Abstract

This work presents a novel workflow for data-driven building reconstruction from point clouds acquired with aerial Light Detection and Ranging (LiDAR) sensors. The goal of this thesis is to produce 3D building models of high accuracy and level of detail, including roof superstructures such as dormers. Roof segments shall be connected to each other by intersection edges or step edges, represented by vertical walls. The workflow comprises the extraction of building point clouds from the LiDAR scene, roof segmentation, segment boundary creation, and 3D modeling. The workflow is tested and evaluated for two data sets, using the evaluation method and test data of the "ISPRS Test Project on Urban Classification and 3D Building Reconstruction".

LiDAR points of buildings are extracted from the scene using previously available 2D building boundary polygons. Nearby points from terrain and vegetation are removed using filtering procedures.

For roof segmentation, a robust region growing technique is developed. A unique feature of the segmentation method is the growing of triangles of a Triangulated Irregular Network (TIN) instead of LiDAR points. This minimizes the gaps between segments, because LiDAR points at segment intersections can be assigned multiple segment labels. Additionally, robust adaptive thresholds are introduced as region growing criteria. These enable the region growing procedure to stop at weak edges, while also segmenting non-planar roof segments. Results show that the proposed segmentation outperforms other methods concerning undersegmentation, and that it recognizes even weak edges. Evaluation and an extensive analysis of the input parameters' effects on the results have shown that the segmentation is very robust against LiDAR point cloud characteristics and segment shape.

Segment boundaries are cretated by collapsing the convex hull of segment points. Point density variations in across-track and along-track directions are considered in the collapsing procedure.

For building modeling, the 2.5D dual contouring approach of Zhou and Neumann [2010] is adapted to model complex roofs. After overlying a 2D grid to the segmented point cloud, vertices of the 3D building model are estimated for each grid cell by minimizing a Quadratic Error Function (QEF). Each QEF minimization results in a hyperpoint, which consists of one or more vertices of the building model at the same x-y-coordinates. This 2.5D-characteristic enables the connection of building vertices at step edges with vertical walls. In contrast to Zhou and Neumann [2010], the proposed method

uses a detailed roof segmentation, where segments can be connected by step edges or intersection edges to each other. The main contribution of this work is the modification and weighting of the QEF, such that the number of hyperpoint vertices resulting from QEF minimization adapts to whether step edges or intersection edges shall be modeled.

For enhancing model simplicity, the QEF solutions of all cells are merged by collapsing a quadtree. The quadtree collapsing stops, when the QEF residual exceeds a user-defined threshold. The final hyperpoint vertices are connected according to their adjacency in the quadtree to form 3D faces of the polygonal building model.

Testing of the procedure has shown that the resulting building models are very detailed and watertight. Building superstructures can be represented accurately, and also non-flat roof segments can be modeled in detail. Each roof segment is represented by a triangulation of the building vertices, such that the building models precisely fit the input data, depending on the flexibly chosen input parameters. As the building models are composed of many polygon faces, subsequent regularization is recommended to further enhance model simplicity.

Evaluation has shown that both proposed segmentation and reconstruction methods outperform other methods in important quality measures. The tested scenes show outstanding completeness and undersegmentation, and comparative planimetric accuracy compared to scenes produced with other reconstruction methods. An extensive analysis of the impact of input parameters on the results has shown that the procedure is very robust. The user can influence the level of detail of the builing models by choosing the input parameters. To the best knowledge of the author, the proposed reconstruction method is the first dual contouring approach for modeling complex roof height layers.

Acknowledgements

First and foremost, I wamt to express my deep thanks to my supervisor Janja Avbelj for her mentoring and guidance. She patiently expedited the work of this thesis, always asking the right questions and thereby teaching me a lot about estimation, scientific working and writing, project organization, and the processing of point clouds. I also want to thank her for the motivational support and high expectations, without which this thesis would not have reached the presented scope and quality.

Additionally, I want to express many thanks Dr. Stefan Gernhardt, who supervised the thesis on part of the Technische Universität München (TUM).

Further thanks go to Working Group 4 of Commission III of the International Socienty of Photogrammetry and Remote Sensing (ISPRS), who - via the "ISPRS Test Project on Urban Classification and 3D Building Reconstruction" - provided a data set for testing, evaluating, and comparing the presented 3D reconstruction procedure to other approaches.

