction results and which are inside a roof plane in the reconstruction results.

Bright blue: Pixels in buildings for which at least one reference roof plane was not detected in the reconstruction results and which are not inside a building in the reconstruction results.

Figure 8.24. Assessment of the completeness once for all detected roof planes (left) and once for roof planes covering an area larger than 2.5 m² (right).

The presented assessment of the correctness in Figure 8.25 reveals that a large majority of automatically reconstructed buildings feature roof planes of sufficient overlap in the reference. However, there is one building in the result which features at least one automatically reconstructed roof planes of an area greater than 2.5 m² with insufficient overlap to the reference roof planes. The low number of automatically reconstructed roof plains with insufficient overlap to a reference roof plain explains the high correctness value of 97.1 % for AOI 2.

Ochre: Pixels in buildings for which all automatically reconstructed roof planes were found to have correspondences of sufficient overlap in the reference, but only if the pixel is inside a building both in the reference and in the reconstruction results.

Yellow: Pixels in buildings for which all automatically reconstructed roof planes were found to have correspondences of sufficient overlap in the reference; however, these pixels are not inside a building in the reconstruction results.

Bright red: Pixels in correctly detected roof planes that are not inside a building in the reference.

Dark blue: Pixels in buildings for which at least one automatically reconstructed roof plane was not correct and which are inside a roof plane in the reconstruction results.

Bright blue: Pixels in buildings for which at least one automatically reconstructed roof plane was not correct and which are not inside a building in the reconstruction results.

Figure 8.25. Assessment of the correctness once for all detected roof planes (left) and once for roof planes covering an area larger than 2.5 m² (right).

In order to determine the impact of the roof plane area on the completeness, correctness, and quality, the roof plane evaluation were also carried out as a function of the roof plane size. As illustrated in Figure 8.26, completeness and quality are analogous to AOI 1 greater than 95 % if only roof planes larger than 55 m² are considered whereas the correctness is already greater than 95 % if all roof planes are taken into account. It can be furthermore stated that the greatest challenge for the proposed reconstruction approach was to detect and reconstruct small roof planes that feature an area less or equal than 15 m². However, there are also some roof planes of greater area (between 45 m² and 55 m² or between 85 m² and 95 m²) that were not automatically reconstructed.

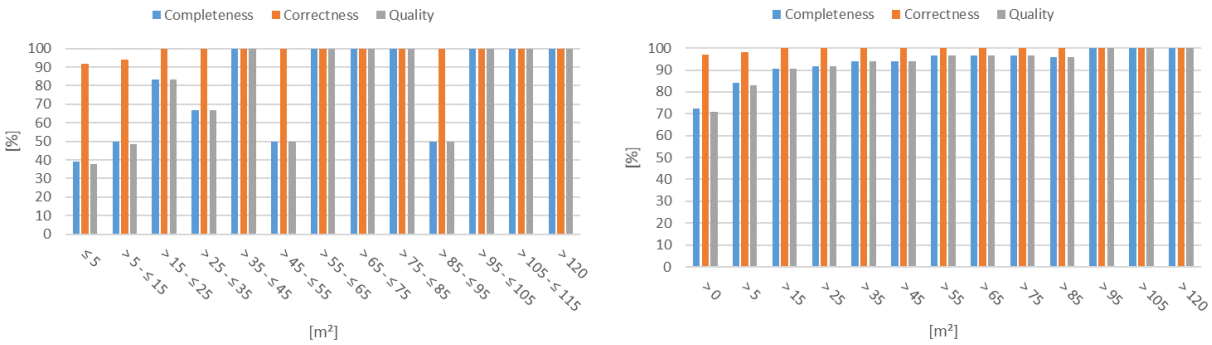


Figure 8.26. Histograms of the roof plane evaluation result as a function of the roof plane size once for area intervals (left) and once for roof planes larger than a certain area (cumulative) (right).

The two histograms of RMS errors shown in Figure 8.27 reveal that more than 50 % of the roof planes feature along the reference boundaries and along the result boundaries an RMS error less than 0.4 m and 0.3 m, respectively. Additionally, there are only a few planes with an RMS error greater than 1 m why the median RMS error becomes much lower than the above stated average RMS error (see Table 8.5). For more than 75 % of the reference and the result roof planes, the RMS error is still less than 0.8 m and 0.9 m, respectively. Consequently, it can be stated that most reconstructed roof planes feature a lower RMS error than the average RMS error but that there are also a few outliers with a rather large RMS error compared to other reconstructed roof planes.

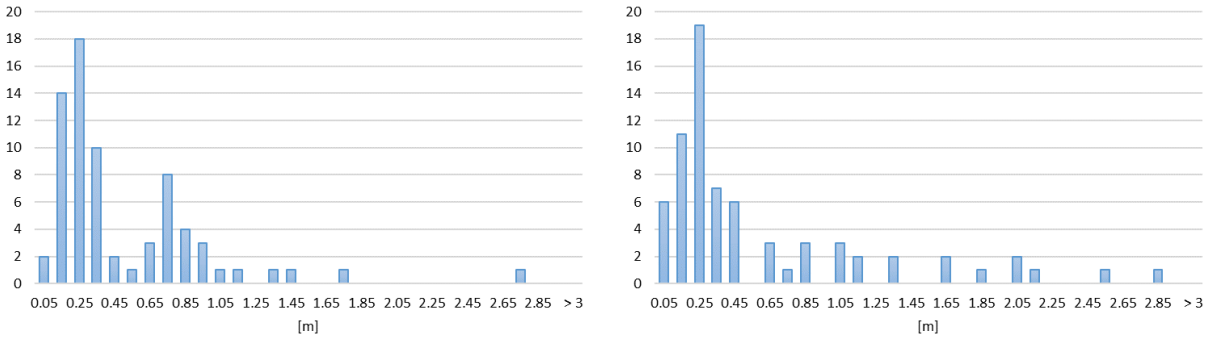


Figure 8.27. Histograms of RMS errors once along reference boundaries (left) and once along result boundaries (right).

The height differences between roof planes in the reference and in the result are presented in Figure 8.28. It shows that most automatically reconstructed areas feature a height difference less than 0.25 m. The roof planes of the result thus represent the building heights of the reference very well. However, missing details in the reference (e.g., balconies) and the adjustment of facades to support natural building regularities cause in some areas height differences larger than 1.2 m. The latter becomes particularly apparent in buildings that feature complex flat rooftop structures.

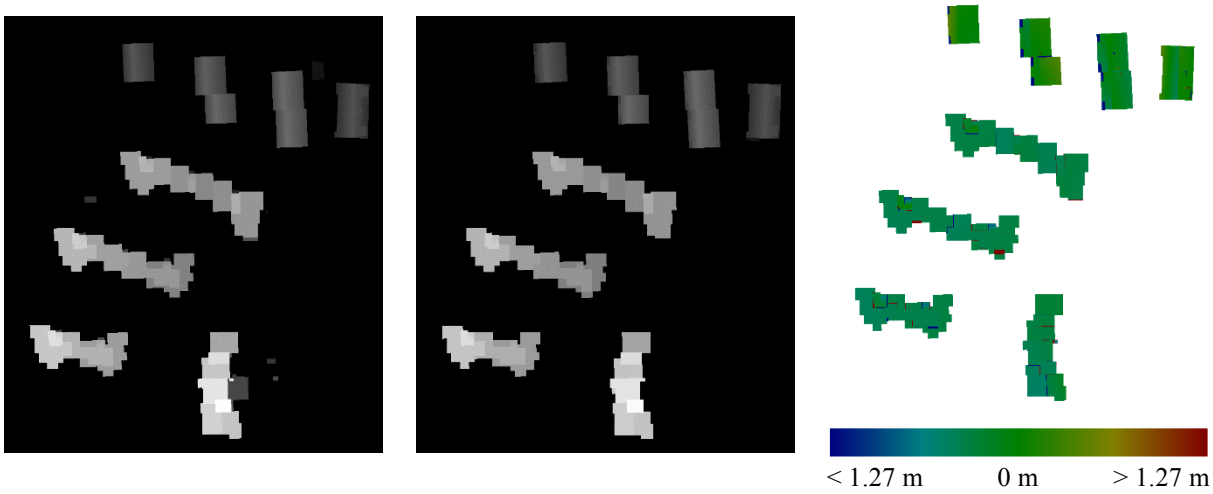


Figure 8.28. Digital surface models derived from the roof planes of the reference (left), from the roof planes of the result (middle), and the difference of these models where planes were found in both data sets (right).

A comparison of the evaluation result to other fully and semi-automatic reconstruction approaches is presented in Table 8.8. It has to be considered that the number of buildings in this area is lower than in the other AOIs so that a change of only a few roof planes might have a visible change in the evaluation result. As can be seen, the completeness of the proposed reconstruction approach TUB is only on average for AOI 2 which is basically due to misclassified building points. However, it is still competitive for both Comp and $Comp_{10}$ if ITCE1 and ITCE2, which feature significantly high completeness values but also significantly high RMS_{XY} error values compared to all other approaches, are excluded. Analogous to other reconstruction approaches (e.g., ITCX2, Mon2, Mon5, etc.), over-segmentation could be assumed to be the major segmentation issue of TUB in AOI 2. However, the high number of 1:N relations is caused by several balconies that were reconstructed in the result but that are not present in the reference. If they are ignored, then the 1:N relations are not dominant anymore compared to the N:1 and N:M relations. With the exception of ITCE1 and ITCE2, all reconstruction approaches have a high geometric accuracy in x-y direction of less or equal than twice of the point spacing. Therefore, it is not surprising that TUB features an RMS_{XY} value that is only slightly better than the average.

Table 8.8. Evaluation result (TUB) of AOI 2 in comparison to other evaluated (fully/ semi-automatic) reconstruction approaches presented in (ISPRS, 2017).

Abbrev.	Comp [%]	Corr [%]	$Comp_{10}$ [%]	$Corr_{10}$ [%]	Topo 1:N	Topo N:1	Topo N:M	RMS_{XY} [m]
CKU	78.3	93.1	93.8	**100.0**	8	4	**0**	0.5
ITCE1	**79.7**	73.7	**97.9**	73.7	**0**	7	**0**	1.2
ITCE2	**79.7**	95.0	**97.9**	**100.0**	**0**	7	**0**	1.2
ITCX1	62.3	95.1	77.1	94.9	2	4	**0**	0.5
ITCX2	75.4	98.2	91.7	**100.0**	15	3	1	0.5
ITCX3	71.0	**100.0**	89.6	**100.0**	3	4	1	0.5
TUD	68.1	98.1	85.4	**100.0**	5	3	**0**	0.6
VSK	73.9	**100.0**	91.7	**100.0**	3	5	1	0.7
YOR	66.7	**100.0**	83.3	**100.0**	5	3	**0**	0.5
CAS	63.8	**100.0**	79.2	**100.0**	3	3	**0**	0.7
MON	73.9	91.9	93.8	92.6	7	3	1	0.7
MON_mod	69.6	96.8	85.4	**100.0**	8	3	2	0.8
MON2	71.0	90.7	85.4	**100.0**	11	2	**0**	0.7
TUD2	71.0	**100.0**	89.6	**100.0**	2	3	**0**	**0.3**
MEL_HE	71.0	98.1	91.7	**100.0**	2	2	**0**	0.6
BNU2	73.9	**100.0**	93.8	**100.0**	2	9	**0**	0.5
MON5	72.5	94.8	89.6	**100.0**	27	**1**	2	0.7
TUB	72.5	97.1	91.5	**100.0**	14	3	1	0.6

In summary, the proposed grammar-guided reconstruction approach is generally suitable to reconstruct high-rising residential buildings with mixed flat and sloped roof types. However, some difficulties have been encountered on the automatic reconstruction of all details belonging to large flat rooftops that feature a complex agglomeration of several roof planes.

8.3.3 Vaihingen: AOI 3

AOI 3 comprises with 56 buildings the largest number of buildings of all AOIs in the Vaihingen test site. They are located in a purely residential area and most of them feature either a flat or a saddleback roof with small dormers or attached canopies. The pixel-based building detection result presented in Figure 8.29 shows that most of the buildings are represented in the result. A closer look reveals, however, that several small buildings were not automatically reconstructed at all. In total, there a 12 buildings missing in the result of which most of them represent arbors or gazebos. As it is typically for these kind of buildings, they are, on the one hand, represented by only a small number of points and, on the other hand, surrounded by vegetation that covers and hides several parts. Some of the missing buildings are, therefore, only recognizable in the images, which were used to manually reconstruct the

reference buildings, but not in the ALS data. The classification of their points as building points is thus rather challenging which explains why all points of the missing buildings were classified as non-building points so that they have not been incorporated in the automatic reconstruction process.

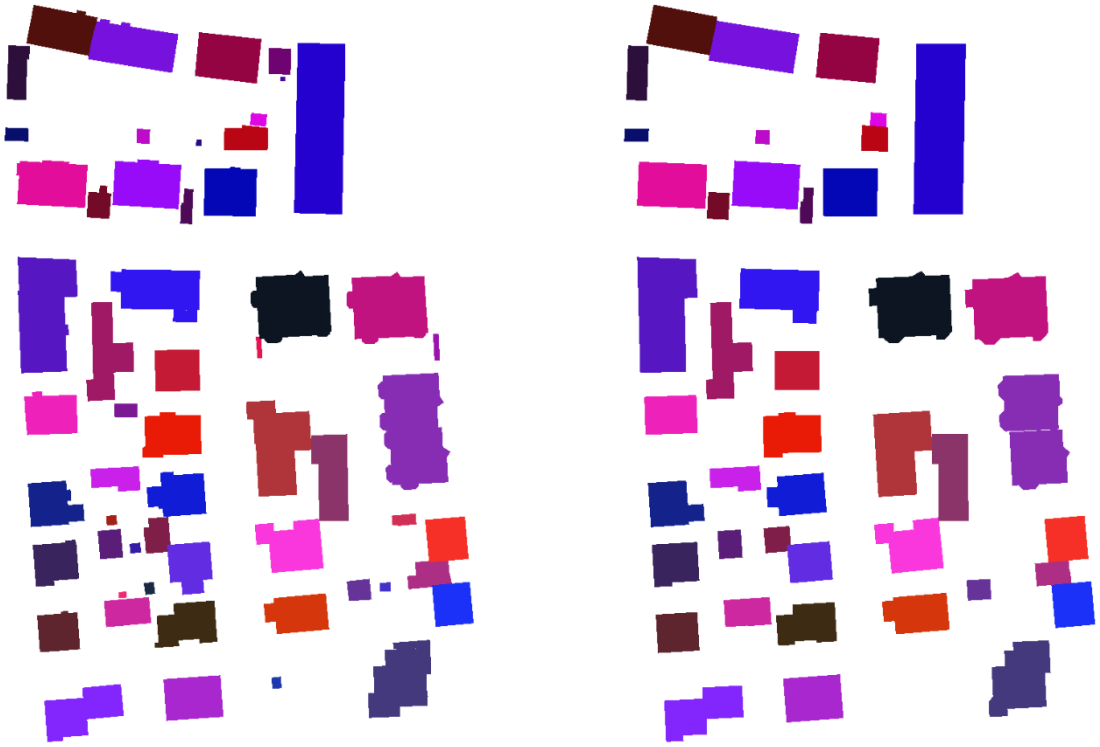


Figure 8.29. Pixel-based building detection result of the building models in the reference (left) and in the result (right).

Further differences in the pixel-based building detection result become visible in Figure 8.30. Similar to AOI 1 and AOI 2, they are mainly close to building facades and caused either by the resolution of the point cloud, which does not allow to precisely detect them, or by the adjustment step of the automatic reconstruction approach, which tries to support natural regularities. Additionally, there were also two small areas classified as false positive that are attached to the building in the north. Both areas represent awnings which are visible in the point cloud but missing in the manually reconstructed reference.

Figure 8.30. Evaluation result of the pixel-based building detection result (yellow: true positive pixels, blue: false negative pixels, and red: false positive pixels).

The roof plane detection result of the reference and the result is presented in Figure 8.31. As can be seen, the majority of automatically reconstructed rooftops feature the same roof surface distribution as their corresponding rooftops in the reference. A closer look on the roof plane detection result reveals, however, that several small roof planes that consist of only a few pixels are not present in the result. They essentially represent chimneys that can be only hardly recognized in the ALS data or that are only visible in the images. In contrast, larger roof planes in buildings that have been reconstructed are well represented in the result.

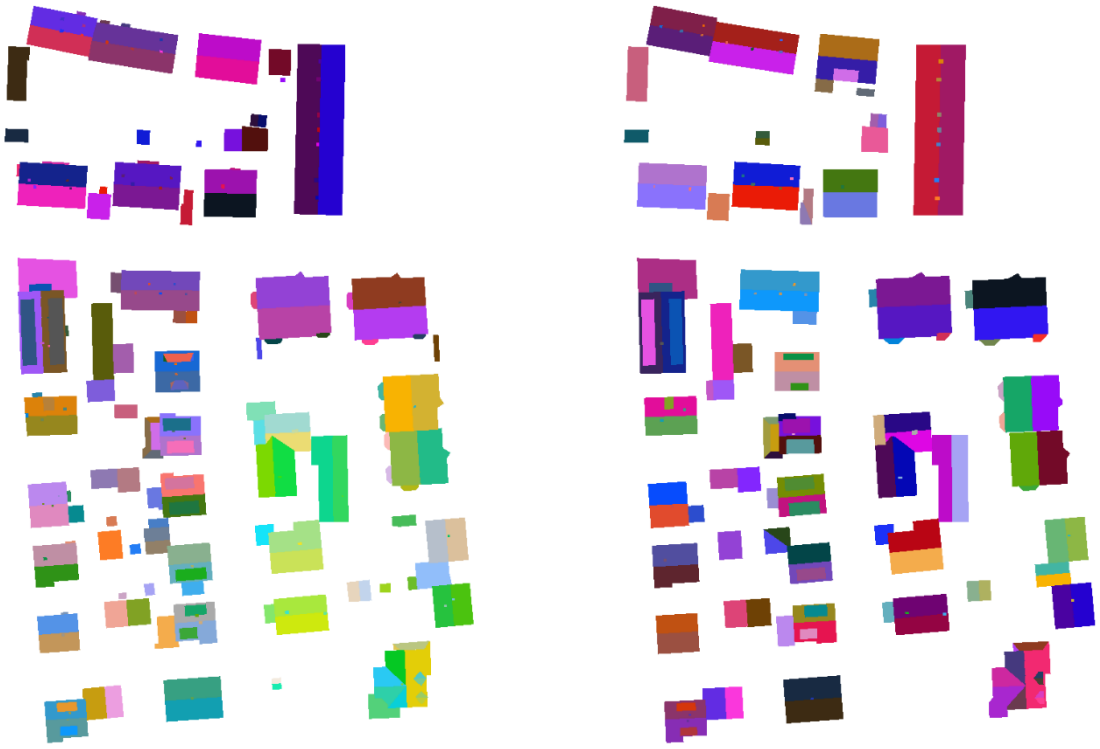
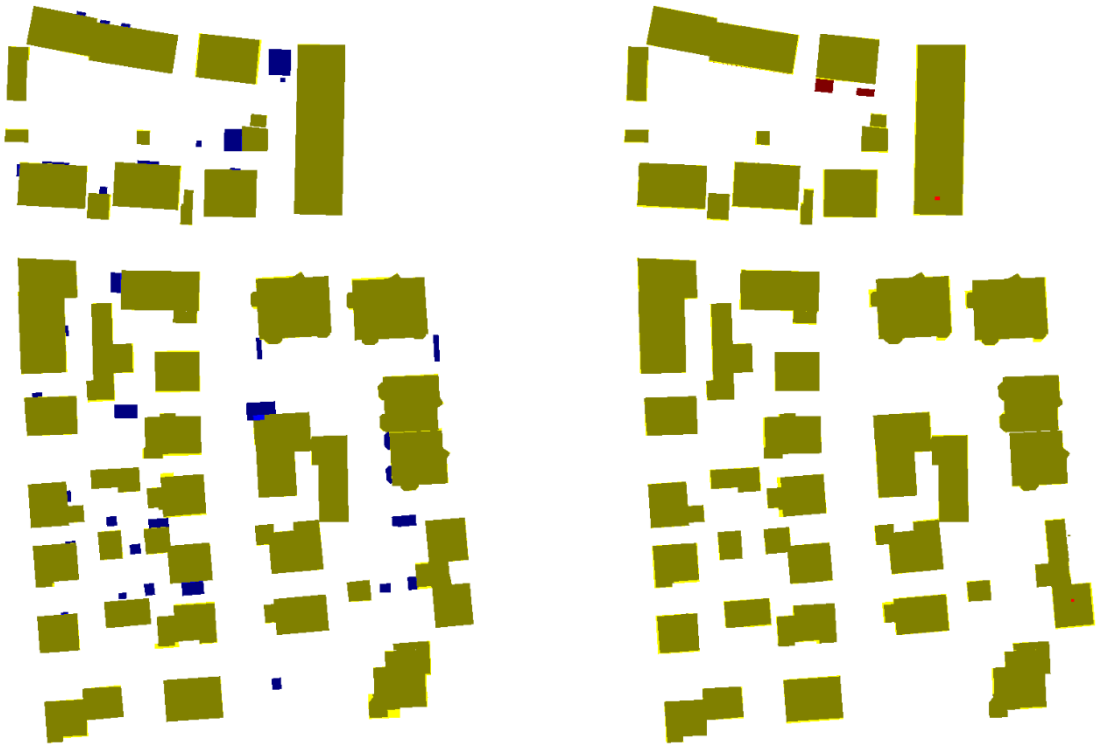


Figure 8.31. Roof plane detection result of the building models in the reference (left) and in the result (right).

As illustrated in Figure 8.32, a large majority of the reconstructed areas have pixels that feature a sufficient overlap between roof planes in the reference and in the result. The previously made statement that most automatically reconstructed rooftops feature the same roof surface distribution as their corresponding rooftops in the reference is thus confirmed by the evaluation result of the roof plane detection result. Furthermore, it can be seen that from the point of view of the reference some canopies are missing in the result in addition to the above mentioned missing arbors and gazebos. This is particularly the case for buildings on the western side of AOI 3. Contrary, there are only two roof planes in the result within the buildings of the reference that have no corresponding reference roof plane. As the small areas already imply, these roof planes belong to chimneys that are missing in the reference. Their presence can be clearly seen in the images but they are only hardly to detect in the point cloud. The proposed ICP based reconstruction of small roof superstructures, however, was able to correctly identify and to reconstruct the missing chimney in the northeast because there are six other chimneys on the same rooftop that feature a similar shape. The same also applies to the missing chimney in the southeast.



Ochre: Pixels available in both the reconstruction results and the reference and belonging to a reference plane found to have sufficient overlap with planes in the roof reconstruction results.

Yellow: Pixels that belong to a reference plane found to have sufficient overlap with planes in the roof reconstruction results but that are not inside a building in the reconstruction results.

Dark blue: Pixels in reference roof planes not having sufficient overlap with detected roof planes and not being inside a building in the reconstruction results.

Bright blue: Pixels in reference roof planes not having sufficient overlap with detected roof planes, but being inside a building in the reconstruction results.

Ochre: Pixels available in both the reconstruction results and the reference and belonging to a detected roof plane found to have sufficient overlap with planes in the reference.