Contents

Abbreviations

ALS	**A**irborne **L**aser **S**canning
amsl	**a**bove **m**ean **s**ea **l**evel
DC	**D**ual **C**ontouring
DEM	**D**igital **E**levation **M**odel
DTM	**D**igital **T**errain **M**odel
GIS	**G**eographic **I**nformation **S**ystem
LiDAR	**Li**ght **D**etection **A**nd **R**anging
LUF	**L**ocal **U**nevenness **F**actor **l**evel
MRF	**M**arkov **R**andom **F**ield
NDVI	**N**ormalized **D**ifference **V**egetation **I**ndex
QEF	**Q**uadratic **E**rror **F**unction
RANSAC	**RAN**dom **S**ample **C**onsensus
RG	**R**egion **G**rowing
RTINS	**Robust**dom **TIN**-based Region Growing **S**egmentation
SEP	**S**tep **E**dge **P**robability
TIN	**T**riangulated **I**rregular **N**etwork

To Patrick. You are my rock!

1 Introduction

1.1 Motivation

For more than two decades, 3D building reconstruction has been an active research topic of remote sensing, photogrammetry, and computer vision [Rottensteiner et al., 2012a, Wang, 2013, Haala and Kada, 2010, Lafarge and Mallet, 2012]. Continuing research is driven by the increasing demand for accurate, automatically produced, and detailed 3D city models [Wang, 2013], originating from the increasing number of applications requiring frequent updates.

3D city models were initially used for city planning and in private sector applications, whereas today's utilization of virtual city environments expands to everyday user-driven mobile applications, such as location based services [Wang, 2013, Brenner, 2005], 3D Geographic Information Systems (GIS) for navigation, driver assistance systems, virtual tourism [Zhou and Neumann, 2010], and augmented reality. City models are used for urban planning [Verma et al., 2006], change detection [Rau and Lin, 2011], in commercial and public sector simulations for environmental research[1] [Brenner, 2001, Geibel and Stilla, 2000, Rau and Lin, 2011], telecommunication[2] or solar potential analysis [Brenner, 2001]. Also security issues such as damage assessment in disaster response or military and surveillance mission planning are of relevance [Geibel and Stilla, 2000]. Very detailed building models are requested by video and computer gaming applications [Wang, 2013].

User-driven mobile applications require spatially and temporally accurate information [Brenner, 2005]. Since 3D spatial data is experiencing a higher change rate as 2D maps [Brenner, 2001], the effort for keeping city models up to date depends on the level of automation in 3D building reconstruction. Until a few years ago, large-scale city models were entirely measured manually [Brenner, 2005], since building reconstruction algorithms could not meet the requirement of producing sufficiently detailed building models in an automated manner. As the requirements for the level of detail are increasing with the quality of the input data [Haala and Kada, 2010], the lack of automation in producing detailed building models is still the main reason for the topic of 3D building modeling being under research [Rottensteiner, 2003, Awrangjeb and Fraser, 2014].

High level of detail city models can be computed from pointwise 3D data of high accuracy, density and regularity, such as LiDAR (Light Detection And Ranging [Wehr and Lohr, 1999]) point clouds. In comparison to deriving 3D matching points from optical imagery, LiDAR has the advantage that 3D information is directly measured [Meng et al., 2010, Haala and Kada, 2010]. With continuously improving sensor capabilities and point densities, LiDAR data is now available in a sufficient quality to outperform optical data in regard to 3D point density and regularity [Brenner, 2001, Oude Elberink and Vosselman, 2009]. Even though optical images have a higher resolution and contain textural information for postprocessing, they require stereo point matching and solving for 3D geometry to extract 3D data. This implies additional computational effort, potential error sources [Maas and Vosselman, 1999], and sparse 3D information [Wang, 2013, Brenner, 2001] due to e.g. occlusions, shadows and poor contrast [Awrangjeb and Fraser, 2014]. As a result of the difficulties in using optical imagery,

[1] Simulations for microclimate, air pollution and noise propagation [Rau and Lin, 2011]
[2] Electromagnetic waves propagation for optimal antenna placement [Brenner, 2001]

building reconstruction research has set more focus on LiDAR pointclouds in the last years [Geibel and Stilla, 2000, Haala and Kada, 2010]. There have been several attempts to integrate optical and LiDAR data for building reconstruction [see Wang, 2013, Zhang et al., 2012]; however, acquisitions from within a short time span and a from similar acquisition angle are needed for accurate coregistration [Awrangjeb and Fraser, 2014]. For increasing the reconstruction procedures' independence from compatible data sets and auxiliary data, the research focus was set on buiding reconstruction from only LiDAR point clouds in the last years.

Given a 3D point cloud, building reconstruction requires a sequence of severals tasks [Dorninger and Pfeifer, 2008], for each of which a broad range of methods is available from literature (Chapter 3). Choosing the optimal approach for each step is a major challenge of 3D building reconstruction. Each step has the requirement to perform robustly, maximizing the independence of data characteristics, roof complexity, and assumptions about the building. The goal is to create hole-free polyhedral building models consisting of flat polygon faces. Each roof segment shall be represented by one or more polygon faces, depending on a trade-off between the model's fit to the input data and its simplicity.

Evaluation and comparability of existing approaches has been a challenge with reseachers using different test data and evaluation techniques. Ground truth is in general not available for LiDAR city scenes and can only be created semi-automatically, relying on individual decisions about delineations of roof segments. Therefore, benchmark projects such as the recent *ISPRS Test Project on Urban Classification and 3D Building Reconstruction* (ISPRS benchmark project) [Rottensteiner et al., 2012a] are important events for improving the transparency in the research field. Submitted building reconstruction results were evaluated for different areas using object-based evaluation techniques (Section 3.4), with most methods reaching a completeness[3] between 60 and 85% and a correctness[4] between 80 and 100%.

Common problems of the evaluated approaches are undersegmentation, the detection of small roof segments and producing hole-free building models [Rottensteiner et al., 2012b]. These problems can originate in sparse and irregular LiDAR data, which is a challenge for various tasks (deciding on delimiting parameters for segmentation, creating hypotheses on the building shape, connecting the correct segments during building modeling,...) [Oude Elberink, 2008]. Varying evaluation results of the ISPRS benchmark project for different test areas and segment sizes have shown that many approaches are not robust, as they perform only reliably if their hypotheses on the building form are fulfilled. An optimal approach would therefore be completely independent from the roof complexity and from assumptions on the underlying building roof structure.

[3] Cm, object-based evaluation measure, computed as the ratio of detected reference segments (true positives defined as 50% overlap to an estimated segment) and all reference segments

[4] Cr, object-based evaluation measure, computed as ratio of correct estimated segments (true positives defined as 50% overlap to a reference segment) and all estimated planes

1.2 Definition of Scope and Objectives

This work aims at designing a robust building reconstruction workflow which can flexibly model also complex roof shapes, while reducing the necessary assumptions about the building shape to a minimum. The goals of this thesis are stated as follows:

Literature research on related work: An extensive literature research on building reconstruction from LiDAR data shall precede the development of a new workflow. Different existing methods for solving the sequential tasks of building reconstruction shall be introduced.

Procedure and Output: An own building reconstruction procedure shall be developed and implemented, with the following requirements:

- Scope of the procedure: The procesure shall comprise methods for presegmentation, segmentation, segment boundary generation and model creation. Regularization (improving building simplicity, orthogonalty and parallelity) is not part of the thesis. Focus of the work shall be robust segmentation and modeling, i.e. 3D building polygon creation, once a segmentation is available.
- Accuracy: The segmentation procedure shall segment all relevant roof segments as accurately as possible, maximizing the independence of data characteristics and roof segment shape. The segmentation procedure shall robustly segment also roof segments which are connected to each other in obtuse angles.
- Robustness: Segmentation and modeling procedures shall both be able to create results robustly, maximizing the independence from the input data characteristics and from the ground truth building complexity.
- Output model: The modeling procedure shall create hole-free polyhedral building models, where step edges between segments and to ground are represented by vertical walls. The final model shall preserve a high fit to the original data, while modeling all relevant building parts independently from roof complexity. The procedure shall be able to produce complex roof models including superstructures, which are connected to the basic roof structure by intersection edges or step edges.
- Flexibility: The modeling procedure shall be designed to allow the user to influence the level of detail of the output model by means of input parameters.
- Computational efficiency: Implementation efficiency is of secondary importance. The approach is required to maximize the accuracy of results and the level of detail of building models.

Test and evaluation: The proposed workflow shall be tested using two data sets of different LiDAR point density and regularity. Both data sets shall be evaluated against a ground truth by applying a 2D-referencing procedure. The evaluation method used in the ISPRS benchmark project shall be applied, with the purpose of comparing the performance of the proposed workflow to existing methods. The influence of important parameters for segmentation and reconstruction on the results shall be analysed in detail.

1.3 Overview of the Contents

Chapter 2 presents briefly the necessary background information on the LiDAR system, on the definition of 3D building models, and on assumptions which are made about the building shape during for the reconstruction procedure.