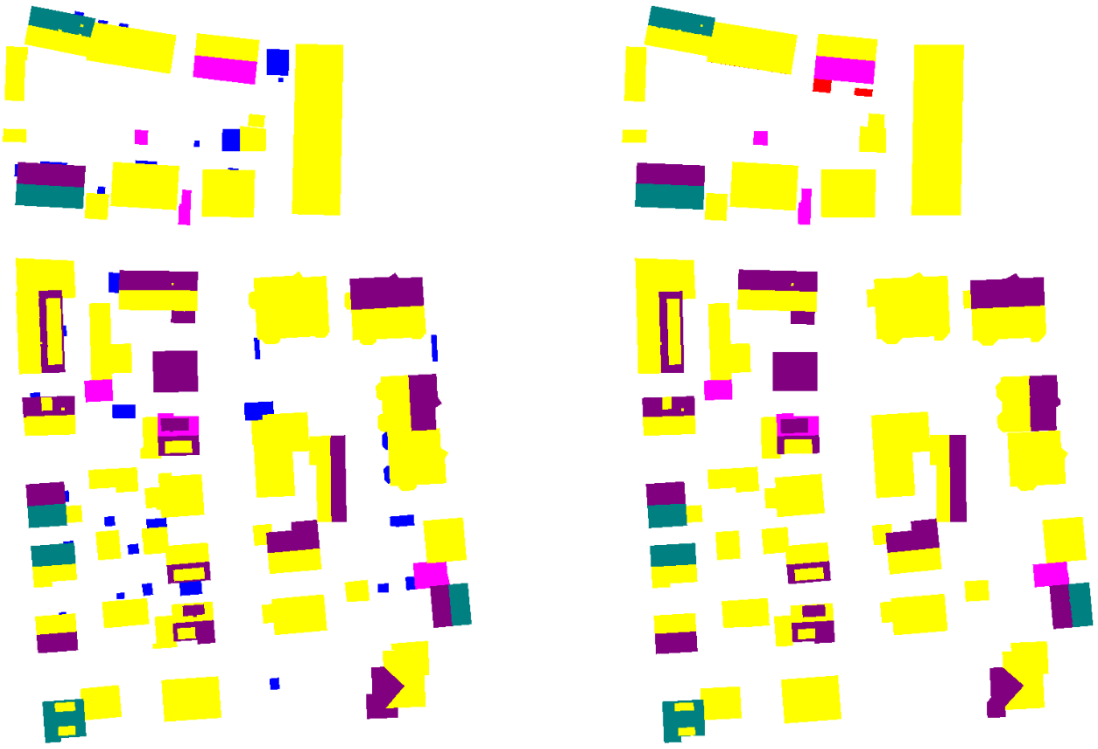
Yellow: Pixels that belong to a detected roof plane found to have sufficient overlap with planes in the reference but that are not inside a building in the reference.

Dark red: Pixels in detected roof planes not having sufficient overlap with roof planes in the reference and not being inside a building in the reference.

Bright red: Pixels in detected roof planes not having sufficient overlap with reference planes, but being inside a building in the reference.

Figure 8.32. Evaluation result of the roof plane detection result once from the point of view of the reference (left) and once from the point of view of the result (right).

The assessment of the topological differences presented in Figure 8.33 shows that 1:1 relations are dominant in AOI 3 but that there are also several roof planes in the reference that have been merged to a single roof plane in the result. Most of them are caused by chimneys which are either not recognizable in the ALS data or represented by only a few points. Consequently it can be said that regarding chimneys, the problem of under-segmentation is in AOI 3 particularly apparent. Also the ICP based roof superstructure reconstruction approach was not able to recognize them because their number of instances on a single building is too low or often only one.



Yellow: Pixels inside reference planes having a 1:1 relation to planes in the reconstruction results.

Blue: Pixels inside reference planes that have no corresponding plane in the extraction results.

Dark magenta: Pixels in reference roof planes that are merged with other planes in the reconstruction results (N:1 relation).

Bright magenta: Pixels in reference roof planes split into two or more planes in the reconstruction results (1:N relation).

Dark cyan: Pixels in reference roof planes being part of a plane cluster having N:M relations between planes in the reference and planes in the reconstruction results.

Yellow: Pixels inside reconstructed planes having a 1:1 relation to planes in the reference.

Red: Pixels inside reconstructed planes that have no corresponding plane in the reference.

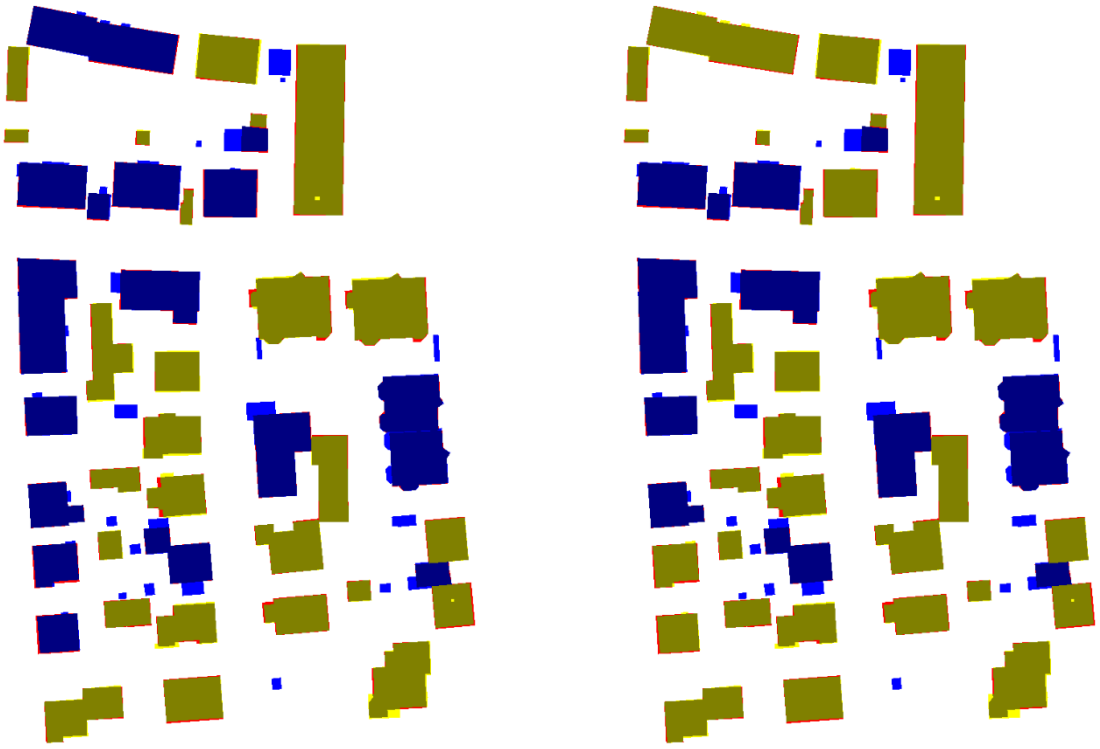
Dark magenta: Pixels in reconstructed roof planes that correspond to several planes of the reference (1:N relation).

Bright magenta: Pixels in reconstructed roof planes that are the result of a split of a plane in the reconstruction results (N:1 relation).

Dark cyan: Pixels in reconstructed roof planes being part of a plane cluster having N:M relations between planes in the reconstruction results and planes in the reference.

Figure 8.33. Assessment of the topological differences between the reference and the reconstruction result once from the point of view of the reference data (left) and once from the point of view of the detected roof planes (right).

The assessment of the completeness shown in Figure 8.34 reveals that there are in total 18 reference buildings with a corresponding building in the result that feature at least one reference roof reference plane that is missing in the result. As previously stated, this large number is essentially due to missing chimneys and canopies. Therefore, their number is already decreased from 18 to 13 if only roof planes larger than 2.5 m² are considered. In order to examine also the impact of missing canopies on the completeness, a further investigation on the roof plane evaluation is carried out later as a function of the roof plane size.



Ochre: Pixels in buildings for which all reference roof planes were found to have correspondences of sufficient overlap in the reconstruction results, but only if the pixel is inside a building both in the reference and in the reconstruction results.

Yellow: Pixels in buildings for which all reference roof planes were found to have correspondences of sufficient overlap in the reconstruction results; however, these pixels are not inside a building in the reconstruction results.

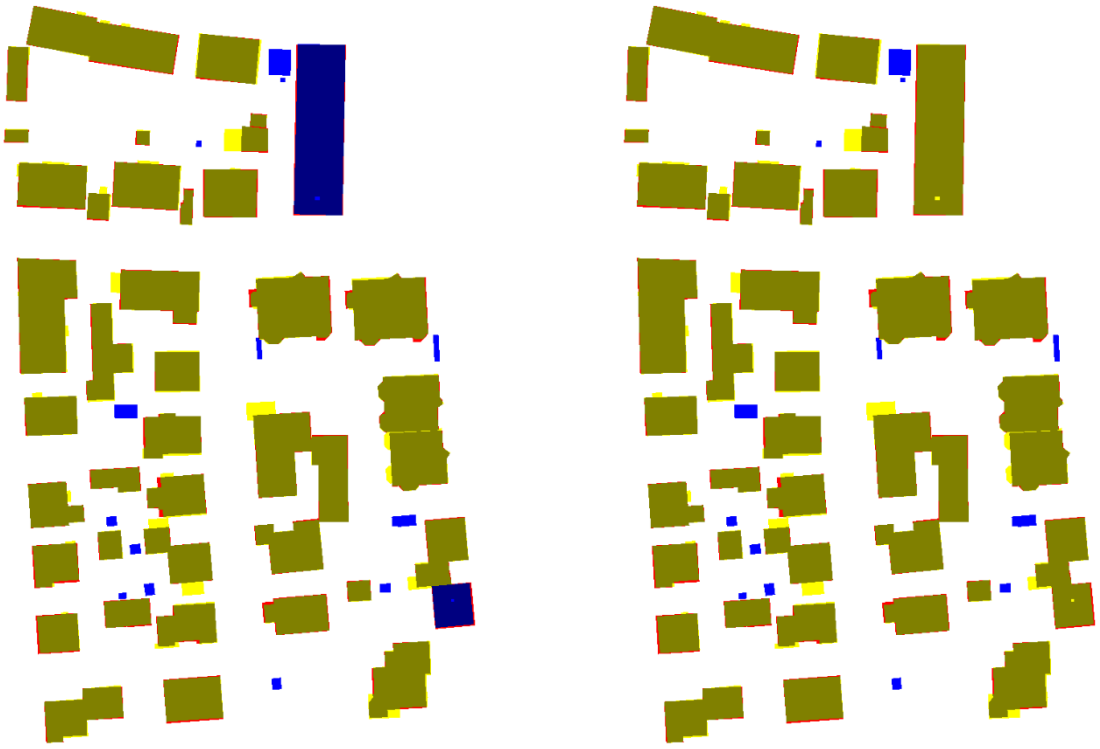
Bright red: Pixels in correctly detected roof planes that are not inside a building in the reference.

Dark blue: Pixels in buildings for which at least one reference roof plane was not detected in the reconstruction results and which are inside a roof plane in the reconstruction results.

Bright blue: Pixels in buildings for which at least one reference roof plane was not detected in the reconstruction results and which are not inside a building in the reconstruction results.

Figure 8.34. Assessment of the completeness once for all detected roof planes (left) and once for roof planes covering an area larger than 2.5 m² (right).

In contrast to the completeness, the correctness of the automatically reconstructed roof planes is significantly higher as shown in Figure 8.35. A great majority of the pixels in the automatically reconstructed buildings were found to have correspondences of sufficient overlap in the reference. There are only two buildings in the result whose roof planes have not all a corresponding roof plane in the reference. As discussed before, missing chimneys in the reference are in both cases the reason. This statement is supported by the fact that corresponding roof planes were found for all automatically reconstructed roof planes larger than 2.5 m².



Ochre: Pixels in buildings for which all automatically reconstructed roof planes were found to have correspondences of sufficient overlap in the reference, but only if the pixel is inside a building both in the reference and in the reconstruction results.

Yellow: Pixels in buildings for which all automatically reconstructed roof planes were found to have correspondences of sufficient overlap in the reference; however, these pixels are not inside a building in the reconstruction results.

Bright red: Pixels in correctly detected roof planes that are not inside a building in the reference.

Dark blue: Pixels in buildings for which at least one automatically reconstructed roof plane was not correct and which are inside a roof plane in the reconstruction results.

Bright blue: Pixels in buildings for which at least one automatically reconstructed roof plane was not correct and which are not inside a building in the reconstruction results.

Figure 8.35. Assessment of the correctness once for all detected roof planes (left) and once for roof planes covering an area larger than 2.5 m² (right).

The previously made statement that particularly small roof structures cause problems during the automatic building reconstruction of AOI 3 is confirmed by the two histograms shown in Figure 8.36. As can be seen, all roof planes in the reference that feature an area greater than 45 m² were successfully reconstructed with the proposed approach. Moreover, all automatically reconstructed roof planes greater than 25 m² were correctly reconstructed. As can be further seen, most problems are caused by reference roof planes that feature an area between 5 and 15 m². Therefore, the completeness is increased from 87.6 % to 93.9 % if in addition to all roof planes smaller or equal than 5 m² also all roof planes smaller or equal than 15 m² are ignored.

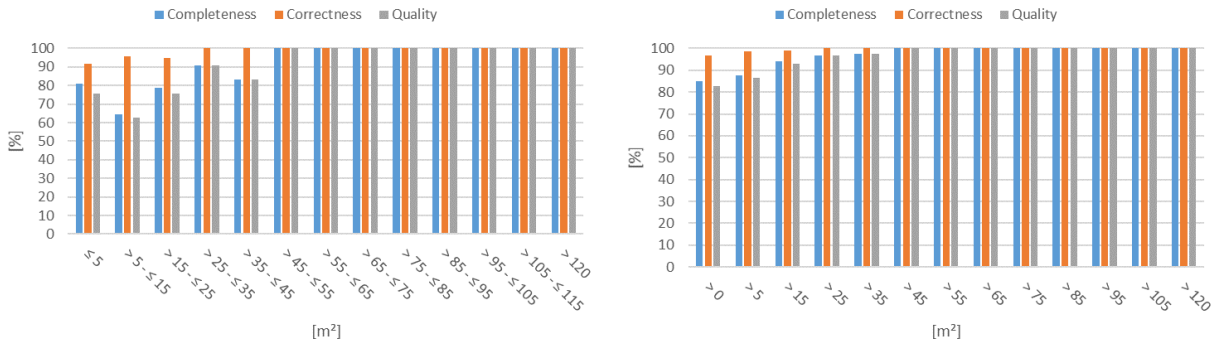


Figure 8.36. Histograms of the roof plane evaluation result as a function of the roof plane size once for area intervals (left) and once for roof planes larger than a certain area (cumulative) (right).

Regarding the geometric accuracy, the two histograms of RMS errors presented in Figure 8.37 show that more than 50 % of the roof planes feature along the reference boundaries and along the result boundaries an RMS error less than 0.3 m. For more than 75 % of the reference and the result roof planes, the RMS error is for both still less than 0.5 m. By taking the point spacing in the ALS data (~ 0.4 m) into account, it can be thus stated that the automatically reconstructed building models feature a high geometric accuracy. Compared to the other evaluated AOIs, the proposed reconstruction approach achieved in AOI 3 its best performance regarding the RMS error.

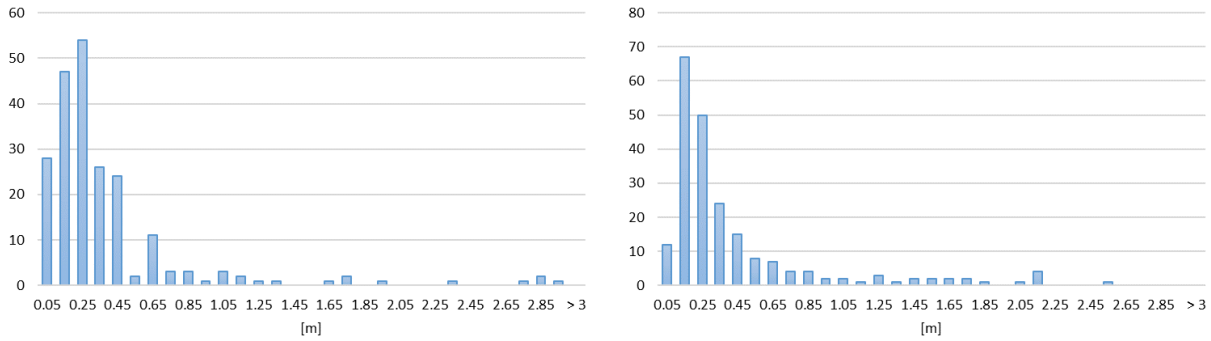


Figure 8.37. Histograms of RMS errors once along reference boundaries (left) and once along result boundaries (right).

A comparison of the height differences reveals that most areas where a correspondence between roof planes in the reference and the result was found feature a height difference less than 0.25 m. However, as can be seen in Figure 8.38, there are also a few buildings that partially exhibit areas with a height difference of more than 1 m. The blue colored area in the building in the southwest, for example, is caused by an actually present step edge that was detected during the automatic reconstruction but that have been differently reconstructed in the reference. The same applies to the rectangular area colored in red in the building that is located in the center of the AOI.

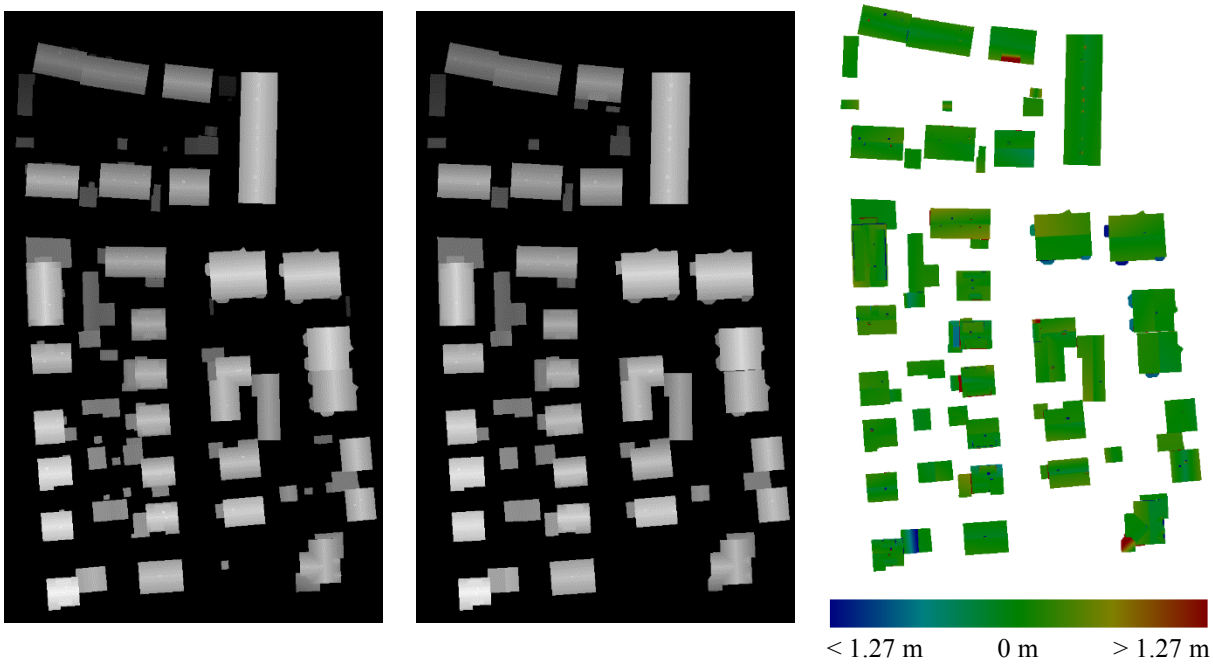


Figure 8.38. Digital surface models derived from the roof planes of the reference (left), from the roof planes of the result (middle), and the difference of these models where planes were found in both data sets (right).

As can be seen in Table 3.1, compared to other fully and semi-automatic building reconstruction approaches, the proposed grammar-guided approach TUB generally provides high completeness and correctness. There are only three other approaches out of 22 evaluated reconstruction approaches that have a slightly higher completeness than TUB (between 0.9 % and 3 % higher). Two of them feature, however, a significantly lower completeness compared to TUB (8.5 % and 12.3 % lower). The other one is BNU whose completeness value is only 2.1 % higher than TUB. Regarding the correctness, there are several reconstruction approaches that feature a slightly better correctness than TUB (max. 3.3 % higher). However, with the exception of BNU, all of them have a lower completeness compared to TUB (between 0.4 % and 20.8 % lower). An increase of the completeness if only roof planes larger than 10 m² are considered can be still identified for TUB (from 85.1 % to 92.7 %) but it is lower than for many other reconstruction approaches (e.g., YOR and MEL_HE). By considering only Comp, Corr, $Comp_{10}$, and $Corr_{10}$ it can be stated only BNU is slightly better in all of these metrics than TUB. However, if the topological differences between the roof planes in the reference and in the result are examined, it has to be stated that the number of N:1 relations is in BNU as well as in TUB dominant but their number is with 52 in BNU significantly higher than the 23 detected N:1 relations in TUB. In general, there seems to be a serious problem for all proposed reconstruction approaches in this AOI regarding N:1 relations which might be caused by the large number of chimneys which are difficult to recognize in the ALS data. With regard to the geometric accuracy in x-y direction, there is only one reconstruction approach (TUD2) that features with 0.5 m a 0.1 m smaller RMS_{XY} than TUB.

Table 8.9. Evaluation result (TUB) of AOI 3 in comparison to other evaluated (fully/ semi-automatic) reconstruction approaches presented in (ISPRS, 2017).

Abbrev.	Comp [%]	Corr [%]	$Comp_{10}$ [%]	$Corr_{10}$ [%]	Topo 1:N	Topo N:1	Topo N:M	RMS_{XY} [m]
CKU	81.3	98.4	91.9	99.1	4	48	2	0.8
FIE	82.6	83.1	88.7	93.4	7	44	5	1.1
ITCE1	67.7	**100.0**	72.6	**100.0**	**0**	47	2	0.8
ITCE2	64.3	**100.0**	64.5	**100.0**	**0**	46	**0**	1.0
ITCX1	70.2	**100.0**	72.6	**100.0**	1	48	**0**	0.7
ITCX2	86.0	84.4	93.5	95.0	5	48	2	0.7
ITCX3	**88.1**	88.2	96.8	95.8	3	50	2	0.7
TUD	74.5	93.0	83.1	98.0	**0**	42	1	0.7
VSK	76.6	99.1	86.3	**100.0**	3	50	**0**	0.8
YOR	84.7	**100.0**	**97.6**	**100.0**	2	51	1	0.6
CAS	73.2	**100.0**	83.1	**100.0**	**0**	48	**0**	0.8
MON	82.1	93.9	92.7	96.7	5	45	**0**	0.9
KNTU	80.4	96.7	91.9	97.7	**0**	52	**0**	0.9
BNU	87.2	**100.0**	96.0	**100.0**	2	52	**0**	0.6
MON_mod	74.5	96.2	91.1	**100.0**	5	39	2	0.8
MON2	73.2	89.2	91.9	99.1	7	34	2	0.8
TUD2	73.6	**100.0**	81.5	**100.0**	**0**	42	**0**	**0.5**
MEL_HE	82.6	97.8	96.0	**100.0**	2	44	1	0.7
WROC	80.4	98.2	89.5	**100.0**	**0**	49	3	0.8
WROC_2a	81.3	**100.0**	91.9	**100.0**	2	54	**0**	0.7
WROC_2b	81.7	**100.0**	92.7	**100.0**	3	52	3	0.6
MON5	80.9	99.3	91.1	99.1	7	36	4	0.8
TUB	85.1	96.7	92.7	98.3	6	**23**	8	0.6

In summary, it can be stated that the proposed reconstruction approach is generally suitable to reconstruct residential area with small detached houses if their points can be identified as building points. However, the automatic reconstruction of small roof superstructures like chimneys which are only difficult to recognize in the point cloud is still challenging.

8.3.4 Toronto: AOI 4

The 58 buildings of AOI 4 represent a mixture of low and high-storey buildings of which most of them feature flat rooftops on multiple levels. A major challenge for their automatic reconstruction is the large number of ventilations, cooling systems and other complex roof structures that are either located on top of the base roof or embedded in the base roof. The

pixel-based building detection result of AOI 4 presented in Figure 8.39 shows that most reference buildings were automatically reconstructed. There are only two small buildings in the northeast whose points were not correctly classified as building points and that were therefore not used as input data for the automatic reconstruction. Similarly, the points of the small gazebo with the conical shape in the center of the park in the east have not been classified as buildings during the classification process. This might be due to the reason that its shape is only difficult to distinguish from the shape of the surrounding trees.