Different existing approaches to building reconstuction from airborne LiDAR data are presented in chapter 3 (State of the Art). The focus is set on solutions for the sequential tasks of data-driven building reconstruction (presegmentation, segmentation, border creation, modeling, and regularization). Also model-driven methods and existing evaluation techniques for building reconstruction are presented.

Chapter 4 describes the development and implementation of a new workflow for building reconstruction. Proposed solutions for segmentation, segment boundary creation, and modeling are presented in detail.

The designed workflow is tested and evaluated for two data sets in chapter 5.

In chapter 6, the proposed workflow and the evaluation results are discussed, and potential improvements for the proposed segmentation and reconstruction methods are derived.

The proposed workflow, its performance and main characteristics are summarized in chapter 7, and the thesis is concluded by a brief statement on the research field of 3D building reconstruction from airborne LiDAR point clouds.

2 Background Theory

2.1 Airborne Laser Scanning

Airborne laser scanning (ALS) systems use an active sensor technology mounted on planes or helicopters at typical flight heights between 200 - 300 m or 500 - 1000 m respectively [Baltsavias, 1999b]. The advantage of ALS compared to radar technology is the significantly smaller wavelength (typical wavelengths are 800 - 1500 nm), allowing a high sampling frequency and a more accurate range determination [Wehr and Lohr, 1999]. Range measurements are achieved either by comparing the phase difference of the backscattered light to the emitted continuous beam (Continuous Wave LiDAR) or, which is more popular among current LiDAR systems, by measuring the time elapsed between a sent light pulse and the received signal [Wehr and Lohr, 1999]. Pulsed systems are able to catch different pulse returns such as first and last pulse. Different return times indicate height discrepancies within the same lidar beam's ground footprint and can be used e.g. for identification of trees and building corners.

LiDAR data is acquired in overlapping "strips" along flight direction. The main data characteristics of pulse LiDAR are swath width, laser footprint diameter and the laser point density. Current sensors can cover large areas with a point density from one [Rottensteiner, 2003] to ten points per square meter [Oude Elberink and Vosselman, 2009]. Smaller regions are available in up to 25 points per square meter [Oude Elberink, 2008], since the point density depends, amongst others, on flight height: The smaller the region, the higher the point density, when using the same sensor. The regularity of point density varies according to the scan pattern [Baltsavias, 1999b], which depends on the used LiDAR system. Possible patterns are parallel lines, meanderwise bidirectional parallel lines, bidirectional Z-shaped lines, and sinusoidal or elliptical patterns [Baltsavias, 1999a]. In line scanners (parallel lines), the strongest variations in point spacing is the difference of *along-track point spacing* - resulting from the flight speed over ground - and *across-track point spacing*, which depends on the swath width and the number of points in one scan line. The along-track point spacing is usually smaller than across-track [Baltsavias, 1999b]. Range accuracy depends on the relative positioning of the sensor to the reflecting object at sending and receiving time. The relative position is determined using GNSS and inertial navigation systems, and depends on the steepness of the received pulse[1]. Therefore, range accuracy varies even within one data acquisition strip (typically ~10 - 20 cm; horizontal accuracy: ~5 - 40 cm)[AG, 2014].

Although LiDAR provides a dense point cloud compared to stereo matching from optical data, gaps and irregularities in the data can occur due to light-absorbing, transparent or mirroring surfaces. Also occlusions due to height discrepancies at a non-vertical acquisition angle cause data gaps [Oude Elberink and Vosselman, 2009].

[1]The higher the acquisition angle, the longer the rise time of the pulse, the lower the range accuracy [Baltsavias, 1999b]: A long time span between transmission and reception of the light beam leads to less accuracy in the relative positioning of the sensor to the object.

2.2 3D Building Model - Definitions and Assumptions

3D building models are defined as polygonal mesh models which represent the building with a significantly reduced number of data points than the original point cloud [Wang, 2013]. Most authors define 3D building models in a 2.5-D sense, assuming buildings to be composed of planar shapes[2] [Dorninger and Pfeifer, 2008, Sampath and Shan, 2010, Perera et al., 2012, Sohn et al., 2012, Kada and Wichmann, 2012, Verma et al., 2006]. Roof segments are connected by intersection edges or vertical step edges to each other or to ground; overhangs and small building parts such as small dormers and chimneys are neglected [Dorninger and Pfeifer, 2008]. This corresponds to the definition of Level of Detail (LoD) 2 [3] of the official OGC standard *City Geography Markup Language* (CityGML), an information model intending a standardized "representation, storage, and exchange of virtual 3D city and landscape models" [Kolbe, 2012, Kolbe et al., 2005]. A simple triangulation of the available LiDAR points is not considered as a building model, as it constitutes no simplification, and because different height layers are not respresented by step edges.

Additionally to the planarity of roof segments, further assumptions on the building form are about:

- Allowing intersection edges: Poullis and You [2009] developed a simple city reconstruction workflow assuming all roof segments to be connected by step edges to each other. This assumption is true for certain urban regions, where building parts can mostly be approximated by cuboid forms, but does not apply e.g. in a European town, where intersection edges, and complex roof shapes dominate the scene; simple cuboid forms are not sufficient to model gable roofs and dormers.

- Steepness of roof segments: Most methods restrict the steepness of roof segments to a certain angle, removing LiDAR points where the local plane is steeper than a threshold already before segmentation. This assumption leads to problems when modeling buildings with steep mansards.

- Orientation of roof segments' plane normal: Some methods assume the roof segment planes to be oriented perpendicular or parallel to the main building direction, or perpendicular to the longest line segment of the building's polygonal boundary it is connected to [Haala et al., 1998].

- Main building direction: The main building direction is assumed to be constant within the city scene [You et al., 2003], and is often defined to be aligned with the longest longest line segment of the building's polygonal boundary.

- Directions of boundary line segments: The building's outer boundary is often assumed to be a polygon consisting only of perpendicular and parallel edges [Maas and Vosselman, 1999, Rau and Lin, 2011, Matei et al., 2008]. Some authors additionally allow multiples of 45Âř to the main building direction.

[2] The planarity of roof segments is especially useful because representing arbitrary (non-planar) shapes often requires computationally intensive surface fitting procedures [Wang, 2013].

[3] LoD2 requires a positional accuracy of 2 m and a height accuracy of 1 m [Kolbe et al., 2005], which is especially important for 3D GIS, where textured facades are mapped onto the models [Kada and McKinley, 2009]

3 State of the Art

The first task in building reconstruction is to extract building point couds from the LiDAR scene (presegmentation). While most authors directly work on LiDAR data, some authors use rasterized and interpolated Digital Elevation Models (DEM) for building reconstruction. After presegmentation, building reconstruction approaches can be classified into two major categories [e.g. Awrangjeb and Fraser, 2014] (Fig. 3.1)[1]:

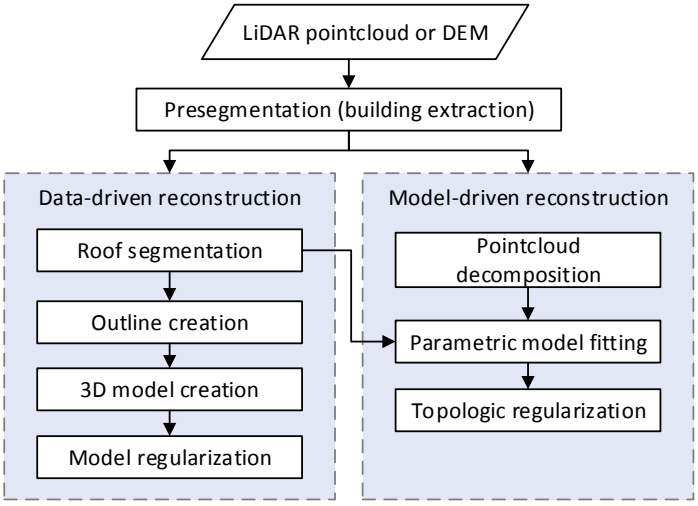


Figure 3.1: Classification of building reconstruction approaches

Data-driven (bottom-up) methods (Section 3.2) connect individual roof segments, which are constructed according to a preliminary segmentation of the building point cloud. This is usually followed by creating polygonal segment boundaries. Polyhedral 3D models are constructed using heuristic approaches or structural modeling procedures. Subsequent regularization procedures ensure simplicity, parallelity, and orthogonality of the 3D model. Even though data-driven approaches require a high regularization effort, they are very popular in recent research [see Rottensteiner et al., 2012a, Wang, 2013], because they are more flexible in modeling complex roof shapes than model-driven approaches. Building models from data-driven methods fit most accurately the input data and are therefore suited for applications where the focus is put on accuracy and knowledge about small roof parts, such as for simulations or augmented reality.