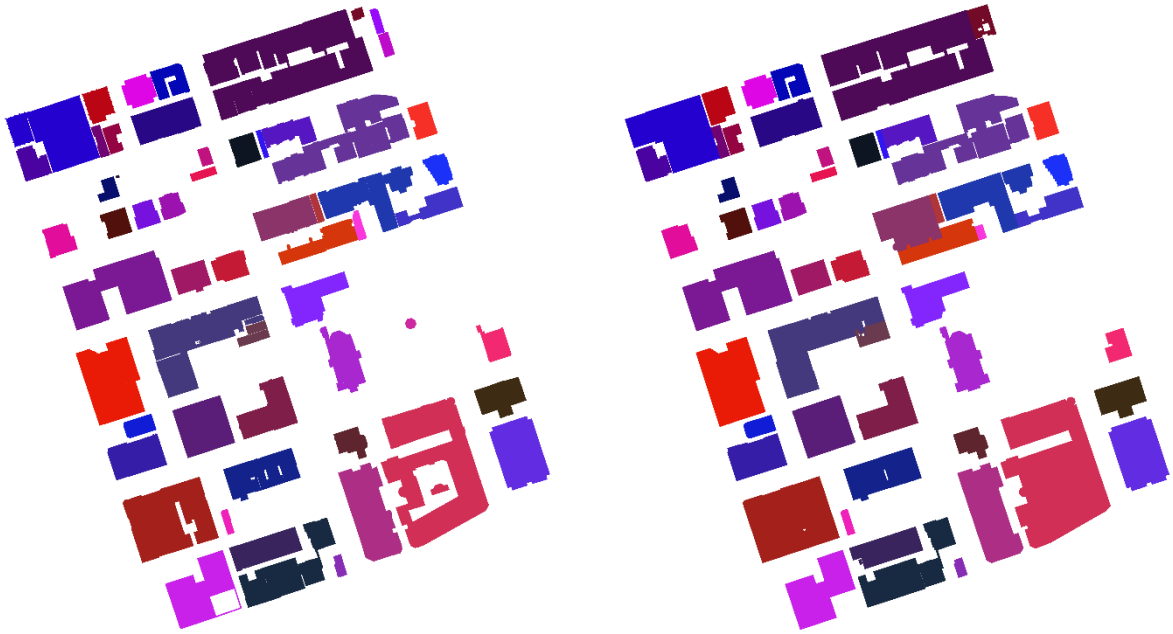


Figure 8.39. Pixel-based building detection result of the building models in the reference (left) and in the result (right).

A closer look on the evaluation result of the pixel-based building detection result which highlights the differences between the reference and the result supports the impression that most building areas were automatically reconstructed (see Figure 8.40). In contrast to the AOIs of the Vaihingen test site, areas of false positive classified pixels predominate the areas of false negative pixels. In order to explain the reason for this, various aspects have to be considered. Firstly, those false positive classified pixels that are located close to the building outline are essentially due to the facade adjustment procedure. Secondly, there are several narrow alleyways in AOI 4 that need to be represented as intrusions in the resulting building models. However, the proposed reconstruction approach did not recognize them as alleyways that need to be maintained but tried to close these gaps in the buildings. Note, this is not a general problem of the proposed reconstruction approach as it can be seen in the building in the northeast that features some small alleyways. Thirdly, some non-building points were classified as building points why the automatic reconstruction approach tried to incorporate them in the resulting building models. This is, for example the case for the narrow alleyway of the building in the southwest, the backyard of the building in the southeast, the M-shaped area of false positive classified pixels in the center, and the backyard in the northeast of

AOI 4. Fourthly, one building in the center next to the false positive classified M-shaped area is missing in the reference. This building, in which the Mozart Project Orchestra is currently located, is recognizable in the ALS data and the images as well and is therefore reasonably present in the result.

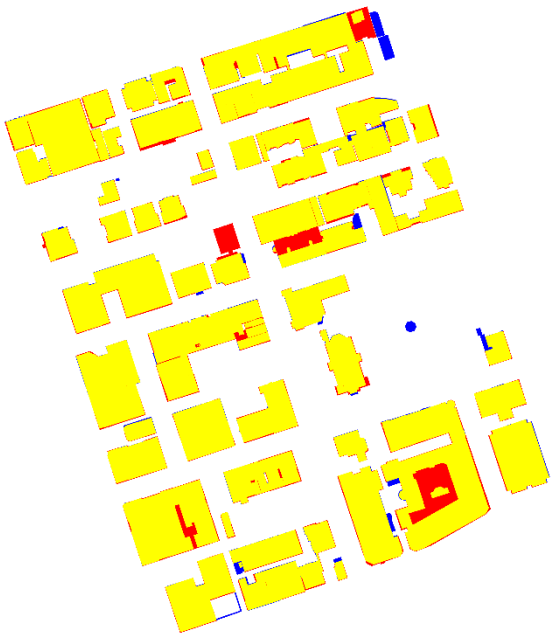


Figure 8.40. Evaluation result of the pixel-based building detection result (yellow: true positive pixels, blue: false negative pixels, and red: false positive pixels).

The roof plane detection result presented in Figure 8.41 shows a large number of small roof planes that are located on top of the buildings and that represent the aforementioned small roof structures which are challenging to automatically reconstruct. As can be seen, some of these small roof planes are either present in the reference or in the result but not in both. Particularly these roof planes cause the large number of 1:N, N:1, and N:M relations that become visible in the assessment of the topological differences. In contrast, roof planes belonging to base roofs and other large roof planes are generally well represented in the result. The advantage of applying sub-surface growing becomes particularly visible in this AOI. Several small roof details that would be otherwise lost were maintained and became part of larger roof planes. Gaps in lower level roof structures that were caused by upper level roof structures have been automatically closed by sub-surface growing so that their identification and reconstruction became straightforward.

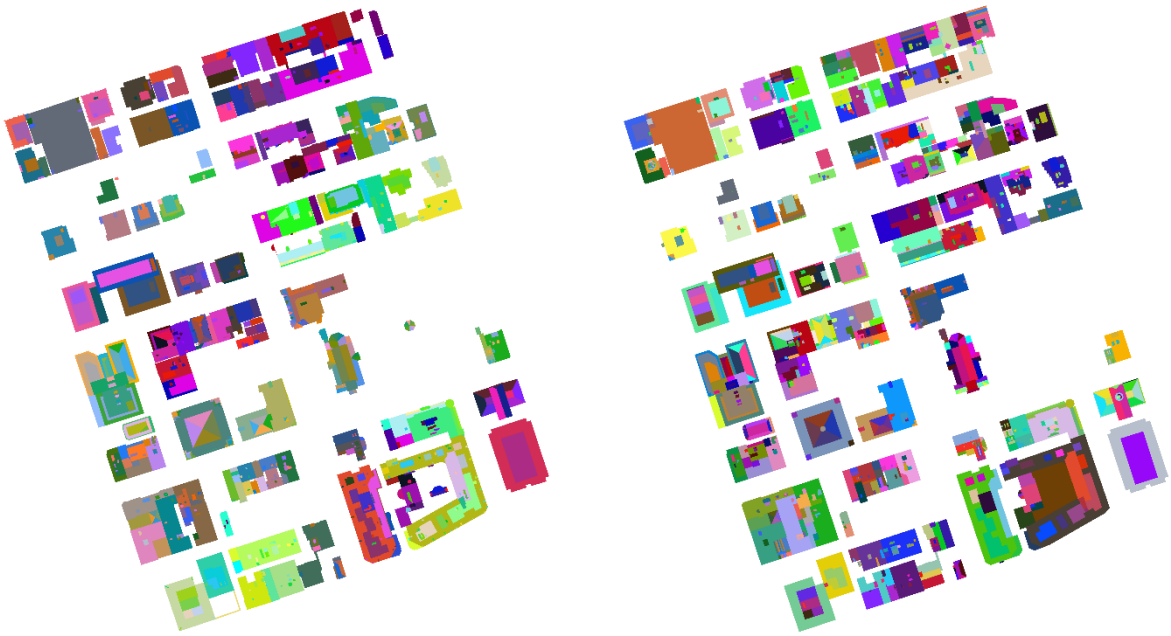
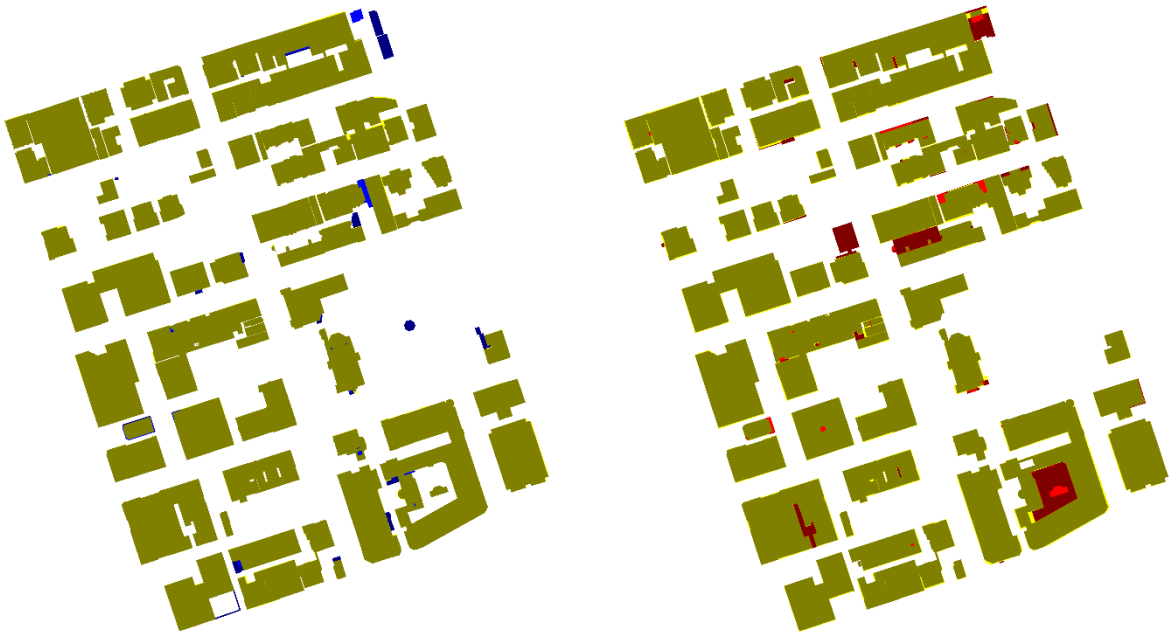


Figure 8.41. Roof plane detection result of the building models in the reference (left) and in the result (right).

By taking only the evaluation result of the roof plane detection result into account (see Figure 8.42), the impression is given that there is a high correspondence between roof planes in the reference and in the result. Besides those areas where a misclassification of points is present, there are only a few pixels in the reference roof planes that have an insufficient overlap with the roof planes of the result or vice versa. The circular roof plane in the center of the squared building in the west, for example, is only present in the result but not in the reference and represents a missing chimney in the reference.



Ochre: Pixels available in both the reconstruction results and the reference and belonging to a reference plane found to have sufficient overlap with planes in the roof reconstruction results.

Yellow: Pixels that belong to a reference plane found to have sufficient overlap with planes in the roof reconstruction results but that are not inside a building in the reconstruction results.

Dark blue: Pixels in reference roof planes not having sufficient overlap with detected roof planes and not being inside a building in the reconstruction results.

Bright blue: Pixels in reference roof planes not having sufficient overlap with detected roof planes, but being inside a building in the reconstruction results.

Ochre: Pixels available in both the reconstruction results and the reference and belonging to a detected roof plane found to have sufficient overlap with planes in the reference.

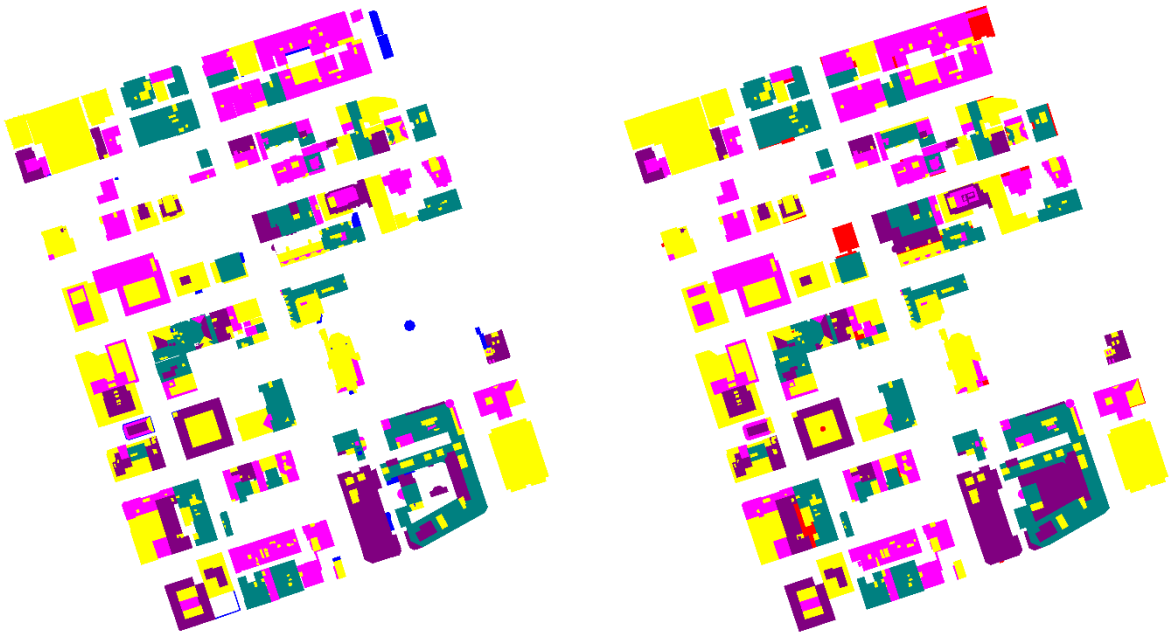
Yellow: Pixels that belong to a detected roof plane found to have sufficient overlap with planes in the reference but that are not inside a building in the reference.

Dark red: Pixels in detected roof planes not having sufficient overlap with roof planes in the reference and not being inside a building in the reference.

Bright red: Pixels in detected roof planes not having sufficient overlap with reference planes, but being inside a building in the reference.

Figure 8.42. Evaluation result of the roof plane detection result once from the point of view of the reference (left) and once from the point of view of the result (right).

The difficulties for building reconstruction approaches to automatically reconstruct buildings in AOI 4 become particularly apparent if topological differences between roof planes in the reference and in the result are examined. As can be seen in Figure 8.43, in comparison to the AOIs in Vaihingen, there is a large number of roof planes that do not feature a 1:1 relation. This are primarily caused by the aforementioned large number of small roof structures. It is, however, important to note that not all N:1 and 1:N relations indicate a missing roof plane in the result or in the reference but some of them are due to the implementation of the topological test itself. As a result, only a few buildings consist of roof planes that all feature a 1:1 relation. This is, for example, the case for the two buildings that are once located in the northwest and once in the southeast.



Yellow: Pixels inside reference planes having a 1:1 relation to planes in the reconstruction results.

Blue: Pixels inside reference planes that have no corresponding plane in the extraction results.

Dark magenta: Pixels in reference roof planes that are merged with other planes in the reconstruction results (N:1 relation).

Bright magenta: Pixels in reference roof planes split into two or more planes in the reconstruction results (1:N relation).

Dark cyan: Pixels in reference roof planes being part of a plane cluster having N:M relations between planes in the reference and planes in the reconstruction results.

Yellow: Pixels inside reconstructed planes having a 1:1 relation to planes in the reference.

Red: Pixels inside reconstructed planes that have no corresponding plane in the reference.

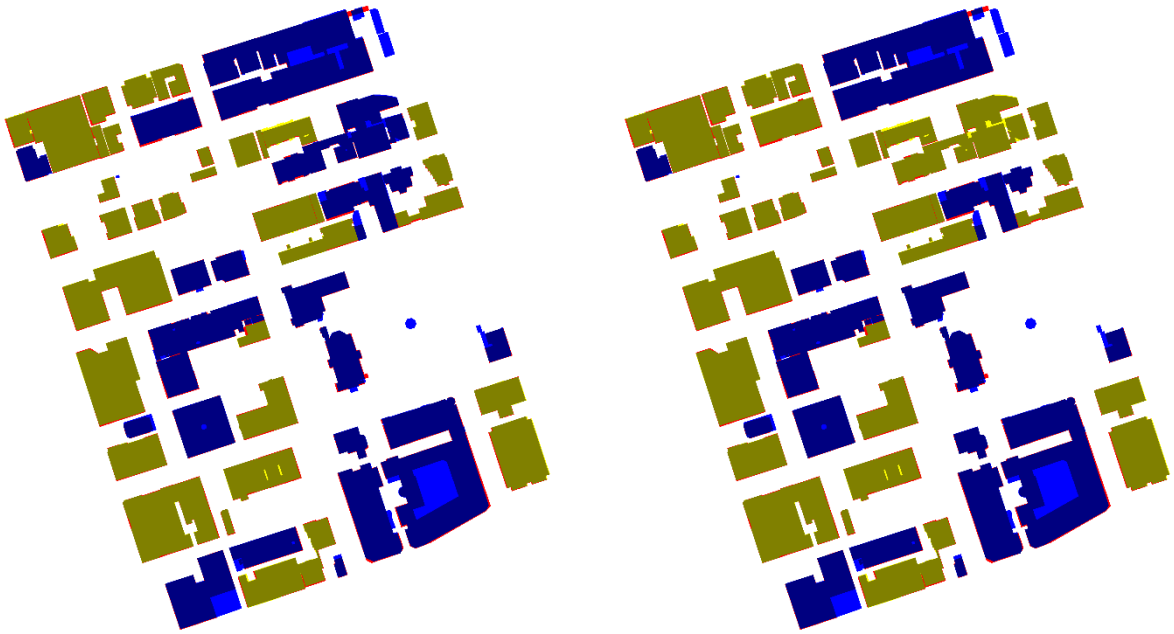
Dark magenta: Pixels in reconstructed roof planes that correspond to several planes of the reference (1:N relation).

Bright magenta: Pixels in reconstructed roof planes that are the result of a split of a plane in the reconstruction results (N:1 relation).

Dark cyan: Pixels in reconstructed roof planes being part of a plane cluster having N:M relations between planes in the reconstruction results and planes in the reference.

Figure 8.43. Assessment of the topological differences between the reference and the reconstruction result once from the point of view of the reference data (left) and once from the point of view of the detected roof planes (right).

The result of the assessment of the completeness shown in Figure 8.44 reveals that, on the one side, the majority of buildings (~ 57 %) feature roof planes for which corresponding roof planes with sufficient overlap were detected in the result. On the other side, a significant proportion (~ 43 %) of the reference buildings seems to be incomplete in the result. Their number is only decreased by two if only roof planes greater than 2.5 m² are considered. In fairness, it has to be considered that the average number of roof planes of which a building in AOI 4 consists is significant larger compared to the buildings in the Vaihingen test site. This, in turn, makes the complete reconstruction of a building more difficult.



Ochre: Pixels in buildings for which all reference roof planes were found to have correspondences of sufficient overlap in the reconstruction results, but only if the pixel is inside a building both in the reference and in the reconstruction results.

Yellow: Pixels in buildings for which all reference roof planes were found to have correspondences of sufficient overlap in the reconstruction results; however, these pixels are not inside a building in the reconstruction results.

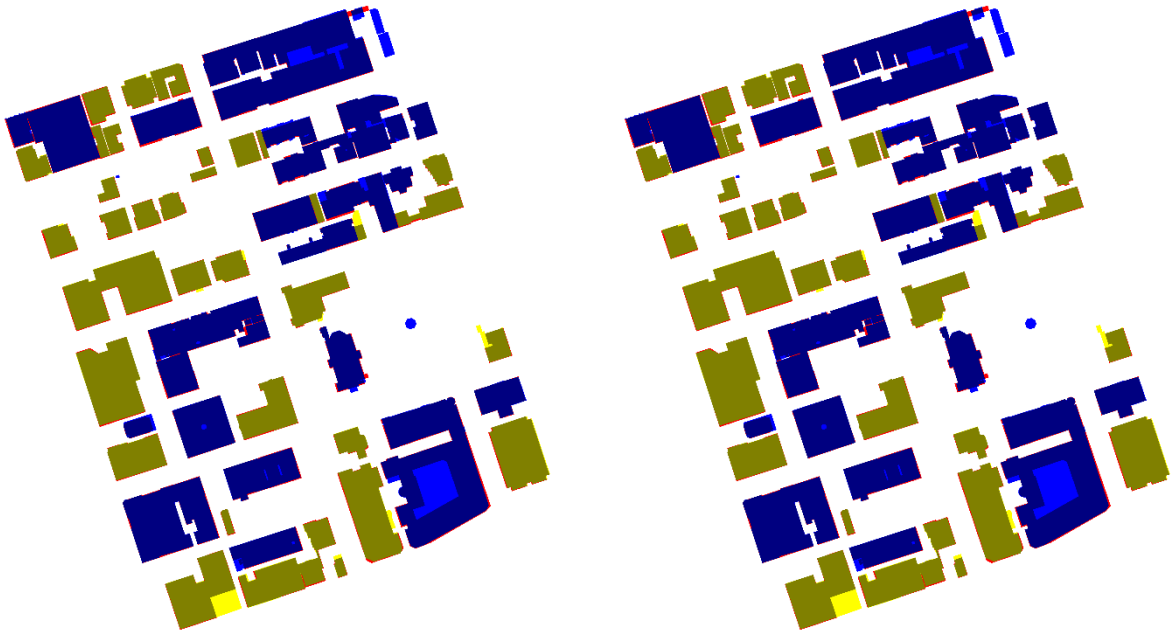
Bright red: Pixels in correctly detected roof planes that are not inside a building in the reference.

Dark blue: Pixels in buildings for which at least one reference roof plane was not detected in the reconstruction results and which are inside a roof plane in the reconstruction results.

Bright blue: Pixels in buildings for which at least one reference roof plane was not detected in the reconstruction results and which are not inside a building in the reconstruction results.

Figure 8.44. Assessment of the completeness once for all detected roof planes (left) and once for roof planes covering an area larger than 2.5 m² (right).

A similar pattern can be seen in Figure 8.45 which shows the correctness of the roof planes. The majority of automatically reconstructed buildings (~ 59 %) feature pixels for which all roof planes were found to have correspondences of sufficient overlap in the reference. Conversely, a significant proportion (~ 41 %) of the automatically reconstructed buildings feature at least one roof plane that is not correct according to the reference. It is interesting to note, that 60 % of the buildings that have been identified as incomplete belong also to the class of buildings that feature at least one incorrect roof plane. Particularly these buildings (~ 26 % of the buildings in AOI 4) are difficult to reconstruct with all their details.



Ochre: Pixels in buildings for which all automatically reconstructed roof planes were found to have correspondences of sufficient overlap in the reference, but only if the pixel is inside a building both in the reference and in the reconstruction results.

Yellow: Pixels in buildings for which all automatically reconstructed roof planes were found to have correspondences of sufficient overlap in the reference; however, these pixels are not inside a building in the reconstruction results.

Bright red: Pixels in correctly detected roof planes that are not inside a building in the reference.

Dark blue: Pixels in buildings for which at least one automatically reconstructed roof plane was not correct and which are inside a roof plane in the reconstruction results.

Bright blue: Pixels in buildings for which at least one automatically reconstructed roof plane was not correct and which are not inside a building in the reconstruction results.

Figure 8.45. Assessment of the correctness of all detected roof planes (left) and assessment of the correctness of roof planes covering an area larger than 2.5 m².

The statement that particularly small roof structures caused problems during the automatic building reconstruction is confirmed by the two histograms shown in Figure 8.46. As can be seen, the completeness of roof planes that feature an area less or equal than 5 m² is only 72.9 %. Due to their large number in AOI 4, the cumulative completeness is significantly increased from 88.1 % to 95 % if these roof planes are ignored. Further low completeness values were detected for roof planes that feature an area around 65 m². However, their total number is low why each of these roof planes has a high impact on the completeness values. This is reflected by the rather small change in the cumulative histogram that becomes visible if these roof planes are ignored. It can be thus once again concluded that particularly small rooftop details cause major problems but that there are also a few rather large roof planes in AOI 4 that are incomplete in the automatically reconstructed building models. Additionally, by considering the large number of small roof planes in AOI 4, it is interesting to note that the

correctness of roof planes smaller or equal than 5 m² is with 94.3 % rather high why it changes in the cumulative histogram only between 93.1 % and 97.5 %.