Model-driven (top-down) methods (Section 3.3) select best fitting parametric models from a prepared catalogue and estimated the corresponding model parameters according to the best fit to the LiDAR data. Many approaches decompose the building point cloud into different parts and fit a model to each part [Haala and Kada, 2010, Tarsha-Kurdi et al., 2007]. Those models are then connected and corrected to achieve topologic plausibility. Model-driven reconstruction is limited to the catalogue models and can therefore not flexibly represent any roof shape. The procedure is however robust,

[1]As a matter of course, also hybrids of distinguished approaches exist. In this thesis, hybrid approaches are not explicitly stated in the overview flowcharts.

effective and fast, due to a minimal regularization effort. Constraints such as parallelity and orthogonality of polygon edges are already provided with the parametric model. The building models are considered to be simple and aesthetic, and are therefore suited in applications where the building's accuracy is of less importance, such as 3D automobile navigation.

While data-driven heuristic methods were dominating in the past and are still actively researched [Rottensteiner, 2003, Poullis and You, 2009, Dorninger and Pfeifer, 2008, Sampath and Shan, 2010], recent methods - especially those emerging from computer vision community [Wang, 2013] - tend to model-driven reconstruction [Kada and McKinley, 2009, Verma et al., 2006, You et al., 2003] and structural data-driven methods [Zhou and Neumann, 2010, Lafarge and Mallet, 2012, Sohn et al., 2008, Fiocco et al., 2005].

3.1 Presegmentation

Presegmentation typically classifies the LiDAR point cloud into buildings, terrain, and vegetation (including other non-terrain objects and clutter). Presegmentation can be performed in one step or sequentially, by first separating elevated points from those on ground, and then removing vegetation from the remaining data, or vice versa. Popular ground filtering method is to set a height threshold on a Digital Terrain Model (DTM), which can be produced e.g. by morphological filter operations [Morgan and Tempfli, 2000, Zhang et al., 2006, Ameri and Fritsch, 2000]. Other approaches are to identify planar LiDAR points[2] and to create connected components of the latter, assuming the largest connected component to be ground [Verma et al., 2006]. Connected components can also be used for vegetation filtering, assuming connected components of small size [Verma et al., 2006] or of low planarity [Sampath and Shan, 2010] to be vegetation. The sequential process can be inverted, e.g. Sun and Salvaggio [2013] classify first vegetation with a graph-cuts method, and then use Euclidean clustering to identify buildings. A one-step scene classification can be achieved e.g. by graph-cut optimization. Lafarge and Mallet [2012] define expectation values for buildings, vegetation, ground and clutter by combining different covariance-based measures and height information in an energy optimization term. Dorninger and Pfeifer [2008] extract all planar regions of the scene using a region growing segmentation[3] in feature space and group the extracted points to buildings with a mean-shift algorithm. Alternatively, building point clouds can be directly extracted from 2D building footprints, which are available beforehand[Rau and Lin, 2011], or which are provided interactively by user inputs [You et al., 2003].

3.2 Data-driven Reconstruction

3.2.1 Roof Segmentation

Roof segmentation is the task of classifying LiDAR points into different roof segments. With the purpose of creating hole-free building models, each segment is defined as a group of LiDAR points

[2]The planarity of points can be estimated e.g. by covariance analysis, see section 3.2.1
[3]see Section 3.2.1

on a plane or another continuous shape, which is connected to other segments or to ground by intersection edges or step edges. Existing segmentation methods are classified into *surface fitting* methods and *region-based* methods. Surface fitting methods fit surface models, mostly planes, directly to the point cloud. Region-based methods group points according to their proximity in Euclidean or feature space. Region-based methods are most popular among segmentation methods, as they classify the points in the context of their neighborhood and can therefore delimit the segments accurately without overlap or interleave[4] [Oude Elberink and Vosselman, 2009]. Surface-fitting methods determine the segment's form already during the segmentation step, whereas region-based segmentation might require subsequent surface fitting processes, depending on the requirements of the following modeling procedure (Section 3.2.3). .