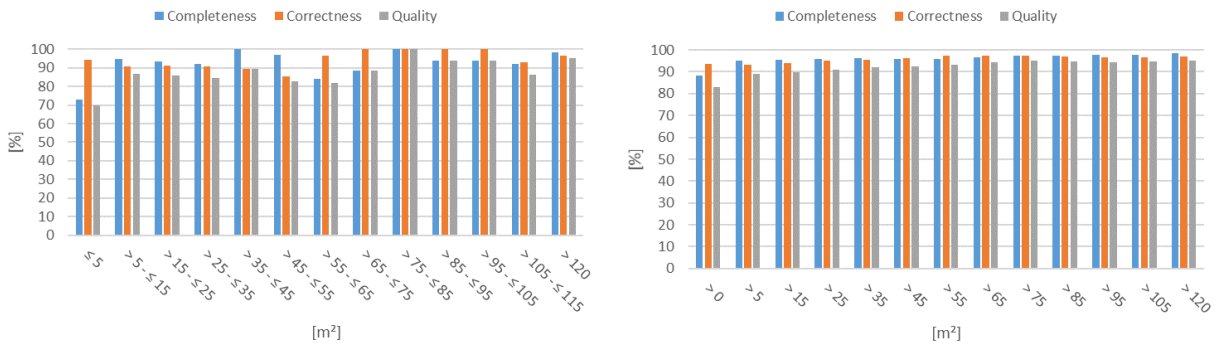


Figure 8.46. Histograms of the roof plane evaluation result as a function of the roof plane size once for area intervals (left) and once for roof planes larger than a certain area (cumulative) (right).

As can be seen in Figure 8.47, more than 50 % of the roof planes feature along the reference boundaries and along the result boundaries an RMS error less than 0.6 m and 0.5 m, respectively. For more than 75 % of the reference and the result roof planes, the RMS error is still less than 1.1 m and 1.0 m, respectively. It is furthermore interesting to note that there is a remarkable large number of roof planes in both the reference and the result of AOI 4 that feature an RMS error less than 0.1 m whereas this is usually the case for roof planes in the AOIs of the Vaihingen test site that feature an area between 0.1 m and 0.3 m. In spite of this, the average RMS error of roof planes is significantly larger in AOI 4 than in the AOIs of the Vaihingen test site. This is mainly due to the roof planes that feature an RMS error larger than 1.5 m because their number is significantly larger in AOI 4 than in the AOIs of the Vaihingen test site.

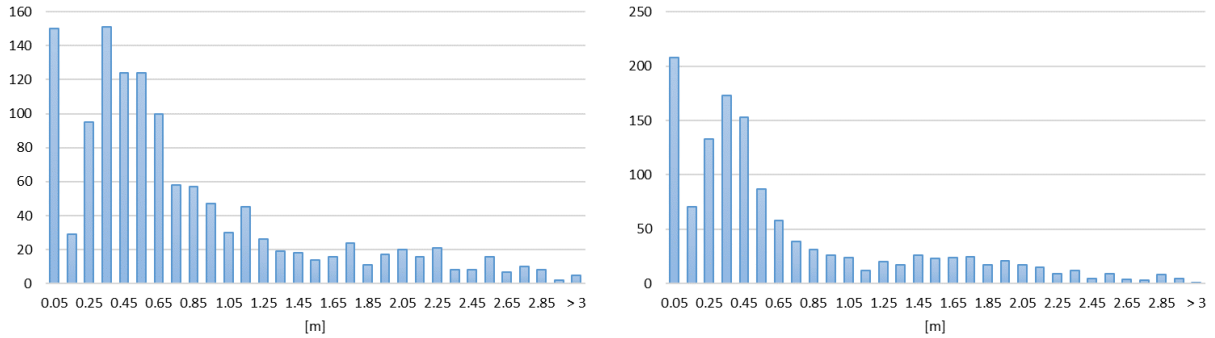


Figure 8.47. Histograms of RMS errors once along reference boundaries (left) and once along result boundaries (right).

The comparison of the height differences shown in Figure 8.48 reveals a more heterogeneous picture than for the AOIs of the Vaihingen test site. Several areas are clearly recognizable that feature a height difference larger than 1 m. These areas do not only belong to small roof planes but to base roofs as well. For example, the building located in the northwest features a large area that indicates with its blue color a height difference of more than 1 m. This building

has a parking lot on the rooftop that was differently reconstructed in the reference and in the result. In the result, the parking lot it represented as a horizontal plane whereas a height change can be detected in the reference. Furthermore, some small dome roofs can be identified that are missing in the manually reconstructed building models but they are present in the result of the automatic reconstruction. This is, for example the case for the building in the northwest as well as for the building in the southeast that feature both a circular red area in their center.

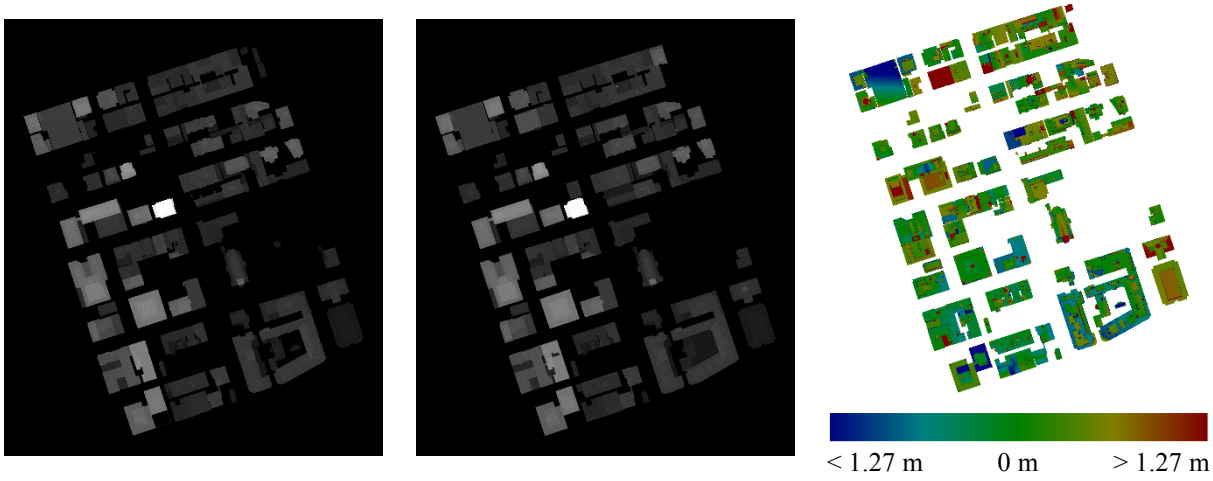


Figure 8.48. Digital surface models derived from the roof planes of the reference (left), from the roof planes of the result (middle), and the difference of these models where planes were found in both data sets (right).

Due to the challenges of AOI 4, the number of submitted fully and semi-automatically reconstructed building models is significantly lower than for the AOIs of the Vaihingen test site. As can be seen in Table 8.10, the automatically reconstructed buildings of the proposed grammar-guided reconstruction approach TUB provides with 88.1 % a significantly higher completeness than the others (12.6 % higher than the second best completeness value). By taking into account the difficulties of AOI 4, it can be stated that the completeness of TUB is rather high. The latter applies to the correctness of TUB as well although YOR provides with 97.5 % a higher completeness than TUB with 93.4 %. A similar pattern can be seen if only roof planes larger than 10 m² are considered. The large number of roof plane relations that do not feature a 1:1 relation has to be seen from the point of view that some details are difficult to detect in the input data or are even missing in the manually reconstructed building models. Therefore, it would be unfair to conclude that over-segmentation is the major problem of TUB for the reconstruction of AOI 4. The geometric accuracy of the resulting building models of TUB is slightly greater than twice the ALS point spacing but it is with 0.9 m still 0.35 m lower than the average and thereby among the best approaches.

Table 8.10. Evaluation result (TUB) of AOI 4 in comparison to other evaluated (fully/ semi-automatic) reconstruction approaches presented in (ISPRS, 2017).

Abbrev.	Comp [%]	Corr [%]	$Comp_{10}$ [%]	$Corr_{10}$ [%]	Topo 1:N	Topo N:1	Topo N:M	RMS_{XY} [m]
CKU	68.6	80.2	72.9	79.1	42	74	86	1.7
YOR	75.5	**97.5**	86.0	**97.6**	**27**	109	**19**	**0.8**
FIE	52.3	91.5	60.4	91.9	56	62	36	1.4
MON2	70.2	78.3	87.1	89.0	180	**30**	84	1.1
TUB	**88.1**	93.4	**94.7**	93.6	104	52	65	0.9

In summary, since AOI 4 consists of low and high-storey buildings that feature a lot of small roof structures which are not always recognizable in the point cloud due to their low point density, the proposed grammar-guided reconstruction approach reaches its limit if all details are required. However, the same applies for the manually reconstruction which also produces in many cases only incomplete building models. Therefore, by taking the manual effort and the quality of the resulting building models into account, the proposed reconstruction approach still provides good results.

8.3.5 Toronto: AOI 5

AOI 5 comprises 38 buildings that represent high-rise buildings in a typical central business district of North Amerika. As can be seen in Figure 8.49, most buildings of the reference were automatically reconstructed but there are also some small areas that are either missing in the reference or in the result. Most of them are located between Richmond Street West and King Street West. The differences are caused by several reasons. Firstly, some reference buildings were wrongly manually reconstructed. They can neither be detected in the data nor in reality so that they should be excluded from the reference. This applies, for example, to the L-shaped brown and the U-shaped magenta colored buildings in the northern center of AOI 5 and to the small circular shaped building colored in blue that is located in the northwest. Secondly, some differences are caused by missing building parts in the reference that can be only hardly recognized from the ALS data and the images. For example, the three reference buildings that are located between the three magenta colored buildings in the west were merged to a single building in the result that features a significantly larger area than the three separated buildings. A closer look exposes that it is a one-storeyed building whose flat roof is used as a passageway with green plants and as a resting area which includes a hotel pool. Therefore, this building is not wrongly present in the result but it is missing in the manually reconstructed reference. Note, the rectangular gap in the automatically reconstructed building is due to an accumulation of several skylight windows which did not reflect the laser beams during the capturing of the ALS data. Similarly, the building in the north of AOI 5 features a large backyard in the reference. Due to the high vegetation in this area, it cannot be seen from the images if the vegetation is located on top of a roof or on ground level. Only the heights of the points in the ALS data indicate that a building might be below the vegetation. Consequently,

this area is again missing in the manual reconstruction. The same applies to the backyard of the dark colored reference building in the east. The high discrepancy between the reference and the result already shows the high degree of difficulty to reconstruct the buildings of AOI 5.

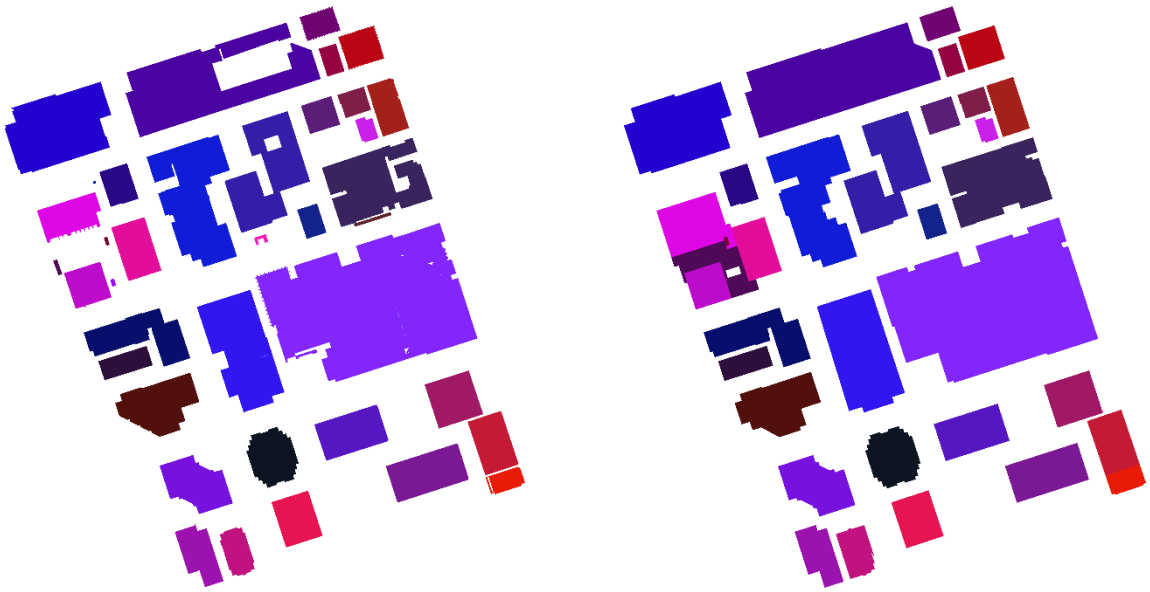


Figure 8.49. Pixel-based building detection result of the building models in the reference (left) and in the result (right).

Further differences of the pixel-based building detection results are highlighted in Figure 8.50. For example, the squared area in the north that is completely surrounded by a building is represented as a building component with a flat roof in the result but it is not part of the reference. Based on the images and due to the high-rise buildings around this area, it cannot be determined if it is a one-storeyed building with a flat rooftop and a single tree on it or if it is a small forecourt. A closer look on the ALS data reveals, however, that its points are around 2 m above the points of the streets that surround the building why it is correctly represented in the result but not in the reference. Further differences between the reference and the result are visible close to the building facades. In contrast to the evaluation results of the AOIs in Vaihingen, they are not all due to the adjustment procedure. Some of them are caused by lower-level building details that are visible in the ALS data but not in the images because they are hidden by high-rise buildings. An example for this is given for the building in the west that features in front of the high-rise facade which is facing southwest, three lower-level flat rooftops with different heights that compose a stepped shape. Another example is given east of this building where an additional U-shaped building part is present in the result. This part represents the partly open entrance area of the high-rise building and it is thus missing in the reference. Overall, it can be concluded that particularly high-rise buildings caused some errors in the manually reconstructed reference so that some lower-level building details are missing in the reference but not necessarily in the result of the automatic reconstruction.

Figure 8.50. Evaluation result of the pixel-based building detection result (yellow: true positive pixels, blue: false negative pixels, and red: false positive pixels).

The roof plane detection result shown in Figure 8.51 reveals that the buildings in AOI 5 feature numerous roof details similar to AOI 4. Additionally, there is a good correspondence between large roof planes in the reference and in the result. However, several differences can be detected, particularly if small roof planes are taken into account. These small roof planes are challenging to reconstruct and cause several topological differences, as will be shown later. As can be further seen, sub-surface growing is able to reconstruct many small roof details that would be otherwise missed. Analogous to the reconstruction of AOI 4, segments were able to grow below upper level roof structures so that lower-level roof structures consist of larger roof segments. This in turn supports their unambiguous identification and helps to significantly reduce the number of production rules of the grammar-guided roof structure recognition step.

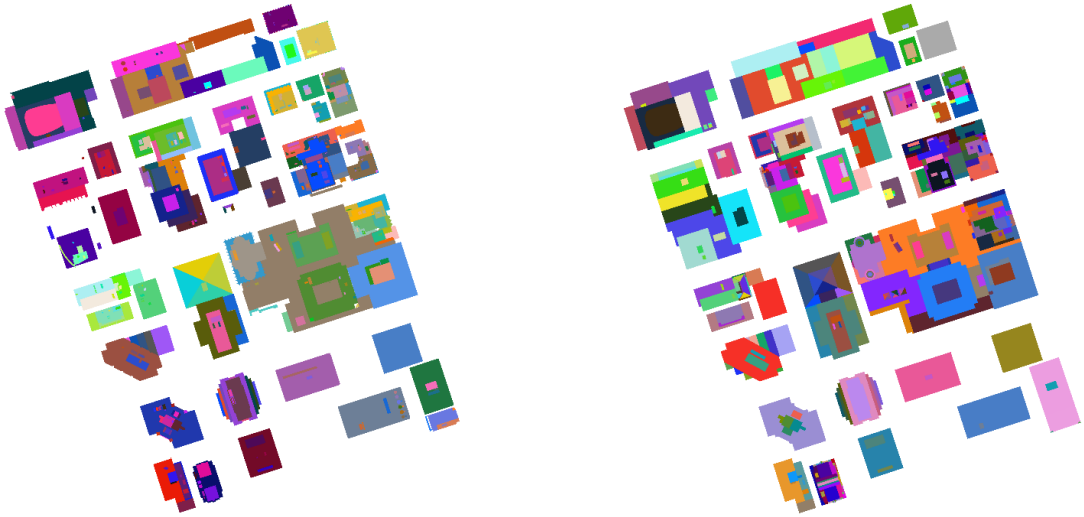
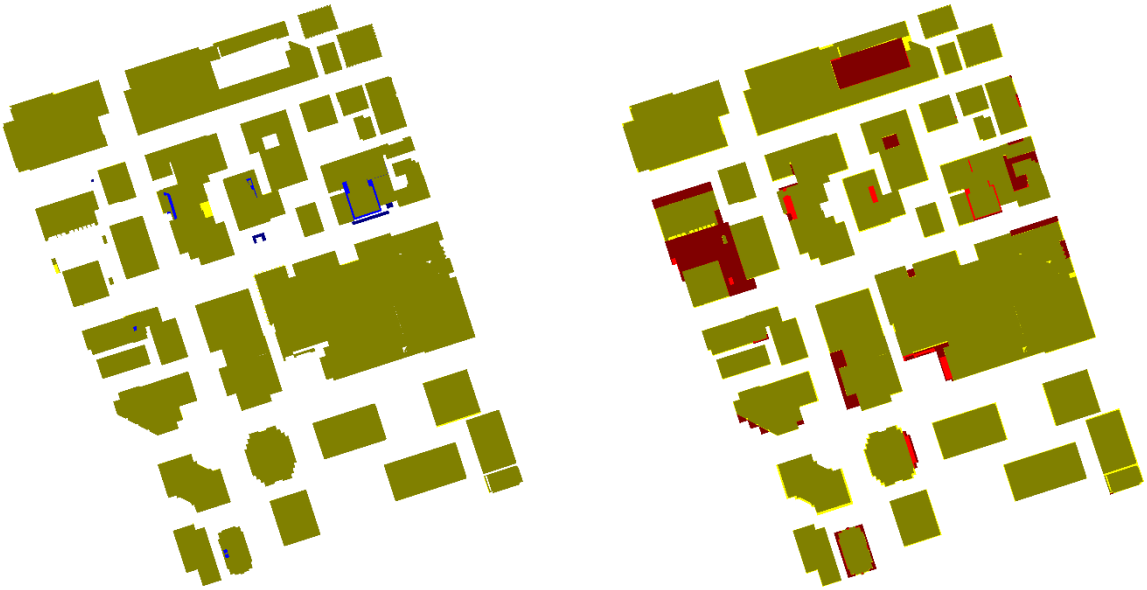


Figure 8.51. Roof plane detection result of the building models in the reference (left) and in the result (right).

The evaluation result of the roof plane detection result presented in Figure 8.52 shows a high correspondence between the reference and the result roof planes. Besides those areas that are missing in the reference, a large majority of pixels belong to a reference and a result plane that feature a sufficient overlap. It can be thus stated that the essential building characteristics are adequately reflected in the automatically reconstructed building models.



Ochre: Pixels available in both the reconstruction results and the reference and belonging to a reference plane found to have sufficient overlap with planes in the roof reconstruction results.

Yellow: Pixels that belong to a reference plane found to have sufficient overlap with planes in the roof reconstruction results but that are not inside a building in the reconstruction results.

Dark blue: Pixels in reference roof planes not having sufficient overlap with detected roof planes and not being inside a building in the reconstruction results.

Bright blue: Pixels in reference roof planes not having sufficient overlap with detected roof planes, but being inside a building in the reconstruction results.

Ochre: Pixels available in both the reconstruction results and the reference and belonging to a detected roof plane found to have sufficient overlap with planes in the reference.

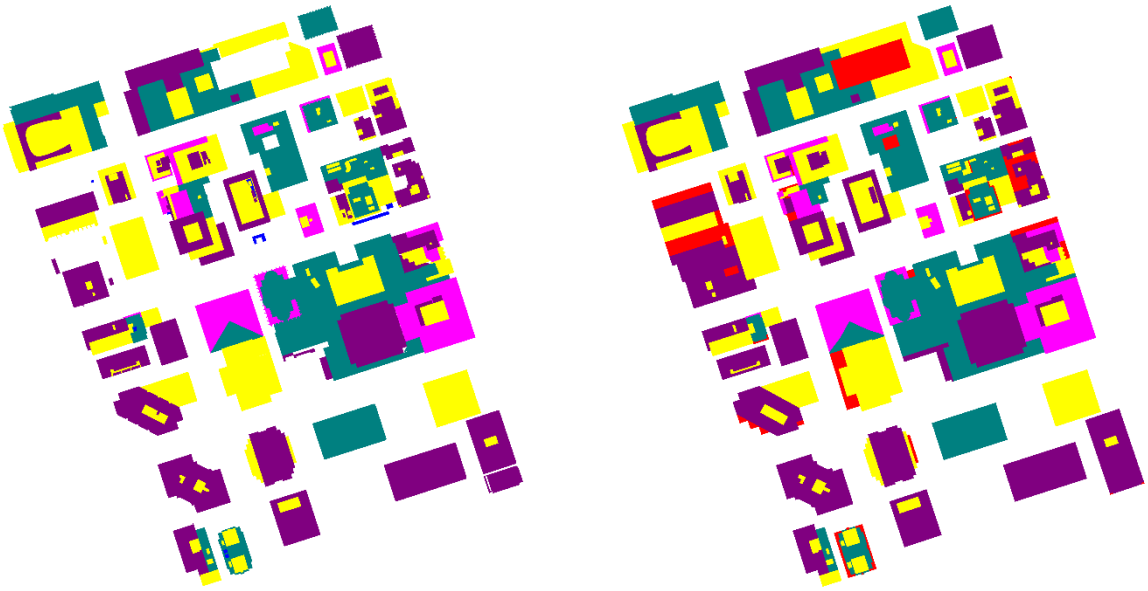
Yellow: Pixels that belong to a detected roof plane found to have sufficient overlap with planes in the reference but that are not inside a building in the reference.

Dark red: Pixels in detected roof planes not having sufficient overlap with roof planes in the reference and not being inside a building in the reference.

Bright red: Pixels in detected roof planes not having sufficient overlap with reference planes, but being inside a building in the reference.

Figure 8.52. Evaluation result of the roof plane detection result once from the point of view of the reference (left) and once from the point of view of the result (right).

Similarly to AOI 4, the assessment of the topological differences reveals that only a few roof planes feature a 1:1 relation in AOI 5 (see Figure 8.53). Large areas that were both manually and automatically reconstructed have often an N:1 or N:M relation while 1:N relations are more rarely. The ambiguities are basically caused by the large number of small roof structures. The topological differences thus illustrate once again the difficulties and challenges for automatic reconstruction approaches to reconstruct high-rise buildings that feature a number of complex roof structures.



Yellow: Pixels inside reference planes having a 1:1 relation to planes in the reconstruction results.

Blue: Pixels inside reference planes that have no corresponding plane in the extraction results.

Dark magenta: Pixels in reference roof planes that are merged with other planes in the reconstruction results (N:1 relation).

Bright magenta: Pixels in reference roof planes split into two or more planes in the reconstruction results (1:N relation).

Dark cyan: Pixels in reference roof planes being part of a plane cluster having N:M relations between planes in the reference and planes in the reconstruction results.

Yellow: Pixels inside reconstructed planes having a 1:1 relation to planes in the reference.

Red: Pixels inside reconstructed planes that have no corresponding plane in the reference.

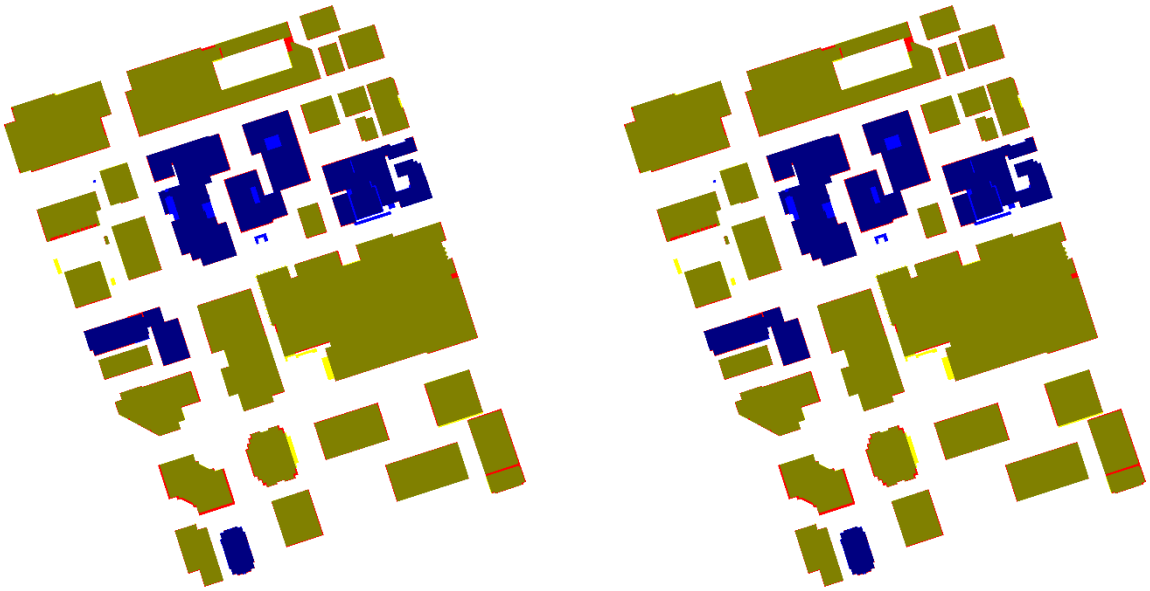
Dark magenta: Pixels in reconstructed roof planes that correspond to several planes of the reference (1:N relation).

Bright magenta: Pixels in reconstructed roof planes that are the result of a split of a plane in the reconstruction results (N:1 relation).

Dark cyan: Pixels in reconstructed roof planes being part of a plane cluster having N:M relations between planes in the reconstruction results and planes in the reference.

Figure 8.53. Assessment of the topological differences between the reference and the reconstruction result once from the point of view of the reference data (left) and once from the point of view of the detected roof planes (right).

As can be seen in Figure 8.54, there are only five automatically reconstructed buildings for which at least one reference roof plane is missing. Their number remain unchanged if only roof planes larger than 2.5 m² are considered. Consequently, some details in the result are either missing or were differently modeled. By taking into account the number of complex building structures that are present in AOI 5, it can be stated that the proposed automatic reconstruction approach shows a very good performance with regard to the completeness.



Ochre: Pixels in buildings for which all reference roof planes were found to have correspondences of sufficient overlap in the reconstruction results, but only if the pixel is inside a building both in the reference and in the reconstruction results.

Yellow: Pixels in buildings for which all reference roof planes were found to have correspondences of sufficient overlap in the reconstruction results; however, these pixels are not inside a building in the reconstruction results.

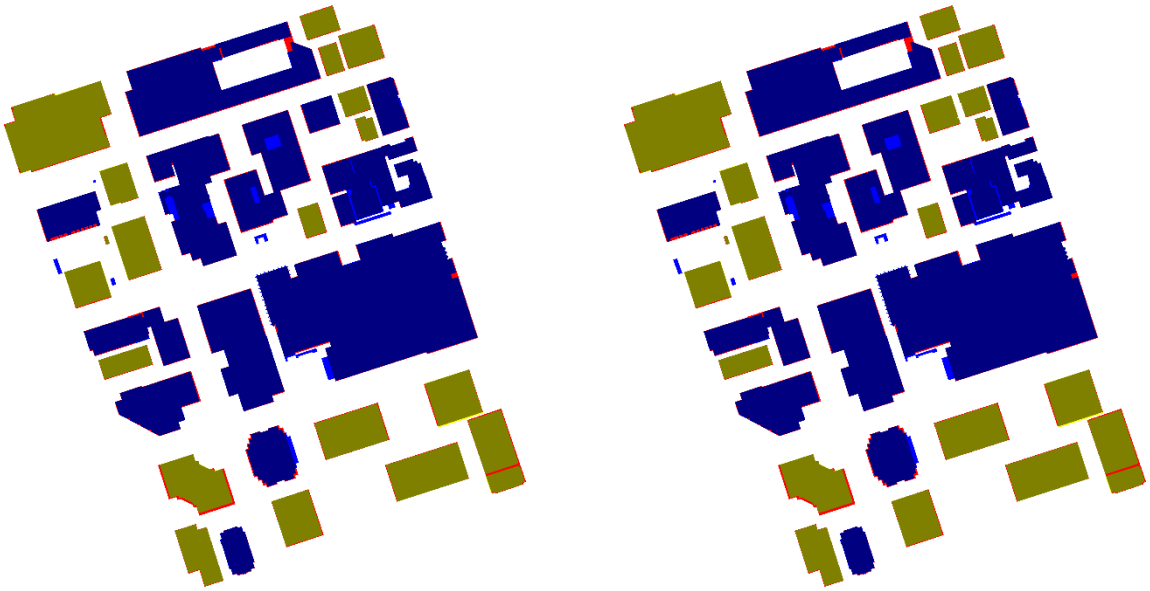
Bright red: Pixels in correctly detected roof planes that are not inside a building in the reference.

Dark blue: Pixels in buildings for which at least one reference roof plane was not detected in the reconstruction results and which are inside a roof plane in the reconstruction results.

Bright blue: Pixels in buildings for which at least one reference roof plane was not detected in the reconstruction results and which are not inside a building in the reconstruction results.

Figure 8.54. Assessment of the completeness once for all detected roof planes (left) and once for roof planes covering an area larger than 2.5 m² (right).

The evaluation of the correctness presented in Figure 8.55, in turn, shows that twelve of the automatically reconstructed buildings feature each at least one roof plane that is not correct regarding the reference roof planes. Its number is only decreased by one if only roof planes larger than 2.5 m² are considered. As can be seen, particularly rooftops of large buildings are affected or those that feature multiple levels.



Ochre: Pixels in buildings for which all automatically reconstructed roof planes were found to have correspondences of sufficient overlap in the reference, but only if the pixel is inside a building both in the reference and in the reconstruction results.

Yellow: Pixels in buildings for which all automatically reconstructed roof planes were found to have correspondences of sufficient overlap in the reference; however, these pixels are not inside a building in the reconstruction results.

Bright red: Pixels in correctly detected roof planes that are not inside a building in the reference.

Dark blue: Pixels in buildings for which at least one automatically reconstructed roof plane was not correct and which are inside a roof plane in the reconstruction results.

Bright blue: Pixels in buildings for which at least one automatically reconstructed roof plane was not correct and which are not inside a building in the reconstruction results.

Figure 8.55. Assessment of the correctness of all detected roof planes (left) and assessment of the correctness of roof planes covering an area larger than 2.5 m².

For a more differentiated analysis of the completeness and the correctness, roof planes of certain sizes were separately analyzed from each other. As can be seen in Figure 8.56, the completeness is already greater than 95 % if only roof planes larger than 15 m² are taken into account whereas a correctness greater than 90 % is achieved if only roof planes greater than 115 m² are considered. It can be thus stated that particularly small reference roof planes are not always properly represented in the result. This is, for example, the case for 30.8 % of the reference roof planes that are smaller or equal than 5 m². Additionally, it is interesting to note that the correctness of the planes that feature an area smaller or equal than 5 m² is with 64.3 % low but that similar values were achieved for larger area ranges as well. The latter, however, have a low impact on the overall correctness (cp. cumulative histogram) due to their low number in AOI 5.

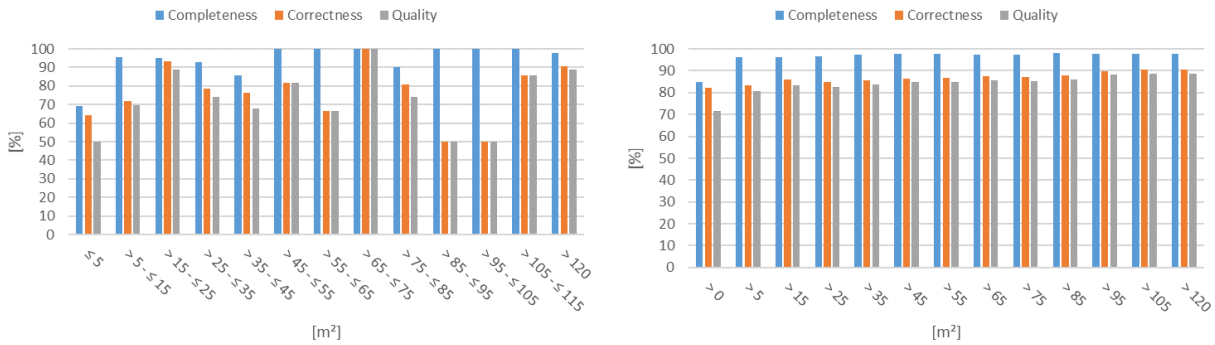


Figure 8.56. Histograms of the roof plane evaluation result as a function of the roof plane size once for area intervals (left) and once for roof planes larger than a certain area (cumulative) (right).

Regarding the geometric accuracy, the two histograms of RMS errors presented in Figure 8.57 show that more than 50 % of the roof planes feature along the reference boundaries and along the result boundaries an RMS error less than 0.5 m and 0.6 m, respectively. For more than 75 % of the reference and the result roof planes, the RMS error is still less than 1.1 m and 1.0 m, respectively. Additionally, a remarkable large number of roof planes in both the reference and the result feature an RMS error less than 0.1 m. However, there are still some roof planes that have an RMS error larger than 1.5 m. A similar pattern was already seen in AOI 4.

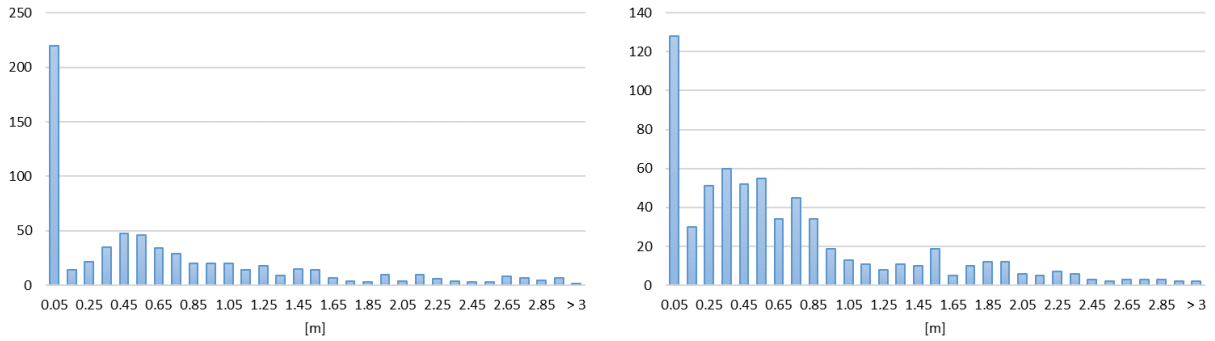


Figure 8.57. Histograms of RMS errors once along reference boundaries (left) and once along result boundaries (right).

The comparison of the height differences shown in Figure 8.58 reveals that several areas feature a height difference greater than 1 m. These areas do not only belong to small roof planes of roof superstructures but to base roofs as well. Some of them are partly covered by vegetation (e.g., trees) on their rooftop which makes it difficult to determine the actual height of the building. This is, for example, the case for the two blue areas of the building complex in the north. Further differences are due to ventilation shafts which are difficult to recognize in the ALS data or to other installations that were differently reconstructed in the reference.

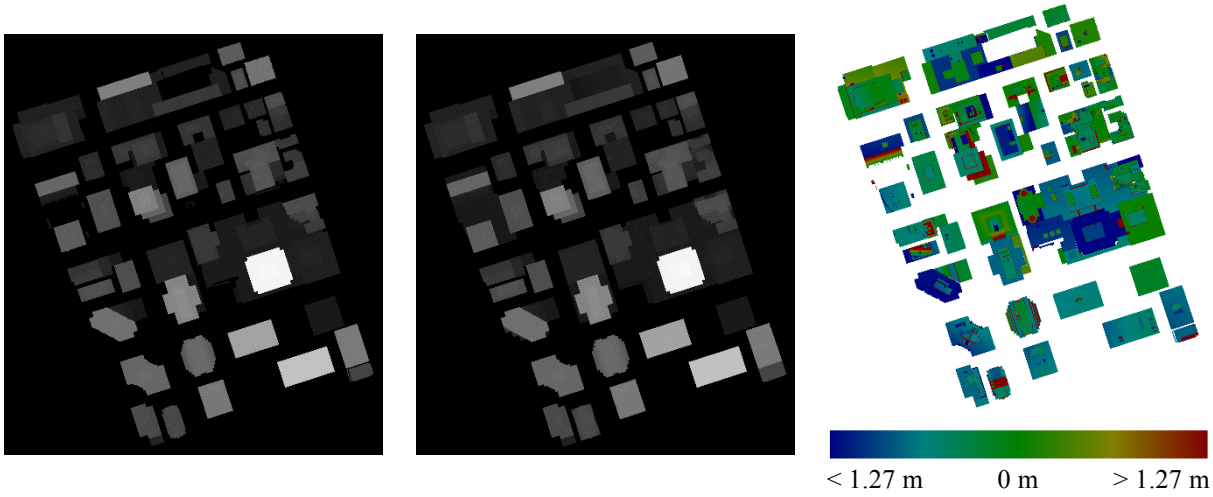


Figure 8.58. Digital surface models derived from the roof planes of the reference (left), from the roof planes of the result (middle), and the difference of these models where planes were found in both data sets (right).

Due to the challenges of AOI 5, the number of submitted fully and semi-automatically reconstructed building models is lower than for all other AOIs. As can be seen in Table 8.11, the automatically reconstructed building models of the developed reconstruction approach TUB provides with 84.7 % a significantly higher completeness than the others (14.4 % higher than the second best completeness value). By taking into account the difficulties of AOI 5, it can be stated that the completeness of TUB is rather high. If only roof planes larger than 10 m² are considered, it is increased to 96.1 % which is still 7.9 % higher than the second best approach. The correctness of TUB is, however, only average but not significantly lower than the best achieved correctness. In general, the number of automatically reconstructed roof planes that do not feature a 1:1 relation to the reference roof planes is for TUB not significantly greater than for the other reconstruction approaches if the high completeness value is taken into account. However, the large number of N:1 relations indicates that several building details are still missing in the automatic reconstructed building models.

Table 8.11. Evaluation result (TUB) of AOI 5 in comparison to other evaluated (fully/ semi-automatic) reconstruction approaches presented in (ISPRS, 2017).

Abbrev.	Comp [%]	Corr [%]	$Comp_{10}$ [%]	$Corr_{10}$ [%]	Topo 1:N	Topo N:1	Topo N:M	RMS_{XY} [m]
CKU	70.3	83.3	85.2	83.6	11	45	42	1.8
YOR	64.5	**85.8**	86.8	**86.6**	**4**	58	**24**	**0.9**
MON2	67.9	80.7	88.2	84.3	96	**26**	57	1.1
TUB	**84.7**	82.2	**96.1**	86.0	17	61	35	1.0

In summary, the automatic reconstruction of AOI 5 is the most challenging task with respect to all selected AOIs because it consists of several high-rise buildings that feature many

complex roof structures. Furthermore, these high-rise buildings are often connected by low-storey buildings to large building complexes why their complete reconstruction often becomes difficult. However, the same applies to the manually reconstructed building models which seems to be not always complete or correct. Therefore, by taking the manual effort and the quality of the reference into account, the proposed reconstruction approach still provides good results.

9. Conclusions and Outlook

The automatic reconstruction of 3D city models has become a topic of increasing importance, as they already assume for quite some time a central role in a wide variety of applications. Particularly the reconstruction of building models, as one of the major components of 3D city models, has proven to be difficult. Therefore, there is a great demand for an automatic building reconstruction approach. In this thesis, a new approach to automatically reconstruct 3D building models from ALS data has been proposed. It combines the benefits of data-driven (flexibility and completeness) and model-driven (robustness and visual attractiveness) reconstruction approaches and is able to provide not only geometric but also semantic information. The presented hybrid reconstruction approach consists of three major steps: (i) segmentation of roof planes, (ii) derivation of building knowledge, and (iii) construction of 3D building models.

For the first step, a new method was proposed that is especially designed to segment sufficiently planar roof surfaces. It was implemented as an extension of the well-known surface growing method. But in contrast to surface growing, segments are in the extension additionally able to grow below other segments. It was shown that this segmentation strategy provides for the subsequent steps of the presented reconstruction process several advantages, such as: (i) Holes in roofs that are caused by other overlying roofs are automatically closed in the resulting segments. (ii) Segment patches that belong to the same roof plane but that are separated from one another due to superstructures or crossing roof parts are merged. Thereby, the following problem is solved: If the complexity of roof structures increases, then segments usually increase in number, while becoming smaller, and at the same time more intricate in their shapes. (iii) Merged segments imply a higher accuracy of their estimated plane parameters as the points extend over a larger area. (iv) Consistency is automatically established between nearly coplanar segments that belong to the same roof plane. (v) Segments are closer to each other, share longer common boundaries or even intersect one another so that a reliable detection and precise tracing of intersection and step lines is enabled at a later stage. (vi) Points of small roof details that would usually not make up segments of their own are part of larger segments so that their information is maintained in the reconstruction process. (vii) Segment adjacencies below roofs are detected as well so that roof configurations below roof surfaces can be incorporated at a later stage.

In order to take into account that roof surfaces might be only inadequately represented by the determined segments, building models were not directly constructed from them. Instead it was proposed to first derive building knowledge which is then in addition to the segments used in the construction of models. For the automatic derivation of building knowledge, a multi-scale knowledge graph was introduced as a model for knowledge representation and object categorization. In it building features, their relationships, and their interdependencies are best organized and represented. Furthermore, in order to derive building knowledge in the

different levels of the multi-scale knowledge graph, a graph grammar was proposed which is alternately applied in bottom-up and top-down phases. The experiments showed that the application of the graph grammar in this manner together with the multi-scale knowledge graph enables a reliable derivation of building knowledge. Additionally, it was demonstrated that the presented graph grammar is a good tool to describe a large set of valid building configurations in a very compact way. Due to its expressive power, the developed graph grammar can be easily adapted to different purposes (e.g., to obtain only specific building types or LOD) so that the transferability of the whole building knowledge derivation procedure generally is very good. Furthermore, it was demonstrated how reinforcement learning can be applied to support the detection of repetitive building features during the grammar-guided derivation of building knowledge. The experiments showed that the applied reinforcement learning technique significantly supports the recognition and reconstruction of symmetric building parts.

In the final step, half-space modeling was proposed for the construction of 3D building models. For this, building knowledge was used to define half-spaces whose hyperplanes coincide with the determined segments. It was shown that the presented construction strategy provides several advantages, such as: (i) Partly occluded building components are completely constructed. (ii) Building components that are combined to define more complex buildings do not need to be disjoint and may intersect just like the segments obtained from sub-surface segmentation. (iii) The number of half-spaces needed to construct a complex building is rather low so that buildings can be defined in a compact way. (iv) The resulting building models are always closed solids. (v) All building types can be represented in canonical form. (vi) The reuse of hyperplanes in connected building components provides a high degree of flexibility if one component needs to be adjusted. It can be thus concluded that half-space modeling automatically solves several problems that usually occur during the direct construction of B-rep models. If B-rep models are needed, building models defined by half-spaces can be directly converted with a common CAD kernel to B-rep models that are always guaranteed to be closed.

In order to improve the resulting building models, two extensions of the presented reconstruction workflow were introduced in the construction step. One extension was introduced to add further details to the building models. It was demonstrated that the proposed ICP based approach is able to automatically recognize and reconstruct even in low-density point clouds building details that usually would otherwise be discarded. A major limitation of it is, however, that it can be only used for roof details that feature several instances of the same shape on the same roof. The other extension was introduced to improve the geometry of the resulting building models so that they become more accurate and pleasing to the human eye. It was shown that the proposed local and global regularization rules are well suited to support frequently occurring regularities in the building models. Depending on the intended use of the resulting building models, parameters of the divisive clustering procedure can be adjusted so that the building models either resemble more closely the input data or become more regularized.

In addition to the proposed automatic reconstruction workflow, it was also shown how building knowledge can be integrated into an existing data-driven reconstruction approach to improve the regularization of the building models without lessening the flexibility to generate all roof shapes occurring in the real-world. It was thereby proven that the presented graph grammar in combination with the multi-scale knowledge graph is in its use not limited to the proposed reconstruction workflow. The experiments with the selected BSP based data-driven reconstruction approach showed that the proposed concept of hyperpolylines, which are defined based on building knowledge, can significantly reduce the number of partitions of the partitioned space. This simplifies the merging of partitions considerably. Thus, it can be concluded that the automatic derivation and incorporation of building knowledge is essential for the improvement of automatically reconstructed building models.

The application of the proposed grammar-guided reconstruction approach on the five AOIs of the ISPRS benchmark dataset has demonstrated that it is effective for the reconstruction of 3D building models from aerial LiDAR data. Moreover, it was proven that all research objectives (accuracy, automation, complexity, efficiency, quality, resolution, robustness, and transferability) set out in section 1.3 of this thesis have been achieved. The evaluation of the automatically reconstructed building models confirmed that the proposed approach is in many respects better suited for the reconstruction of buildings than other state-of-the-art reconstruction approaches. Particularly the incorporation of automatically derived building knowledge significantly helped to avoid the construction of unrealistic building parts. Furthermore, the multi-scale knowledge graph in combination with the graph grammar offers the possibility not only to improve the geometry of building models but also to enhance them with semantic information. The increased demand for semantically enriched building models is therefore met by the proposed reconstruction method. However, during the experiments some deficiencies have been encountered that are mainly related to the details of very complex rooftops or to the reconstruction of small roof superstructures. The evaluation showed that similar deficiencies occurred in the manually reconstructed reference building models. Although the automatically obtained models do not perfectly agree with these reference models, it can be concluded that the automatically reconstructed building models represent the real buildings very well in terms of to the applied evaluation metrics.

Despite the fact that the proposed reconstruction approach provides good results, some recommendations for future research to improve the proposed reconstruction approach and the resulting building models can be made: (i) The proposed reconstruction workflow should be extended to also reconstruct non-planar building shapes such as spheres, cylinders, cones, etc. For this, the sub-surface segmentation needs to be enhanced so that continuous surfaces can be segmented as well. The subsequent reconstruction steps can be easily adapted by making minor modifications. This basically means that additional production rules need to be defined for the building knowledge derivation step and hypersurfaces instead of hyperplanes need to be used for the model construction. (ii) To further improve the quality and accuracy of the resulting 3D building models, additional data sources (e.g., terrestrial laser scanning data, images, outdated building models, etc.) should be optionally incorporated. (iii) A drawback of the proposed approach is that the building knowledge derivation procedure highly depends on

production rules that need to be defined in advance. In future work, methods should be developed that are able to automatically define these production rules either in advance or during the derivation of building knowledge. (iv) The proposed building reconstruction method requires many parameters whose values are adapted during the reconstruction process (e.g., through reinforcement learning). In order to reduce the computational time, further methods should be developed that also automatically initialize the parameters with reasonable values depending on the input data and the prior knowledge of the area to be reconstructed. (v) The three major steps of the proposed reconstruction method have been implemented as separate modules so that any of them can be replaced by another suitable procedure. A greater interaction between these modules could, however, be beneficial. Specifically, if the results of a module are not conclusive, then the procedure should be able to return to a previous module in order to further improve the results. For example, if a recognized building feature indicates a missing segment, it should be able to activate a local segmentation with specific parameter values.