Surface-fitting techniques

Surfaces can be fitted to a point cloud by maximizing the number of inliers, i.e. points whose orthogonal distances to the estimated surface are below a threshold. RANSAC and Hough Transform are examples for such surface fitting procedures:

RANSAC (RANdom SAmple Consensus) is an iterative model fitting procedure where in each iteration, a model is created by from a randomly selecting a necessary number of samples from the data set (i.e. for a random plane, selecting three random points). The quality of the random model is typically determined by counting the number inlier points using a distance threshold. Any other measure of quality (e.g. RMSE of all points, standard deviation of inliers,...) can be applied. In case the quality measure is better than in the previous iteration, the model is kept as the currently best estimate. The necessary number of iterations is estimated as

$$N = \frac{\log(1-p)}{log(1-(1-\varepsilon)^s)}, \tag{3.1}$$

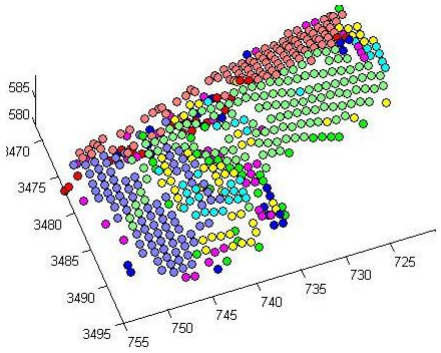
where s is the number of necessary random samples, ε is the percentage of expected outliers and p is the required probability of selecting at least one outlier-free random set of points in N iterations [Tarsha-Kurdi et al., 2008]. Sohn et al. [2008], Tarsha-Kurdi et al. [2008], Ameri and Fritsch [2000], and Brenner [2000] use RANSAC for extraction of planar segments.

Hough Transform is an image processing method which is often applied for line detection. The 2D Hough Transform transforms Euclidean space coordinates of points to a discrete 2D parameter space representing lines passing through the points. The accumulation of parameters in a 2D histogram results in an indication for lines, with the histogram maxima defining those lines' parameters. Rau and Lin [2011] use the 2D Hough Transform to detect planes indirectly after projecting the 3D point cloud onto planes parallel and orthogonal to the principal building direction. [Vosselman and Dijkman, 2001, Sohn et al., 2008, 2012, Vosselman et al., 2004] use a 3D Hough transform for iteratively detecting planes in the building point cloud.

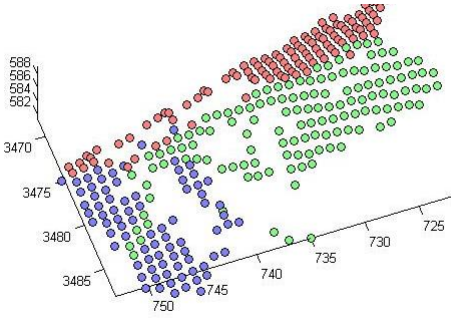
RANSAC outperforms the 3D Hough Transform in both accuracy and computation speed [Tarsha-Kurdi et al., 2007]. Even though surface fitting techniques are computationally very efficient, both

[4]See the shortcomings of surface-fitting segmentation illustrated in Fig. 3.2

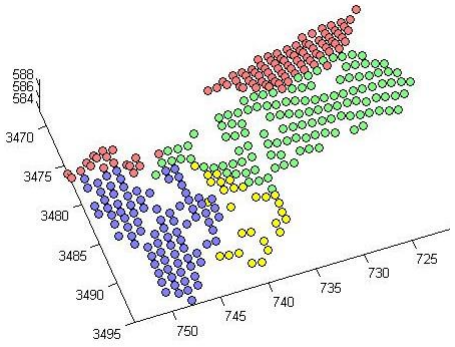
methods have the disadvantage of approaching the segmentation problem in a global way, i.e. points are segmented independently of their proximity or similarity to each other. This leads to misclassification in case of complex roof structures and non-symmetric shapes (Fig. 3.2), such that extensive additional plane verification and outlier detection procedures are necessary.



(a) Segmentation errors from RANSAC plane fitting: Erroneous planes are detected from non-connected region fragments



(b) Segmentation errors from RANSAC plane fitting and additional plane verification: Erroneous planes are interleaving with each other



(c) Segmentation errors from RANSAC plane fitting, plane verification and additional outlier removal: Points are falsely removed, if their local planes don't fit the RANSAC plane

FIGURE 3.2: Errors resulting from RANSAC segmentation, iteratively applied to the point cloud from which all inliers to previous planes are removed. Plane verification (checking for elevation and point inlier density of the estimated plane) and outlier removal (checking for similarity of the points' local normals with the estimated plane) are applied in b) and c) after each plane estimation. Different colored points belong to different planes. Plots are shown in a local coordinate system [m].

Region-based methods

Region-based segmentation methods can be classified into edge-based techniques and region-growing techniques [Sampath and Shan, 2010].

Edge-based methods detect edges in rasterized data (DEM) applying image processing techniques. The detected edges are connected to closed polyons, which delimit the roof segments [Sampath and Shan, 2010]. Edge-based methods were mostly researched in the early years of LiDAR building reconstruction, when LiDAR points were not available in sufficient point density to apply region-growing; an overview of edge-based segmentation is given by Hoover et al. [1996].