Bibliography

Abdullah, S. M., Awrangjeb, M., Lu, G., 2014. LiDAR Segmentation Using Suitable Seed Points for 3D Building Extraction. In: *ISPRS Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, XL-3, pp. 1-8.

Abelson, H., diSessa, A. A., 1981. Turtle Geometry: The Computer as a Medium for Exploring Mathematics. The MIT Press, Cambridge London, USA England.

Aggarwal, C. C., Reddy, C. K., 2014. Data Clustering: Algorithms and Applications. Chapman and Hall/CRC, Boca Raton, FL, USA.

Agoston, M. K., 2005. Computer Graphics and Geometric Modeling: Implementation and Algorithms. Springer-Verlag, London, England.

Aichholzer, O., Aurenhammer, F., Alberts, D., Gärtner, B., 1995. A Novel Type of Skeleton for Polygons. In: Journal of Universal Computer Science, 1(12), pp. 752-761.

Albers, B., Kada, M., Wichmann, A., 2016. Automatic Extraction and Regularization of Building Outlines from Airborne LiDAR Point Clouds. In: *ISPRS Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, XLI-B3, pp. 555-560.

Alexa, M., 2002. Recent Advances in Mesh Morphing. In: *Computer Graphics Forum*, 21(2), pp. 173-198.

Alharthy, A., Bethel, J., 2004. Detailed Building Reconstruction from Airborne Laser Data Using a Moving Surface Method. In: *ISPRS Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences*, XXXV-B3, pp. 213-218.

Aliaga, D. G., Rosen, P. A., Bekins, D. R., 2007. Style Grammars for Interactive Visualization of Architecture. In: *IEEE Transactions on Visualization and Computer Graphics*, 13(4), pp. 786-797.

Ameri, B., Fritsch, D., 2000. Automatic 3D Building Reconstruction Using Plane-Roof Structures. In: *Proceedings of the American Society of Photogrammetry and Remote Sensing Conference*, Washington, D.C., USA.

Andrew, A. M., 1979. Another Efficient Algorithm for Convex Hulls in Two Dimensions. In: *Information Processing Letter*, 9(5), pp. 216-219.

Arun, K. S., Huang, T. S., Blostein, S. D., 1987. Least Square Fitting of Two 3-D Point Sets. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 9(5), pp. 698-700.

Awodey, S., 2010. Category Theory (2nd edition). In: Macintyre, A. J., Scott, D. S. (eds.), *Oxford Logic Guides 52*, Oxford University Press, Oxford, England.

Awrangjeb, M., Fraser, C. S., 2014. Automatic Segmentation of Raw LiDAR Data for Extraction of Building Roofs. In: *International Journal of Remote Sensing*, 6(5), pp. 3716-3751.

Ayhan, D., Sağlam, Ş., 2012. A Technical Review of Building-Mounted Wind Power Systems and a Sample Simulation Model. In: *Renewable and Sustainable Energy Reviews*, 16(1), pp. 1040-1049.

Baraldi, A., Blonda, P., 1999. A Survey of Fuzzy Clustering Algorithms for Pattern Recognition (Part I+II). In: *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, 29(6), pp. 778-801.

Barequet, G., Kumar, S., 1997. Repairing CAD Models. In: *Proceedings of the 8th Conference on Visualization*, pp. 363-370.

Baumgart, B. G., 1972. Winged Edge Polyhedron Representation. In: *Technical Report STAN-CS·320*, Computer Science Department, Stanford University, Palo Alto, CA, USA.

Baumgart, B. G., 1975. A Polyhedron Representation for Computer Vision. In: *Proceedings of the National Computer Conference and Exposition*, pp. 589-596.

Becker, S., 2009. Generation and Application of Rules for Quality Dependent Façade Reconstruction. In: *ISPRS Journal of Photogrammetry and Remote Sensing*, 64(6), pp. 640-653.

Becker, S., Peter, M., Fritsch, D., 2015. Grammar-Supported 3D Indoor Reconstruction from Point Clouds for "As-Built" BIM. In: *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, II-3/W4, pp. 17-24.

Becker, S., Peter, M., Fritsch, D., Philipp, D., Baier, P., Dibak, C., 2013. Combined Grammar for the Modeling of Building Interiors. In: *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, II-4/W1, pp. 1-6.

Bekins, D., Aliaga, D. G., 2005. Build-by-Number: Rearranging the Real World to Visualize Vovel Architectural Spaces. In: *Proceedings of the 16th IEEE Conference on Visualization*, pp. 143-150.

Bentley, J. L., 1975. Multidimensional Binary Search Trees Used for Associative Searching. In: *Communications of the ACM*, 18(9), pp. 509-517.

Besl, P. J., McKay, N. D., 1992. A Method for Registration of 3-D Shapes. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 14(2), pp. 239-256.

Biljecki, F., Ledoux, H., Stoter, J., 2016a. Generation of Multi-LOD 3D City Models in CityGML with the Procedural Modelling Engine Random3DCity. In: *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, IV-4/W1, pp. 51-59.

Biljecki, F., Ledoux, H., Stoter, J., Vosselman, G., 2016b. The Variants of an LOD of a 3D Building Model and Their Influence on Spatial Analyses. In: *ISPRS Journal of Photogrammetry and Remote Sensing*, 116, pp. 42-54.

Biljecki, F., Stoter, J., Ledoux, H., Zlatanova, S., Çöltekin, A., 2015. Applications of 3D City Models: State of the Art Review. In: *ISPRS International Journal of Geo-Information*, 4(4), pp. 2842-2889.

Bogdanovich, P., Samet, H., 1999. The ATree: A Data Structure to Support Very Large Scientific Databases. In: Agouris, P., Stefanidis , A. (eds.), *Integrated Spatial Databases: Digital Images and GIS*, Springer-Verlag, Berlin Heidelberg, Germany, pp. 235-248.

Bokeloh, M., Wand, M., Seidel, H.-P., 2010. A Connection Between Partial Symmetry and Inverse Procedural Modeling. In: *ACM Transactions on Graphics*, 29(4), pp. 104:1-104:10.

Bradbury, J., Ziebart, M., Cross, P. A., Boulton, P., Read, A., 2007. Code Multipath Modelling in the Urban Environment Using Large Virtual Reality City Models: Determining the Local Environment. In: *The Journal of Navigation*, 60(1), pp. 95-105.

Braden, B., 1986. The Surveyor's Area Formula. In: *The College Mathematics Journal*, 17(4), pp. 326-337.

Brédif, M., Boldo, D., Pierrot-Deseilligny, M., Maître, H., 2007. 3D Building Reconstruction with Parametric Roof Superstructures. In: *Proceedings of the IEEE International Conference in Image Processing*, 2, pp. 537-540.

Brenner, C., 2000. Towards Fully Automatic Generation of City Models. In: *ISPRS Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, XXXIII-B3, pp. 85-92.

Brenner, C., 2004. Building Reconstruction from Images and Laser Scanning. In: *International Journal of Applied Earth Observation and Geoinformation*, 6(3-4), pp. 187-198.

Brenner, C., 2005. Building Reconstruction from Images and Laser Scanning. In: *International Journal of Applied Earth Observation and Geoinformation*, 6(3-4), pp. 187-198.

Brenner, C., 2010. Building Extraction. In: Vosselman, G., Maas, H.-G. (eds.), *Airborne and Terrestrial Laser Scanning*, Whittles Publishing, Dunbeath, Scotland, UK, pp. 169-212.

Bretar, F., Roux, M., 2005. Hybrid Image Segmentation Using LiDAR 3D Planar Primitives. In: *ISPRS Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, XXXVI-3/W19, pp. 72-78.

Buron, C., Marvie, J.-E., Gautron, P., 2013. GPU Roof Grammars. In: *Eurographics 2013: Short Papers*, pp. 57-60.

Cappelle, C., El Najjar, M. E., Charpillet, F., Pomorski, D., 2012. Virtual 3D City Model for Navigation in Urban Areas. In: *Journal of Intelligent & Robotic Systems*, 66(3), pp. 337-399.

Chen, D., Zhang, L., Li, J., Liu, R., 2012. Urban Building Roof Segmentation from Airborne LiDAR Point Clouds. In: *International Journal of Remote Sensing*, 33(20), pp. 6497-6515.

Chen, G., Esch, G., Wonka, P., Müller, P., Zhang, E., 2008. Interactive Procedural Street Modeling. In: *ACM Transactions on Graphics*, 27(3), pp. 103:1-103:10.

Chen, R., 2011. The Development of 3D City Model and Its Applications in Urban Planning. In: *Proceedings of the 19th International Conference on Geoinformatics*, pp. 1-5.

Chen, Y., Medioni, G., 1991. Object Modeling by Registration of Multiple Range Images. In: *Proceedings of the IEEE International Conference on Robotics and Automation*, pp. 2724-2729.

Chin, N., Feiner, S., 1989. Near Real-Time Shadow Generation Using BSP Trees. In: *ACM SIGGRAPH Computer Graphics*, 23(3), pp. 99-106.

Chomsky, N., 1956. Three models for the Description of Language. In: *IRE Transactions on Information Theory*, 2(3), pp. 113-124.

Chomsky, N., 1957. Syntactic Structures. Mouton Publishers, The Hague, Netherlands.

Chomsky, N., 1959. On Certain Formal Properties of Grammars. In: *Information and Control*, 2(2), pp. 137-167.

Chow, C. K., Tsui, H. T., Lee, T., 2004. Surface Registration Using a Dynamic Genetic Algorithm. In: *Pattern Recognition*, 37(1), pp. 105-117.

City of Berlin, 2017. Stadtmodelle – Digitale Innenstadt: Das 3D-Modell. http://www.stadtentwicklung.berlin.de/planen/stadtmodelle/de/digitale_innenstadt/3d/index.shtml, last visited: 14.02.2018.

City of New York, 2017. Building Footprints – Shapefile of Footprint Outlines of Building in New York City. http://opendata.cityofnewyork.us/data, last visited: 14.02.2018.

City of Toronto, 2016. 3D Massing. https://www.toronto.ca/city-government/data-research-maps/open-data/open-data-catalogue/#db07630f-252d-f7ae-2dff-8d0b38ec6576, last visited: 14.02.2018.

Coelho, A., Bessa, M., Augusto Sousa, A., Nunes Ferreira, F., 2007. Expeditious Modelling of Virtual Urban Environments with Geospatial L-Systems. In: *Computer Graphics Forum*, 26(4), pp. 769-782.

Coors, V., Huch, T., Kretschmer, U., 2000. Matching Buildings: Pose Estimation in an Urban Environment. In: *Proceedings of the IEEE and ACM International Symposium on Augmented Reality (ISAR)*, pp. 89-92.

Corradini, A., Montanari, U., Rossi, F., Ehrig, H., Heckel, R., Löwe, M., 1997. Algebraic Approaches to Graph Transformation – Part I: Basic Concepts and Double Pushout Approach. In: Rozenberg, G. (ed.), *Handbook of Graph Grammars and Computing by Graph Transformation, Volume 1: Foundations*, World Scientific Publishing Co. Pte. Ltd., Singapore, pp. 163-246.

Coughlan, J. M., Yuille, A., L., 1999. Manhattan World: Compass Direction from a Single Image by Bayesian Inference. In: *Proceedings of the International Conference on Computer Vision (ICCV)*, pp. 941-947.

Courcelle, B., 1990. Graph Rewriting: An Algebraic and Logic Approach. In: van Leeuwen (ed.), *Handbook of Theoretical Computer Science, Volume B: Formal Models and Semantics*, pp. 193-242.

Cramer, M., 2010. The DGPF-Test on Digital Airborne Camera Evaluation – Overview and Test Design. In: *Photogrammetrie – Fernerkundung – Geoinformation*, 2010(2), pp. 73-82.

Czerwinski, A., Sandmann, S., Stöcker-Meier, E., Plümer, L., 2007. Sustainable SDI for EU Noise Mapping in NRW – Best Practice for INSPIRE. In: *International Journal for Spatial Data Infrastructures Research*, 2, pp. 90-111.

Czyńska, K., Rubinowicz, P., 2014. Application of 3D Virtual City Models in Urban Analyses of Tall Buildings – Today Practice and Future Challenges. In: *Architecturae et Artibus*, 6(1), pp. 9-13.

Danahy, J., 1999. Visualization Data Needs in Urban Environmental Planning and Design. In: Fritsch, D., Spiller, R. (eds.), *Photogrammetric Week '99*, Wichmann Verlag, Heidelberg, pp. 351-365.

De Berg, M., Cheong, O., van Kreveld, M., Overmars, M., 2008. Computational Geometry: Algorithms and Applications (3rd edition). Springer-Verlag, Berlin Heidelberg, Germany.

De Oliveira, J. V., Pedrycz, W., 2007. Advances in Fuzzy Clustering and Its Applications. John Wiley & Sons, Ltd., Chichester, England.

Dehbi, Y., Gröger, G., Plümer, L., 2016a. Identification and Modelling of Translational and Axial Symmetries and their Hierarchical Structures in Building Footprints by Formal Grammars. In: *Transactions in GIS*, 20(5), p. 645-663.

Dehbi, Y., Hadiji, F., Gröger, G., Kersting, K., Plümer, L., 2017. Statistical Relational Learning of Grammar Rules for 3D Building Reconstruction. In: *Transactions in GIS*, 21(1), pp. 134-150.

Dehbi, Y., Plümer, L., 2011. Learning Grammar Rules of Building Parts from Precise Models and Noisy Observations. In: *ISPRS Journal of Photogrammetry and Remote Sensing*, 66(2), pp. 166-176.

Dehbi, Y., Staat, C., Mandtler, L., Plümer, L., 2016b. Incremental Refinement of Facade Models with Attribute Grammar from 3D Point Clouds. In: *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, III-3, pp. 311-316.

Díez, Y., Roure, F., Lladó, X., Salvi, J., 2015. A Qualitative Review on 3D Coarse Registration Methods. In: *ACM Computing Surveys*, 47(3), pp. 45:1-45:36.

Dokter, M., 2014. Deriving Shape Grammars on the GPU. In: *Proceedings of the 18th Central European Seminar on Computer Graphics*, pp. 49-56.

Döllner, J., Kolbe, T. H., Liecke, F., Sgouros, T., Teichmann, K., 2006. The Virtual 3D City Model of Berlin – Managing, Integrating and Communicating Complex Urban Information. In: *Proceedings of the 25th International Symposium on Urban Data Management (UDMS)*.

Dornaika, F., Brédif, M., 2008. An Efficient Approach to Building Superstructure Reconstruction Using Digital Elevation Maps. In: *ISPRS Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, XXXVII-B3a, pp. 179-184.

Dorninger, P., Nothegger, C., 2007. 3D Segmentation of Unstructured Point Clouds for Building Modelling. In: *ISPRS Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, XXXVI-3/W49A, pp. 191-196.

Dorninger, P., Pfeifer, N., 2008. A Comprehensive Automated 3D Approach for Building Extraction, Reconstruction, and Regularization from Airborne Laser Scanning Point Clouds. In: *Sensors*, 8(11), pp. 7323-7343.

Douglas, D., Peucker, T., 1973. Algorithms for the Reduction of the Number of Points Required to Represent a Digitized Line or Its Caricature. In: *The Canadian Cartographer*, 10(2), pp. 112-122.

Drewes, F., Habel, A., Kreowski, H.-J., 1997. Hyperedge Replacement Graph Grammars. In: Rozenberg, G. (ed.), *Handbook of Graph Grammars and Computing by Graph Transformation, Volume 1: Foundations*, World Scientific Publishing Co. Pte. Ltd., Singapore, pp. 95-162.

Duckham, M., Kulik, L., Worboys, M., Galton, A., 2008. Efficient Generation of Simple Polygons for Characterizing the Shape of a set of Point in the Plane. In: *Pattern Recognition*, 41(10), pp. 3224-3236.

Dylla, K., Frischer, B., Müller, P., Ulmer, A., Haegler, S., 2009. Rome Reborn 2.0: A Case Study of Virtual City Reconstruction Using Procedural Modeling Techniques. In: *Proceedings of the 37th International Conference in Computer Applications and Quantitative Methods in Archaeology (CAA)*, pp. 62-66.

Edelsbrunner, H., Kirkpatrick, D., Seidel, R., 1983. On the Shape of a Set of Points in the Plane. In: *IEEE Transactions on Information Theory*, 29(4), pp. 551-559.

Edelsbrunner, J., Krispel, U., Havemann, S., Sourin, A., Fellner, D. W., 2016. Constructive Roofs from Solid Building Primitives. In: Gacrilova, M. L., Kenneth Tan, C. J. (eds.), *Transactions on Computational Science XXVI*, Springer-Verlag, Berlin Heidelberg, Germany, pp. 17-40.

Ehrig, H., Ehrig, K., Prange, U., Taentzer, G., 2006. Fundamentals of Algebraic Graph Transformation. Springer-Verlag, Berlin Heidelberg, Germany.

Ehrig, H., Heckel, R., Korff, M., Löwe, M., Ribeiro, L., Wagner, A., Corradini, A., 1997. Algebraic Approaches to Graph Transformation – Part II: Single Pushout Approach and Comparison with Double Pushout Approach. In: Rozenberg, G. (ed.), *Handbook of Graph Grammars and Computing by Graph Transformation, Volume I: Foundations*, World Scientific Publishing Co. Pte. Ltd., Singapore, pp. 247-312.

Ehrig, H., Heckel, R., Llabrés, M., Orejas, F., Padberg, J., Rozenberg, G., 1998. Double-Pullback Graph Transitions: A Rule-Based Framework with Incomplete Information. In: *Lecture Notes in Computer Science*, 1764, Springer-Verlag, Berlin Heidelberg, Germany, pp. 85-102.

Ehrig, H., Pfender, M., Scheider, H. J., 1973. Graph-Grammars: An Algebraic Approach. In: *Proceedings of the 14th Annual Symposium on Switching and Automata Theory*, pp. 167-180.

Eicker, U., Nouvel, R., Duminil, E., Coors, V., 2013. Assessing Passive and Active Solar Energy Resources in Cities Using 3D City Models. In: *Energy Procedia*, 57, pp. 896-905.

Elias, R., 2014. Digital Media: A Problem-Solving Approach for Computer Graphics. Springer-Verlag, Cham, Switzerland.

Engelfriet, J., Rozenberg, G., 1997. Node Replacement Graph Grammars. In: Rozenberg, G. (ed.), *Handbook of Graph Grammars and Computing by Graph Transformation, Volume I: Foundations*, World Scientific Publishing Co. Pte. Ltd., Singapore, pp. 1-94.

Everitt, B. S., Landau, S., Leese, M., Stahl, D., 2011. Cluster Analysis (5th edition). John Wiley & Sons, Ltd., Chichester, England.

Fan, H., Yao, W., Fu, Q., 2014. Segmentation of Sloped Roofs from Airborne LiDAR Point Clouds Using Ridge-Based Hierarchical Decomposition. In: *International Journal of Remote Sensing*, 6(4), pp. 3284-3301.

Finkel, R. A., Bentley, J. L., 1974. Quad Trees: A Data Structure for Retrieval on Composite Keys. In: *Acta Informatica*, 4(1), pp. 1-9.

Fischler, M. A., Bolles, R. C., 1981. Random Sample Consensus: A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography. In: *Communications of the ACM*, 24(6), pp. 381-395.

Florek, K., Lukaszewicz, J., Perkal, J., Steinhaus, H., Zubrzycki, S., 1951. Sur la Liaison et la Division des Points d'un Ensemble Fini. In: *Colloquium Mathematicae*, 2(3-4), pp. 282-285.

Foley, J. D., van Dam, A., Feiner, S., Hughes, J., 1990. Computer Graphics: Principles and Practice (2nd edition). Addison-Wesley, Boston, MA, USA.

Foody, G. M., 2002. Status of Land Cover Classification Accuracy Assessment. In: *Remote Sensing of Environment*, 80(1), pp. 185-201.

Forlani, G., Nardinocchi, C., Scaioni, M., Zingaretti, P., 2003. Building Reconstruction and Visualization from LiDAR Data. In: *ISPRS Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, XXXIV-5/W12, pp. 151-156.

Forlani, G., Nardinocchi, C., Scaioni, M., Zingaretti, P., 2006. Complete Classification of Raw LiDAR Data and 3D Reconstruction of Buildings. In: *Pattern Analysis and Applications*, 8(4), pp. 357-374.

Fuchs, H., Kedem, Z. M., Naylor, B. F., 1980. On Visible Surface Generation by a Priori Tree Structures. In: *ACM SIGGRAPH Computer Graphics*, 14(3), pp. 124-133.

Gan, G., Ma., C., Wu, J., 2007. Data Clustering: Theory, Algorithms, and Applications. Society for Industrial and Applied Mathematics, Philadelphia, PA, USA.

Garey, M. R., Johnson, D. S., 1979. Computers and Intractability: A Guide to the Theory of NP-Completeness. W.H. Freeman and Company, New York, NY, USA.

Geibel, R., Stilla, U., 2000. Segmentation of Laser Altimeter Data for Building Reconstruction: Different Procedures and Comparison. In: *ISPRS Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, XXXIII-B3, pp. 326-334.

Gips, J., 1999. Computer Implementation of Shape Grammars. In: *Workshop on Shape Computation*, Massachusetts Institute of Technology.

Glander, T., Döllner, J., 2009. Abstract Representation for Interactive Visualization of Virtual 3D City Models. In: *Computers, Environment and Urban Systems*, 33(5), pp. 375-387.

Gonzalez, R. C., Woods, R. E., 2008. Digital Image Processing (3rd edition). Pearson Prentice Hall, Upper Saddle River, NJ, USA.

Gröger, G., Kolbe, T. H., Czerwinski, A., Nagel, C., 2008. OpenGIS City Geography Markup Language (CityGML) Encoding Standard, Version 1.0.0, OGC Doc. No. 08-077r1. http://www.opengeospatial.org/standards/citygml, last visited: 14.02.2018.

Gröger, G., Kolbe, T. H., Nagel, C., Häfele, K.-H., 2012. OGC City Geography Markup Language (CityGML) Encoding Standard, Version 2.0.0, OGC Doc. No. 12-019. http://www.opengeospatial.org/standards/citygml, last visited: 14.02.2018.

Gröger, G., Plümer, L., 2010. Derivation of 3D Indoor Models by Grammars for Route Planning. In: *Photogrammetrie – Fernerkundung – Geoinformation*, 2010(3), pp. 191-206.

Gröger, G., Plümer, L., 2012. CityGML – Interoperable Semantic 3D City Models. In: *ISPRS Journal of Photogrammetry and Remote Sensing*, 71, pp. 12-33.

Grün, A., Akca, D., 2005. Least Squares 3D Surface and Curve Matching. In: *ISPRS Journal of Photogrammetry and Remote Sensing*, 59(3), pp. 151-174.

Guibas, L., Stolfi, J., 1985. Primitives for the Manipulation of General Subdivisions and the Computation of Voronoi Diagrams. In: *ACM Transactions on Graphics*, 4(2), pp. 74-123.

Haala, N., Brenner, C., 1999. Extraction of Buildings and Trees in Urban Environments. In: *ISPRS Journal of Photogrammetry and Remote Sensing*, 54(2-3), pp. 130-137.

Haala, N., Hastedt, H., Wolf, K., Ressl, C., Baltrusch, S., 2010. Digital Photogrammetric Camera Evaluation – Generation of Digital Elevation Models. In: *Photogrammetrie – Fernerkundung – Geoinformation*, 2010(2), pp. 99-115.

Haala, N., Kada, M., 2010. An Update on Automatic 3D Building Reconstruction. In: *ISPRS Journal of Photogrammetry and Remote Sensing*, 65(6), pp. 570-580.

Habel, A., 1992. Hyperedge Replacement: Grammars and Languages. In: *Lecture Notes in Computer Science,* 643, Springer-Verlag, Berlin, Germany.

Habib, A. F., Zhai, R., Kim, C., 2010. Generation of Complex Polyhedral Building Models by Integrating Stereo-Aerial Imagery and LiDAR Data. In: *Photogrammetric Engineering & Remote Sensing*, 76(5), pp. 609-623.

Haegler, S., Wonka, P., Müller Arisona, S., van Gool, L., Müller, P., 2010. Grammar-based Encoding of Facades. In: *Computer Graphics Forum*, 29(4), pp. 1479-1487.

Hampel, F. R., Ronchetti, E. M., Rousseeuw, P. J., Stahel, W. A., 1986. Robust Statistics. John Wiley and Sons, New York, NY, USA.

Havemann, S., 2005. Generative Mesh Modeling. PhD thesis, Technical University Braunschweig, Germany.

Henn, A., Gröger, G., Stroh, V., Plümer, L., 2013. Model Driven Reconstruction of Roofs from Sparse LiDAR Point Clouds. In: *ISPRS Journal of Photogrammetry and Remote Sensing*, 76, pp. 17-29.

Hoffmann, C. M., 1989. Geometric and Solid Modeling. Morgan Kaufmann Publishers, San Mateo, CA, USA.

Hohmann, B., Krispel, U., Havemann, S., Fellner, D., 2009. Cityfit: High-Quality Urban Reconstructions by Fitting Shape Grammars to Images and Derived Textured Point Clouds. In: *ISPRS Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences,* XXXVIII-5/W1.

Höppner, F., Klawonn, F., Kruse, R., Runkler, T., 1999. Fuzzy Cluster Analysis: Methods for Classification, Data Analysis and Image Recognition. John Wiley & Sons, Ltd., New York, NY, USA.

Horn, B. K. P., Hilden, H. M., Negahdaripour, S., 1988. Closed-Form Solution of Absolute Orientation Using Orthonormal Matrices. In: *Journal of the Optical Society of America - A*, 5(7), pp. 1127-1135.

Hotelling, H., 1933. Analysis of a Complex of Statistical Variables Into Principal Components. In: *Journal of Educational Psychology*, 24, pp. 417-441 and pp. 498-520.

Hough, P. V. C., 1962. Methods and Means for Recognizing Complex Patterns. U.S. Patent 3069654.

Hsu, L.-T., Gu, Y., Kamijo, S., 2015. NLOS Correction/Exclusion for GNSS Measurement Using RAIM and City Building Models. In: *Sensors*, 15(7), pp. 17329-17349.

Huang, H., Brenner, C., 2011. Rule-Based Roof Plane Detection and Segmentation from Laser Point Clouds. In: *Proceedings of the Joint Urban Remote Sensing*, pp. 293-296.

Huang, H., Brenner, C., 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*, pp. 16-24.

Huang, H., Brenner, C., Sester, M., 2013. A Generative Statistical Approach to Automatic 3D Building Roof Reconstruction from Laser Scanning Data. In: *ISPRS Journal of Photogrammetry and Remote Sensing*, 79, pp. 29-43.

Huber, P. J., 1981. Robust Statistics. John Wiley and Sons, New York, NY, USA.

Huffman, D. A., 1952. A Method for the Construction of Minimum-Redundancy Codes. In: *Proceedings of the IRE*, 40(9), pp. 1098-1101.

ISPRS, 2017. ISPRS Test Project on Urban Classification and 3D Building Reconstruction: Results. http://www2.isprs.org/commissions/comm3/wg4/results.html, last visited: 14.02.2018.

Jackins, C. L., Tanimoto, S. L., 1980. Oct-Trees and Their Use in Representing Three-Dimensional Objects. In: *Computer Graphics and Image Processing*, 14(3), pp. 249-270.

Jarzabek-Rychard, M., 2012. Reconstruction of Building Outlines in Dense Urban Areas Based on LiDAR Data and Address Points. In: *ISPRS Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, XXXIX-B3, pp. 121-126.

Jarząbek-Rychard, M., Borkowski, A., 2016. 3D Building Reconstruction from ALS Data Using Unambiguous Decomposition Into Elementary Structures. In: *ISPRS Journal of Photogrammetry and Remote Sensing*, 118, pp. 1-12.

Jarząbek-Rychard, M., Maas, H.-G., 2017. Geometric Refinement of ALS-Data Derived Building Models Using Monoscopic Aerial Images. In: *Remote Sensing*, 9(3), pp. 282:1-282:16.

Jiang, J., Cheng, J., Chen, X., 2009. Registration for 3-D Point Cloud Using Angular-Invariant Feature. In: *Neurocomputing*, 72(16-18), pp. 3839-3844.

Jolliffe, I. T., 2002. Principle Component Analysis. Springer-Verlag, New York, NY, USA.

Ju, T., Losasso, F., Schaefer, S., Warren, J., 2002. Dual Contouring of Hermite Data. In: *Proceedings of the 29th International Conference on Computer Graphics and Interactive Techniques (SIGGRAPH)*, pp. 339-346.

Jung, J., Jwa, Y., Sohn, G., 2017. Implicit Regularization for Reconstructing 3D Building Rooftop Models Using Airborne LiDAR Data. In: *Sensors*, 17(3), pp. 621:1-621:27.

Jung, J., Sohn, G., Bang, K., Wichmann, A., Armenakis, C., Kada, M., 2016. Matching Aerial Images to 3D Building Models Using Context-Based Geometric Hashing. In: *Sensors*, 16(6), 932.

Kada, M., 2007. Scale-Dependent Simplification of 3D Building Models Based on Cell Decomposition and Primitive Instancing. In: Winter, S., Duckham, M., Kulik, L., Kuipers, B. (eds.), *Spatial Information Theory*, Springer-Verlag, Berlin Heidelberg, Germany, pp. 222-237.

Kada, M., 2009. The 3D Berlin Project. In: Fritsch, D. (ed.), *Photogrammetric Week '09*, Wichmann Verlag, Heidelberg, pp. 331-340.

Kada, M., 2014. Progressive Transmission of 3D Building Models Based on String Grammars and Planar Half-Spaces. In: *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, II-2, pp. 9-14.

Kada, M., McKinley, L., 2009. 3D Building Reconstruction from LiDAR Based on a Cell Decomposition Approach. In: *ISPRS Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, XXXVIII-3/W4, pp. 29-43.

Kada, M., Wichmann, A., 2012. Sub-Surface Growing and Boundary Generalization for 3D Building Reconstruction. In: *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, I-3, pp. 233-238.

Kada, M., Wichmann, A., 2013. Feature-Driven 3D Building Modeling Using Planar Halfspaces. In: *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, II-3/W3, pp. 37-42.

Kaufman, L., Rousseeuw, P. J., 2005. Finding Groups in Data: An Introduction to Cluster Analysis. John Wiley & Sons, Inc., Hoboken, NJ, USA.

Kazhdan, M., Funkhouser, T., Rusinkiewicz, S., 2004. Symmetry descriptors and 3D shape matching. In: *Proceedings of the Eurographics/ACM SIGGRAPH Symposium on Geometry Processing*, 2004, pp. 115-123.

Kelly, G., McCabe. H., 2007. Citygen: An Interactive System for Procedural City Generation. In: *Proceedings of the 5th International Conference in Computer Game Design and Technology*, pp. 8-16.

Kelly, T., Wonka, P., 2011. Interactive Architectural Modeling with Procedural Extrusions. In: *ACM Transactions on Graphics*, 30(2), pp. 14:1-14:15.

Kettner, L., 1999. Using Generic Programming for Designing a Data Structure for Polyhedral Surfaces. In: *Computational Geometry*, 13(1), pp. 65-90.

Khoshelham, K., 2007. Extending Generalized Hough Transform to Detect 3D Objects in Laser Range Data. In: *ISPRS Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, XXXVI-3/W52, pp. 206-210.

Kim, K. H., Shan, J., 2011. Building Roof Modeling from Airborne Laser Scanning Data Based on Level Set Approach. In: *ISPRS Journal of Photogrammetry and Remote Sensing*, 66(4), pp. 484-497.

Kolbe, T. H., 2009. Representing and Exchanging 3D City Models with CityGML. In: Lee, J., Zlatanova, S. (eds.), *3D Geo-Information Sciences*, Springer-Verlag, Berlin Heidelberg, Germany, pp. 15-31.

Kolbe, T. H., Gröger, G., Plümer, L., 2005. CityGML – Interoperable Access to 3D City Models. In: van Oosterom, P., Zlatanova, S., Fendel, E. M. (eds.): *Geo-Information for Disaster Management*, Springer-Verlag, Berlin Heidelberg, Germany, pp. 883-899.

Kong, D., Xu, L., Li, X., 2013. A New Method for Building Roof Segmentation from Airborne LiDAR Point Cloud Data. In: *Measurement Science and Technology*, 24(9), pp. 1-13.

Krecklau, L., Born, J., Kobbelt, L., 2013. View-Dependent Realtime Rendering of Procedural Facades with High Geometric Detail. In: *Computer Graphics Forum*, 32(2pt4), pp. 479-488.

Krecklau, L., Kobbelt, L., 2011a. Procedural Modeling of Interconnected Structures. In: *Computer Graphics Forum*, 30(2), pp. 335-344.

Krecklau, L., Kobbelt, L., 2011b. Realtime Compositing of Procedural Facade Textures on the GPU. In: *ISPRS Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, XXXVIII-5/W16, pp. 177-184.

Krecklau, L., Pavic, D., Kobbelt, L., 2010. Generalized Use of Non-Terminal Symbols for Procedural Modeling. In: *Computer Graphics Forum*, 29(2), pp. 1-12.

Kumar, S., Sallam, M., Goldgof, D., 2001. Matching Point Features Under Small Nonrigid Motion. In: *Pattern Recognition*, 34(12), pp. 2353-2365.

Kwak, E., Habib, A., 2014. Automatic Representation and Reconstruction of DBM from LiDAR Data Using Recursive Minimum Bounding Rectangle. In: *ISPRS Journal of Photogrammetry and Remote Sensing*, 93, pp. 171-191.

Lafarge, F., Descombes, X., Zerubia, J., Pierrot-Deseilligny, 2010. Structural Approach for Building Reconstruction from a Single DSM. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32(1), pp. 135-147.

Lafarge, F., Mallet, C., 2012. Creating Large-Scale City Models from 3D-Point Clouds: a Robust Approach with Hybrid Representation. In: *International Journal of Computer Vision*, 99(1), pp. 69-85.

Lari, Z., Habib, A., 2014. An Adaptive Approach for the Segmentation and Extraction of Planar and Linear/Cylindrical Features from Laser Scanning Data. In: *ISPRS Journal of Photogrammetry and Remote Sensing*, 93, pp. 192-212.

Leszek, K., 2015. Environmental and Urban Spatial Analysis Based on a 3D City Model. In: Gervasi, O., Murgante, B., Misra, S., Gavrilova, M. L., Rocha, A. M., Torre, C., Taniar, D., Apduhan, B. O.(eds.), *Computational Science and Its Applications - ICCSA 2015*, Lecture Notes in Computer Science 9157, Springer-Verlag, Switzland, pp. 633-645.

Li, Y., Wu, X., Chrysanthou, Y., Sharf, A., Cohen-Or, D., Mitra, N. J., 2011. GlobFit: Consistently Fitting Primitives by Discovering Global Relations. In: *ACM Transactions on Graphics*, 30(4), pp. 52:1-52:12.

Lindenmayer, A., 1968. Mathematical Models for Cellular Interactions in Development. In: *Journal of Theoretical Biology*, 18(3), pp. 280-315.

Lipp, M., Wonka, P., Wimmer, M., 2008. Interactive Visual Editing of Grammars for Procedural Architecture. In: *ACM Transactions on Graphics*, 27(3), pp. 102:1-102:110.

Lloyd, S., 1957. Least Squares Quantization in PCM. In: *Reprint: IEEE Transactions on Information Theory (1982)*, 28(2), pp. 129-137.

Lowe, D. G., 1999. Object Recognition from Local Scale-Invariant Features. In: *Proceedings of the International Conference on Computer Vision*, pp. 1150-1157.

Löwe, M., 1993. Algebraic Approach to Single-Pushout Graph Transformation. In: *Theoretical Computer Science*, 109(1-2), pp. 181-224.

Maas, H.-G., Vosselman, G., 1999. Two Algorithms for Extracting Building Models from Raw Laser Altimetry Data. In: *ISPRS Journal of Photogrammetry and Remote Sensing*, 54(2-3), pp. 153-163.

Mac Lane, S., 1998. Categories for the Working Mathematician (2nd edition). Springer-Verlag, New York, NY, USA.

MacQueen, J., 1967. Some Methods for Classification and Analysis of Multivariate Observations. In: *Proceedings of the 5th Berkeley Symposium on Mathematical Statistics and Probability*, 1, pp. 281-297.

Maltezos, E., Ioannidis, C., 2016. Automatic Extraction of Building Roof Planes from Airborne LiDAR Data Applying an Extended 3D Randomized Hough Transform. In: *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, III-3, pp. 209-216.

Mäntylä, M., 1988. An Introduction to Solid Modeling. Computer Science Press, Rockville, MD, USA.

Marvie, J.-E., Buron, C., Gautron, P., Hirtzlin, P., Sourimant, G., 2012. GPU Shape Grammars. In: *Computer Graphics Forum*, 31(7), pp. 2087-2095.

Marvie, J.-E., Perret, J., Bouatouch, K., 2005. The FL-System: A Functional L-System for Procedural Geometric Modeling. In: *The Visual Computer*, 21(5), pp. 329-339.

Masuda, T., 2002. Object Shape Modelling from Multiple Range Images by Matching Signed Distance Fields. In: *Proceedings of the First International Symposium on 3D Data Processing Visualization and Transmission*, pp. 439-448.

McQuitty, L. L., 1960. Hierarchical Linkage Analysis for the Isolation of Types. In: *Educational and Psychological Measurement*, 20(1), pp. 55-67.

Meagher, D., 1982. Geometric Modeling Using Octree Encoding. In: *Computer Graphics and Image Processing*, 19(2), pp. 129-147.

Medeiros, E., Velho, L., Lopes, H., 2003. Restricted BPA: Applying Ball-Pivoting on the Plane. In: *Proceedings of the 17th Brazilian Symposium on Computer Graphics and Image Processing*, pp. 372-379.

Mehlhorn, K., 1984. Data Structures and Algorithms 3: Multi-Dimensional Searching and Computational Geometry. Springer-Verlag, Berlin Heidelberg, Germany.

Meidow, J., Hammer, H., 2016. Algebraic Reasoning for the Enhancement of Data-Driven Building Reconstructions. In: *ISPRS Journal of Photogrammetry and Remote Sensing*, pp. 189-190.

Melzer, T., 2007. Non-Parametric Segmentation of ALS Point Clouds Using Mean Shift. In: *Journal of Applied Geodesy*, 1(3), pp. 159-170.

Milde, J., Brenner, C., 2009. Graph-Based Modeling of Building Roofs. In: *Proceedings of the 12th AGILE International Conference on Geographic Information Science*, (on CD-ROM).

Milde, J., Zhang, Y., Brenner, C., Plümer, L., Sester, M., 2008. Building Reconstruction Using a Structural Description Based on a Formal Grammar. In: *ISPRS Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, XXXVII-B3b, pp. 227-232.

Morgan, M., Habib, A., 2002. Interpolation of LiDAR Data and Automatic Building Extraction. In: *Proceedings of the ACSM-ASPRS Annual Conference*.

Mortenson, M. E., 1997. Geometric Modeling (2rd edition). John Wiley & Sons, Inc., New York, NY, USA.

Moser, J., Albrecht, F., Kosar, B., 2010. Beyond Visualisation – 3D GIS Analyses for Virtual City Models. In: *ISPRS Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, XXXVIII-4/W15, pp. 143-146.

Mukhopadhyay, P., Chaudhuri, B. B., 2015. A Survey of Hough Transform. In: *Pattern Recognition*, 48(3), pp. 993-1010.

Müller, P., Vereenooghe, T., Ulmer, A., van Gool, L., 2005. Automatic Reconstruction of Roman Housing Architecture. In: Baltsavias, M., Gruen, A., van Gool, L., Pateraki, M. (eds.), *Recording, Modeling and Visualization of Cutural Heritage*, Taylor & Francis, London, UK, pp. 287-298.

Müller, P., Vereenooghe, T., Wonka, P., Paap, I., van Gool, L., 2006a. Procedural 3D Reconstruction of Puuc Buldings in Xkipché. In: *Proceedings of the 7th International Symposium on Virtual Reality, Archaeology and Cultural Heritage (VAST)*, pp. 139-146.

Müller, P., Wonka, W., Haegler, S., Ulmer, A., van Gool, L., 2006b. Procedural Modeling of Buildings. In: *ACM Transactions on Graphics*, 25(3), pp. 614-623.

Müller, P., Zeng, G., Wonka, P., van Gool, L., 2007. Image-Based procedural modeling of facades. In: *ACM Transactions on Graphics*, 26(3), pp. 85:1-85:9.

Müller Arisona, S., Zhong, C., Huang, X., Qin, R., 2013. Increasing Detail of 3D Models Through Combined Photogrammetric and Procedural Modelling. In: *Geo-spatial Information Science*, 16(1), pp. 45-53.

Nagl, M., 1987. Set Theoretic Approaches to Graph Grammars. In: Ehrig, H., Nagl, M., Rozenberg, G., Rosenfeld, A. (eds.), *Graph-Grammars and Their Application to Computer Science*, Lecture Notes in Computer Science 291, Springer-Verlag, Berlin Heidelberg, Germany, pp. 41-54.

National Research Foundation, 2017. Virtual Singapore. https://www.nrf.gov.sg/programmes/virtual-singapore, last visited: 14.02.2018.

Naylor, B., 1993. Constructing Good Partitioning Trees. In: *Proceedings of Graphics Interface '93*, pp. 181-191.

Naylor, B. F., Thibault, W. C., 1986. Application of BSP Trees to Ray-Tracing and CSG Evaluation. In: *Technical Report GIT-ICS 86/03*, Georgia Institute of Technology, School of Information and Computer Science.

Neidhart, H., Sester, M., 2008. Extraction of Building Ground Plans from LiDAR Data. In: *ISPRS Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, XXXVII-B2, pp. 405-410.

Nguatem, W., Drauschke, M., Mayer, H., 2013. Roof Reconstruction from Point Clouds Using Importance Sampling. In: *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, II-3/W3, pp. 73-78.

Novacheva, A., 2008. Building Roof Reconstruction from LIDAR Data and Aerial Images through Plane Extraction and Colour Edge Detection. In: *ISPRS Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, XXXVII-B6b, pp. 53-58.

Nüchter, A., Lingemann, K., Hertzberg, J., 2007. Cached K-D Tree Search for ICP Algorithms. In: *Proceedings of the Sixth International Conference on 3-D Digital Imaging and Modeling*, pp. 419-426.

Nurunnabi, A., Belton, D., West, G., 2014. Robust Statistical Approaches for Local Planar Surface Fitting in 3D Laser Scanning Data. In: *ISPRS Journal of Photogrammetry and Remote Sensing*, 96, pp. 106-122.

Oda, K., Takano, T., Doihara, T., Shibasaki, R., 2004. Automatic Building Extraction and 3-D City Modeling from Lidar Data Based on Hough Transformation. In: *ISPRS Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, XXXV-B3, pp. 277-280.

Osher, S., Sethian, J. A., 1988. Fronts Propagating with Curvature Dependent Speed: Algorithms Based on Hamilton-Jacobi Formulations. In: *Journal of Computational Physics*, 79(1), pp. 12-49.

Oude Elberink, S., 2008. Problems in Automated Building Reconstruction Based on Dense Airborne Laser Scanning Data. In: *ISPRS Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, XXXVII-B3a, pp. 93-98.

Oude Elberink, S., Vosselman, G., 2009. Building Reconstruction by Target Based Graph Matching on Incomplete Laser Data: Analysis and Limitations. In: *Sensors*, 9(8), pp. 6101-6118.

Overby, J., Bodum, L., Kjems, E., Ilsøe, P. M., 2004. Automatic 3D Building Reconstruction from Airborne Laser Scanning and Cadastral Data Using Hough Transform. In: *ISPRS Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, XXXV-B3, pp. 296-301.

Parish, Y. I. H., Müller, P., 2001. Procedural Modeling of Cities. In: *Proceedings of the 28th International Conference on Computer Graphics and Interactive Techniques (SIGGRAPH),* pp. 301-308.

Park, J., Lee, I., Choi, Y., Lee, Y. J., 2006. Automatic Extraction of Large Complex Buildings Using Lidar Data and Digital Maps. In: *ISPRS Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, XXXVI-3, pp. 148-154.

Paterson, M. S., Yao, F. F., 1990. Efficient Binary Space Partitions for Hidden-Surface Removal and Solid Modeling. In: *Discrete & Computational Geometry*, 5(5), pp. 485-503.

Pearson, K., 1901. On Lines and Planes of Closest Fit to Systems of Points in Space. In: *Philosphical Magazine Series 6*, 2(11), pp. 559-572.

Perera, G. S. N., 2015. Automated Generation of 3D Building Models from Dense Point Clouds & Aerial Photos. PHD thesis, TU Dresden, Sachsen, Germany.

Perera, G. S. N., Maas, H.-G., 2014. Cycle Graph Analysis for 3D Roof Structure Modelling: Concepts and Performance. In: *ISPRS Journal of Photogrammetry and Remote Sensing*, 93, pp. 213-226.

Peyraud, S., Bétaille, D., Renault, S., Ortiz, M., Mougel, F., Meizel, D., Peyret, F., 2013. About Non-Line-of-Sight Satellite Detection and Exclusion in a 3D Map-Aided Localization Algorithm. In: *Sensors*, 13(1), pp. 829-847.

Pfaltz, J. L., Rosenfeld, A., 1969. Web Grammars. In*: Proceedings of the 1st International Conference on Artificial Intelligence*, pp. 609-619.

Philipp, D., Baier, P., Dibak, C., Dürr, F., Rothermel, K., Becker, S., Peter, M., Fritsch, D., 2014. MapGENIE: Grammar-Enhanced Indoor Map Construction from Crowd-Sourced Data. In: *Proceedings of the IEEE International Conference on Pervasive Computing and Communications*, pp. 139-147.

Pohl, M., Meidow, J., Bulatov, D., 2013. Extraction and Refinement of Building Faces in 3D Point Clouds. In: *Proceedings of SPIE, Image and Signal Processing for Remote Sensing XIX*, 8892, pp. 88920V1-88920V10.

Pomerleau, F., Colas, F., Siegwart, R., 2015. A Review of Point Cloud Registration Algorithms for Mobile Robotics. In: *Foundations and Trends in Robotics*, 4(1), pp. 1-104.

Post, E. L., 1936. Finite Combinatory Process – Formulation 1. In: *The Journal of Symbolic Logic*, 1(3), pp. 103-105.

Poullis, C., You, S., 2009. Photorealistic Large-Scale Urban City Model Reconstruction. In: *IEEE Transactions on Visualization and Computer Graphics*, 15(4), pp. 654-669.

Pratt, W. K., 2007. Digital Image Processing: PIKS Scientific Inside (4th edition). John Wiley & Sons, Inc., Hoboken, NJ, USA.

Prusinkiewicz, P., 1986. Graphical Applications of L-Systems. In: Proceedings on Graphics Interface / Vision Interface, pp. 247-253.

Prusinkiewicz, P., Lindenmayer, A., 1990. The Algorithmic Beauty of Plants. Springer-Verlag, New York, NY, USA.

Rabbani, T., van den Heuvel, F., 2005. Efficient Hough Transform for Automatic Detection of Cylinders in Point Clouds. In: *ISPRS Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, XXXVI-3/W19, pp. 60-65.

Radha, H., Vetterli, M., Leonardi, R., 1996. Image Compression Using Binary Space Partitioning Trees. In: *IEEE Transactions on Image Processing*, 5(12), pp. 1610-1624.

Ranjbar, H. R., Gharagozlou, A. R., Nejad, A. R. V., 2012. 3D Analysis and Investigation of Traffic Noise Impact from Hemmat Highway Located in Tehran on Buildings and Surrounding Areas. In: *Journal of Geographic Information System*, 4(4), pp. 322-334.

Rau, J.-Y., Lin, B.-C., 2011. Automatic Roof Model Reconstruction from ALS Data and 2D Ground Plans Based on Side Projection and the TMR Algorithm. In: *ISPRS Journal of Photogrammetry and Remote Sensing*, 66(6), pp. S13-S27.

Reddy, J. N., 2006. An Introduction to the Finite Element Method (3rd edition). McGraw-Hill Education, New York, NY, USA.

Requicha, A. G., 1980. Representations for Rigid Solids: Theory, Methods, and Systems. In: *ACM Computing Surveys*, 12(4), pp. 437-464.

Robinson, D., 2006. Urban Morphology and Indicators of Radiation Availability. In: *Solar Energy*, 80(12), pp. 1643-1648.

Rosen, J., 1975. Symmetry Discovered: Concepts and Applications in Nature and Science. Cambridge University Press, Cambridge, UK.

Rottensteiner, F., 2003. Automatic Generation of High-Quality Building Models from LiDAR Data. In: *IEEE Computer Graphics and Applications*, 23(6), pp. 42-50.

Rottensteiner, F., 2006. Consistent Estimation of Building Parameters Considering Geometric Regularities by Soft Constraints. In: *ISPRS Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, XXXVI-3, pp. 13-18.

Rottensteiner, F., 2017. Kontextbasierte Ansätze in der Bildanalyse. In: Heipke, C. (ed.), *Photogrammetrie und Fernerkundung*, Springer-Verlag, Berlin Heidelberg, Germany, pp. 555-602.

Rottensteiner, F., Briese, C., 2003. Automatic Generation of Building Models from LiDAR Data and the Integration of Aerial Images. In: Maas, H.-G., Vosselman, G., Streilein, A. (eds.), *Proceedings of the ISPRS Working Group III/3 Workshop on 3D Reconstruction from Airborne Laserscanner and InSAR Data*, 34(3/W13), pp. 174-180.

Rottensteiner, F., Sohn, G., Gerke, M., Wegner, J. D., 2013. ISPRS Test Project on Urban Classification and 3D Building Reconstruction. http://www2.isprs.org/tl_files/isprs/wg34/docs/ComplexScenes_revision_v4.pdf, last visited: 14.02.2018.

Rottensteiner, F., Sohn, G., Gerke, M., Wegner, J. D., Breitkopf, U., Jung, J., 2014. Results of the ISPRS Benchmark on Urban Object Detection and 3D Building Reconstruction. In: *ISPRS Journal of Photogrammetry and Remote Sensing*, 93, pp. 256-271.

Rottensteiner, F., Sohn, G., Jung, J., Gerke, M., Baillard, C., Benitez, S., Breitkopf, U., 2012. The ISPRS Benchmark on Urban Object Classification and 3D Building Reconstruction. In: *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, I-3, pp. 293-298.

Rottensteiner, F., Trinder, J., Clode, S., Kubik, K., 2005. Automated Delineation of Roof Planes from LiDAR Data. In: *ISPRS Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, XXXVI-3/W19, pp. 221-226.

Rozenberg, G., 1997. Handbook of Graph Grammars and Computing by Graph Transformation, Volume I: Foundations, World Scientific Publishing Co. Pte. Ltd., Singapore.

Rozenberg, G., Salomaa, A., 1997. Handbook of Formal Languages, Vol. 1: Word, Language, Grammar, Vol. 2: Linear Modeling: Background and Applications, Vol. 3: Beyond Words. Springer-Verlag, Berlin Heidelberg, Germany.

Rusinkiewicz, S., Levoy, M., 2001. Efficient Variants of the ICP Algorithm. In: *Proceedings of the Third International Conference on 3D Digital Imaging and Modeling*, pp. 145-152.

Rutzinger, M., Rottensteiner, F., Pfeifer, N., 2009. A Comparison of Evaluation Techniques for Building Extraction from Airborne Laser Scanning. In: *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 2(1), pp. 11-20.

Salvi, J., Matabosch, C., Fofi, D., Forest, J., 2007. A Review of Recent Range Image Registration Methods with Accuracy Evaluation. In: *Image and Vision Computing*, 25(5), pp. 578-596.

Samet, H., 2006. Foundations of Multidimensional and Metric Data Structures. Morgan Kaufmann Publishers, San Francisco, CA, USA.

Samet, H., Tamminen, M., 1985. Bintrees, CSG Trees, and Time. In: *ACM SIGGRAPH Computer Graphics*, 19(3), pp. 121-130.

Sampath, A., Shan, J., 2007. Building Boundary Tracing and Regularization from Airborne LiDAR Point Clouds. In: *Photogrammetric Engineering & Remote Sensing*, 73(7), pp. 805-812.

Sampath, A., Shan, J., 2010. Segmentation and Reconstruction of Polyhedral Building Roofs from Aerial LiDAR Point Clouds. In: *IEEE Transactions on Geoscience and Remote Sensing*, 48(3), pp. 1554-1567.

Satari, M., 2012. Recognition of Dormers from LiDAR Data Using Support Vector Machine. In: *Proceedings of the IEEE International Conference on Geoscience and Remote Sensing Symposium*, pp. 6005-6008.

Satari, M., Samadzadegan, F., Maas, H.-G., 2012. A Multi-Resolution Hybrid Approach for Building Model Reconstruction from LiDAR Data. In: *The Photogrammetric Record*, 27(139), pp. 330-359.

Sato, M., Sato, Y., Jain, L. C., 1997. Fuzzy Clustering Models and Applications. Physica-Verlag, Heidelberg, Germany.

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

Schneider, H. J., 1970. Chomsky-Systeme für Partielle Ordnungen. In: *Arbeitsberichte des Instituts für Mathematische Maschinen und Datenverarbeitung*, 3(3), Erlangen.

Schumacker, R. A., Brand, B., Gilliland, M. G., Sharp, W. H., 1969. Study for Applying Computer-Generated Images to Visual Simulation. In: *Report AFHRL-TR-69-14*, U.S. Air Force Human Resources Laboratory.

Schürr, A., 1997. Programmed Graph Replacement Systems. In: Rozenberg, G. (ed.), *Handbook of Graph Grammars and Computing by Graph Transformation, Volume 1: Foundations*, World Scientific Publishing Co. Pte. Ltd., Singapore, pp. 479-546.

Schwarz, M., Müller, P., 2015. Advanced Procedural Modeling of Architecture. In: *ACM Transactions on Graphics*, 34(4), pp. 107:1-107:12.

Shahzad, M., Zhu, X. X., 2015. Reconstruction of Building Footprints Using Spaceborne TomoSAR Point Clouds. In: *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, II-3/W5, pp. 385-392.

Shen, W., Zhang, J., Yuan, F., 2011. A New Algorithm of Building Boundary Extraction Based on LiDAR Data. In: *Proceedings of the 19th International Conference on Geomatics*, pp. 1-4.

Shimrat, M., 1962. Algorithm 112: Position of Point Relative to Polygon. In: *Communications of the ACM*, 5(8), p. 434.

Shiode, N., 2001. 3D Urban Models: Recent Developments in the Digital Modelling of Urban Environments in Three-Dimensions. In: *GeoJournal*, 52(3), pp. 263-269.

Shufelt, J. A., 1999. Performance Evaluation and Analysis of Monocular Building Extraction from Aerial Imagery. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 21(4), pp. 311-326.

Siebe, E., Büning, U., 1997. Application of Digital Photogrammetric Products for Cellular Radio Network Planning. In: Fritsch, D., Hobbie, D. (eds.), *Photogrammetric Week '97*, Wichmann Verlag, Heidelberg, pp. 159-164.

Sigut, J., Fumero, F., Nuñez, O., 2015. Over- and Under-Segmentation Evaluation Based on the Segmentation Covering Measure. In: *Proceedings of the 23rd International Conference on Computer Graphics, Visualization and Computer Vision*, pp. 83-89.

Silva, L., Bellon, O. R. P., Boyer, K. L., 2005. Robust Range Image Registration Using Genetic Algorithms and the Surface Interpenetration Measure. In: *Series in Machine Perception and Artificial Intelligence – Vol. 60*, World Scientific Publishing Co., Singapore.

Singh, S. P., Jain, K., Mandla, V. R., 2013. Virtual 3D City Modeling: Techniques and Applications. In: *ISPRS Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, XL-2/W2, pp. 73-91.

Smelik, R., M., Tutenel, T., Bidarra, R., Benes, B., 2014. A Survey on Procedural Modelling for Virtual Worlds. In: *Computer Graphics Forum*, 33(6), pp. 31-50.

Sneath, P. H. A., 1957. The Application of Computers to Taxonomy. In: *Journal of General Microbiology*, 17, pp. 201-226.

Sohn, G., Huang, X., Tao, V., 2008. Using a Binary Space Partitioning Tree for Reconstructing Polyhedral Building Models from Airborne LiDAR Data. In: *Photogrammetric Engineering & Remote Sensing*, 74(11), pp. 1425-1438.

Sohn, G., Jung, J., Jwa, Y., Armenakis, C., 2013. Sequential Modelling of Building Rooftops by Integrating Airborne LiDAR Data and Optical Imagery: Preliminary Results. In: *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, II-3/W1, pp. 27-33.

Sohn, G., Jwa, Y., Jung, J., Kim, H. B., 2012. An Implicit Regularization for 3D Building Rooftop Modeling Using Airborne LiDAR Data. In: *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, I-3, pp. 305-310.

Stadler, A., Kolbe, T. H., 2007. Spatio-Semantic Coherence in the Integration of 3D City Models. In: *Proceedings of the 5th International Symposium on Spatial Data Quality*, pp. 1-8.

Steder, B., Rusu, R. B., Konolige, K., Burgard., W., 2011. Point Feature Extraction on 3D Range Scans Taking Into Account Object Boundaries. In: *Proceedings of the International Conference on Robotics and Automation*, pp. 2601-2608.

Steinberger, M., Kenzel, M., Kainz, B., Müller, J., Wonka, P., Schmalstieg, D., 2014a. Parallel Generation of Architecture on the GPU. In: *Computer Graphics Forum*, 33(2), pp. 73-82.

Steinberger, M., Kenzel, M., Kainz, B., Wonka, P., Schmalstieg, D., 2014b. On-the-fly Generation of Infinite Cities on the GPU. In: *Computer Graphics Forum*, 33(2), pp. 105-114.

Stilla, U., Jurkiewicz, K., 1999. Automatic Reconstruction of Roofs from Maps and Elevation Data. In: *ISPRS Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, XXXII-7-4-3W6, pp. 139-143.

Stiny, G., 1975. Pictorial and Formal Aspects of Shape and Shape Grammars. Birkhauser Verlag, Basel, Switzerland.

Stiny, G., 1980. Introduction to Shape and Shape Grammars. In: *Environment and Planning B*, 7(3), pp. 343-351.

Stiny, G., 1982. Spatial Relations and Grammars. In: *Environment and Planning B*, 9, pp. 113-114.

Stiny, G., Gips, J., 1972. Shape Grammars and the Generative Specification of Painting and Sculpture. In: *Proceedings of Information Processing 71*, pp. 1460-1465.

Stoter, J., de Kluijver, H., Kurakula, V., 2008. 3D Noise Mapping in Urban Areas. In: *International Journal of Geographical Information Science*, 22(8), pp. 907-924.

Stroud, I., Nagy, H., 2011. Solid Modelling and CAD Systems: How to Survive a CAD System. Springer-Verlag, London, England.

Strzalka, A., Alam, N., Duminil, E., Coors, V., Eicker, U., 2012. Large Scale Integration of Photovoltaics in Cities. In: *Applied Energy*, 93, pp. 413-421.

Su, Y.-T., Bethel, J., Hu, S., 2016. Octree-Based Segmentation for Terrestrial LiDAR Point Cloud Data in Industrial Applications. In: *ISPRS Journal of Photogrammetry and Remote Sensing*, 113, pp. 59-74.

Sun, S., Salvaggio, C., 2013. Aerial 3D Building Detection and Modeling from Airborne LiDAR Point Clouds. In: *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 6(3), pp. 1440-1449.

Tam, G. K. L., Cheng, Z.-Q., Lai, Y.-K., Langbein, F. C., Liu, Y., Marshall, D., Martin, R. R., Sun, X.-F., Rosin, P. L., 2013. Registration of 3D Point Clouds and Meshes: A Survey from Rigid to Nonrigid. In: *IEEE Transactions on Visualization and Computer Graphics*, 19(7), pp. 1199-1217.

Tamminen, M., 1984. Comment on Quad- and Octrees. In: *Communications of the ACM*, 27(3), pp. 248-249.

Tarsha-Kurdi, F., Landes, P., Grussenmeyer, P., 2007a. Hough-Transform and Extended RANSAC Algorithms for Automatic Detection of 3D Building Roof Planes from Lidar Data. In: *ISPRS Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, XXXVI-3/W52, pp. 407-412.

Tarsha-Kurdi, F., Landes, P., Grussenmeyer, P., 2008. Extended RANSAC Algorithm for Automatic Detection of Building Roof Planes from LiDAR Data. In: *The Photogrammetric Journal of Finland*, 21(1), pp. 97-109.

Tarsha-Kurdi, F., Landes, T., Grussenmeyer, P., Koehl, M., 2007b. Model-Driven and Data-Driven Approaches Using LiDAR Data: Analysis and Comparison. In: *ISPRS Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, XXXVI-3/W49A, pp. 87-92.

Thrun, S., Wegbreit, B., 2005. Shape from Symmetry. In: *Proceedings of the IEEE International Conference on Computer Vision*, 2, pp. 1824-1831.

Thue, A., 1906. Über unendliche Zeichenreihen. In: *Skrifter udgivne af Videnskabs-Selskabet i Christiania – I. Mathematisk-Naturvidenskabelig Klasse*, 1(7), pp. 1-22.

Thue, A., 1912. Über die gegenseitige Lage gleicher Teile gewisser Zeichenreihen. In: *Skrifter utgit av Videnskapsselskapet i Kristiania – I. Matematistik-Naturvidenskabelig Klasse*, 1(1), pp. 1-67.

Thue, A., 1914. Probleme über Veränderungen von Zeichenreihen nach gegebenen Regeln. In: *Skrifter utgit av Videnskapsselskapet i Kristiania – I. Matematistik-Naturvidenskabelig Klasse*, 1(10), pp. 1-34.

Toja-Silva, F., Lopez-Garcia, O., Peralta, C., Navarro, J., Cruz I., 2016. An Empirical-Heuristic Optimization of the Building-Roof Geometry for Urban Wind Energy Exploitation on High-rise Buildings. In: *Applied Energy*, 164, pp. 769-794.

Toussaint, G. T., 1983. Solving Geometric Problems with the Rotating Calipers. In: *Proceedings of the Mediterranean Electrotechnical Conference*, pp. A10.02/1-4.

Turing, A. M., 1936. On Computable Numbers, with an Application to the Entscheidungsproblem. In: *Proceedings of the London Mathematical Society*, 2(42), pp. 230-265.

Van Gool, L., Martinovic, A., Mathias, M., 2013. Towards Semantic City Models. In: Fritsch, D. (ed.), *Proceedings of the 54th Photogrammetric Week '13*, Wichmann Verlag, Heidelberg, pp. 217-232.

Van Gool, L., Zeng, G., van den Borre, F., Müller, P., 2007. Towards Mass-Produced Building Models. In: *ISPRS Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, XXXVI-3/W49A, pp. 209-220.

Van Kaick, O., Zhang, H., Hamarneh, G., Cohen-Or, D., 2011. A Survey on Shape Correspondence. In: *Computer Graphics Forum*, 30(6), pp. 1681-1707.

Vanegas, C. A., Aliaga, D. G., Beneš, B., 2010. Building Reconstruction Using Manhattan-World Grammars. In: *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 358-365.

Vanegas, C. A., Garcia-Dorado, I., Aliaga, D. G., Benes, B., Waddell, P., 2012. Inverse Design of Urban Procedural Models. In: *ACM Transactions on Graphics*, 31(6), pp. 168:1-168:11.

Vapnik, V., 2000. The Nature of Statistical Learning Theory. Springer-Verlag, New York, NY, USA.

Verma, V., Kumar, R., Hsu, S., 2006. 3D Building Detection and Modeling from Aerial LiDAR Data. In: *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2, pp. 2213-2220.

Vo, A.-V., Truong-Hong, L., Laefer, D. F., Bertolotto, M., 2015. Octree-Based Region Growing for Point Cloud Segmentation. In: *ISPRS Journal of Photogrammetry and Remote Sensing*, 104, pp. 88-100.

Vosselman, G., 1999. Building Reconstruction Using Planar Faces in Very High Density Height Data. In: *ISPRS Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, XXXII-3-2W5, pp. 87-92.

Vosselman, G., Dijkman, S., 2001. 3D Building Model Reconstruction from Point Clouds and Ground Plans. In: *ISPRS Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, XXXIV-3/W4, pp. 37-43.

Vosselman, G., Gorte, B., Sithole, G., Rabbani, T., 2004. Recognising Structure in Laser Scanner Point Clouds. In: *ISPRS Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, XXXVI-8/W2, pp. 33-38.

Vosselman, G., Klein, R., 2010. Visualisation and Structuring of Point Clouds. In: Vosselman, G., Maas, H.-G. (eds.), *Airborne and Terrestrial Laser Scanning*, Whittles Publishing, Dunbeath, Scotland, UK, pp. 45-82.

Vosselman, G., Maas, H.-G., 2010. Airborne and Terrestrial Laser Scanning. Whittles Publishing, Dunbeath, Scotland, UK.

Wagen, J.-F., Rizk, K., 2003. Radiowave Propagation, Building Databases, and GIS: Anything in Common? A Radio Engineer's Viewpoint. In: *Environment and Planning B: Planning and Design*, 30(5), pp. 767-787.

Walker, M. W., Shao, L., Volz, R. A., 1991. Estimating 3-D Location Parameters Using Dual Number Quaternions. In: *CVGIP: Image Understanding*, 54(3), pp. 358-367.

Wang, R., 2013. 3D Building Modeling Using Images and LiDAR: A Review. In: *International Journal of Image and Data Fusion*, 4(4), pp. 273-292.

Ward Jr., J. H., 1963. Hierarchical Grouping to Optimize an Objective Function. In: *Journal of the American Statistical Association*, 58(301), pp. 236-244.

Weber, B., Müller, P., Wonka, P., Gross, M., 2009. Interactive Geometric Simulation of 4D Cities. In: *Computer Graphics Forum*, 28(2), pp. 481-492.

Weiler, K., 1985. Edge-Based Data Structures for Solid Modeling in Curved-Surface Environments. In: *IEEE Computer Graphics and Applications*, 5(1), pp. 21-40.

Wichmann, A., Jung, J., Sohn, G., Kada, M., Ehlers, M., 2015. Integration of Building Knowledge Into Binary Space Partitioning for the Reconstruction of Regularized Building Models. In: *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, II-3/W5, pp. 541-548.

Wichmann, A., Kada, M., 2014. 3D Building Adjustment Using Planar Half-Space Regularities. In: *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, II-3, pp. 189-196.

Wichmann, A., Kada, M., 2016. Joint Simultaneous Reconstruction of Regularized Building Superstructure from Low-Density LiDAR Data Using ICP. In: *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, III-3, pp. 371-378.

Wolf, M., 1999. Photogrammetric Data Capture and Calculation for 3D City Models. In: Fritsch, D., Spiller, R. (eds.), *Photogrammetric Week '99*, Wichmann Verlag, Heidelberg, pp. 305-312.

Wolff, M., Asche, H., 2008. Geospatial Modelling of Urban Security: A Novel Approach with Virtual 3D City Models. In: Gervasi, O., Murgante, B., Laganà, A., Taniar, D., Mun, Y., Gavrilova, M. L. (eds.), *Computational Science and Its Applications - ICCSA 2008*, Springer-Verlag, Berlin Heidelberg, Germany, pp. 42-51.

Wonka, P., Wimmer, M., Sillion, F., Ribarsky, W., 2003. Instant Architecture. In: *Proceedings of the 30th International Conference on Computer Graphics and Interactive Techniques (SIGGRAPH)*, pp. 669-677.

Wu, F., Wang, F., Jiang, P., Zhao, C., Cheng, J., 2015. A Nearest Neighbor Searches (NNS) Algorithm for Fast Registration of 3D Point Clouds Based on GPGPU. In: *Proceedings of the International Conference on Intelligent Systems Research and Mechatronics Engineering*, pp. 2153-2158.

Wu, F., Yan, D.-M., Dong, W., Zhang, X., Wonka, P., 2014. Inverse Procedural Modeling of Facade Layouts. In: *ACM Transactions on Graphics*, 33(4), pp. 121:1-121:10.

Xiao, Y., Wang, C., Li, J., Zhang, W., Xi, X., Wang., C., Dong, P., 2015. Building Segmentation and Modeling from Airborne LiDAR Data. In: *International Journal of Digital Earth*, 8(9), pp. 694-709.

Xiong, B., 2014. Reconstructing and Correcting 3D Building Models Using Roof Topology Graphs. PHD thesis, University of Twente, Overijssel, Netherlands.

Xiong, B., Jancosek, M., Oude Elberink, S., Vosselman, G., 2015. Flexible Building Primitives for 3D Building Modeling. In: *ISPRS Journal of Photogrammetry and Remote Sensing*, 101, pp. 275-290.

Xiong, B., Oude Elberink, S., Vosselman, G., 2014a. Building Modeling from Noisy Photogrammetric Point Clouds. In: *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, II-3, pp. 197-204.

Xiong, B., Oude Elberink, S., Vosselman, G., 2014b. A Graph Edit Dictionary for Correcting Errors in Roof Topology Graphs Reconstructed from Point Clouds. In: *ISPRS Journal of Photogrammetry and Remote Sensing*, 93, pp. 227-242.

Xu, B., Jiang, W. S., Zhu, Q. S., 2015. A Research on the Hierarchy and Completeness of Roof Topology for Robust Building Reconstruction from Airborne Point Cloud. In: *ISPRS Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, XL-4/W5, pp. 147-152.

Xu, R., Wunsch, D. C., 2009. Clustering. John Wiley & Sons, Inc., Hoboken, NJ, USA.

Yaagoubi, R., Yarmani, M., Kamel, A., Khemiri, W., 2015. HybVOR: A Voronoi-Based 3D GIS Approach for Camera Surveillance Network Placement. In: *ISPRS International Journal of Geo-Information*, 4(2), pp. 754-782.

Yan, J., Shan, J., Jiang, W., 2014. A Global Optimization Approach to Roof Segmentation from Airborne LiDAR Point Clouds. In: *ISPRS Journal of Photogrammetry and Remote Sensing*, 94, pp. 183-193.

Yu, B., Liu, H., Wu, J., Lin, W.-M., 2009. Investigating Impacts of Urban Morphology on Spatio-Temporal Variations of Solar Radiation with Airborne LiDAR Data and a Solar Flux Model: A Case Study of Downtown Houston. In: *International Journal of Remote Sensing*, 30(17), pp. 4359-4385.

Zadeh, L., 1965. Fuzzy Sets. In: *Information and Control*, 8(3), pp. 338-353.

Zhang, W., Grussenmeyer, P., Yan, G., Mohamed, M., 2011. Primitive-Based Building Reconstruction by Integration of LiDAR Data and Optical Imagery. In: *ISPRS Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, XXXVIII-5/W12, pp. 5-12.

Zhong, Y., 2009. Intrinsic Shape Signatures: A Shape Descriptor for 3D Object Recognition. In: *Proceedings of the International Conference on Computer Vision Workshops*, pp. 689-696.

Zhou, Q.-Y., Neumann, U., 2008. Fast and Extensible Building Modeling from Airborne LiDAR Data. In: *Proceedings of the 16th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*, pp. 7:1-7:8.

Zhou, Q.-Y., Neumann, U., 2010. 2.5D Dual Contouring: A Robust Approach to Creating Building Models from Aerial LiDAR Point Clouds. In: *Proceedings of the 11th European Conference on Computer Vision*, pp. 115-128.

Zhou, Q.-Y., Neumann, U., 2011. 2.5D Building Modeling with Topology Control. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2489-2496.

Zhou, Q.-Y., Neumann, U., 2012. 2.5D Building Modeling by Discovering Global Regularities. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 326-333.

Zienkiewicz, O. C., Taylor, R. L., Zhu, J. Z., 2013. The Finite Element Method: Its Basis and Fundamentals (7th edition). Butterworth-Heinemann, Oxford, UK.