Region-growing is the most used segmentation method in literature [Rottensteiner, 2003, Oude Elberink and Vosselman, 2009, Perera et al., 2012, Verma et al., 2006, Nurunnabi et al., 2012, Dorninger and Pfeifer, 2008]: Starting from a seed, each unclassified neighbor to the region's points is added to the region, if predefined criteria are fulfilled.

The local neighborhood of a point can be determined according to different distance measures in Euclidean or feature space[5]:

- Distance-fixed threshold: All points within a certain radius around the inspected point are determined as neighbors. Also a rectangular window can be applied [Verma et al., 2006].
- Grid-based window neighborhood: The neighborhood of all points inside a grid cell is defined as a window of grid cells around this cell; all points in one grid cell have the same neighborhood [Alharthy and Bethel, 2004]
- k-Neighborhood: The k (fixed number) nearest points to the inspected point are determined as neighbors.
- Delaunay neighborhood, also called Voronoi neighborhood: The Delaunay triangulation of a point set connects the points such that the circumcircle around each triangle contains only the three triangle points, resulting in a natural topology even at irregular point distributions [Awrangjeb and Fraser, 2014, Rau and Lin, 2011]. Points connected by triangulation edges are neighbors.

Region growing criteria are thresholds on similarity measures for deciding whether a candidate point is added to a region or not. The most popular criterion is the similarity of the canidate's local plane parameters to the average region's plane parameters. A points local plane, or local normal, can be estimated by fitting a plane to its neighborhood by least squares methods. Another possibility for estimating local normals is covariance analysis of the local neighborhood, using the eigenvector of the smallest eigenvalue as an estimate for local plane parameters. [6] [Sampath and Shan, 2010, Verma et al., 2006, Sun and Salvaggio, 2013]. Different region growing criteria based on local planes are:

- Angular difference of local normals of neighboring points [Ameri and Fritsch, 2000, Lafarge and Mallet, 2012]
- Angular difference of the candidate point's local normal to the average normal of the region [Kada and Wichmann, 2012]
- Distance between a point and the region's plane [Oude Elberink and Vosselman, 2009, Kada and Wichmann, 2012, Nurunnabi et al., 2012]
- Similarity of local plane parameters of neighboring points [Verma et al., 2006]

Kada and Wichmann [2012] iterate the segmentation procedure while increasing the threshold on the angular difference, until each roof point is assigned to at least one segment. Dorninger and Pfeifer [2008] cluster normal parameters in 4D plane feature space, followed by a mean shift segmentation

[5] The neighborhood of a point can be determined according to similarity to other points in terms of predefined characteristics (features).

[6] If all points in the local neighborhood are on a plane, there are only two nonzero eigenvalues in the eigenvalue decomposition . Due to noise and irregularities, the local neighborhood is not perfeclty planar, such that the eigenvector of the smallest eigenvalue can be used as an approximation for the plane normal.

in Euclidean space. Haala et al. [1998] assume that a roof slope is always perpendicular to the building boundary edge, where the segment is adjacent to. They segment points according to their local normals' similarity to the orthogonal directions to the building boundary edges. Sohn et al. [2008] segment the point cloud not into roof segments, but into height layers, using the height discrepancies between point neighbors for defining an adaptive region growing threshold.

Adaptive thresholds are determined according to the local conditions and have the advantage to segment flexibly also non-planar continuous shapes. Nurunnabi et al. [2012] propose an adaptive threshold based on the mean of the reference measures plus a multiple of the standard devation:

$$T = mean(...) + a \cdot std(...), \tag{3.2}$$

where a ranges between 1 and 3. Poullis and You [2009] apply this principle for creating adaptive thresholds for the points' height values and for the local neighborhoods' eigenvalues. Nurunnabi et al. [2012] create robust adaptive thresholds by using the median instead of the mean, and the Median Absolute Deviation (MAD) instead of the standard deviation, which makes the process more robust against outliers.

Region growing seeds can be determined e.g. randomly from the point cloud, or by choosing 2D-spatial extreme points [Poullis and You, 2009]. For efficient processing and stable segmentation, it is recommended to choose a seed which is located most centric within a roof segment [Alharthy and Bethel, 2004]. This has the advantage of allowing the algorithm to proceed fastly in all directions, and to start with the most probable segment characteristics as initial input to the region growing criteria[7]. Such a segment center is estimated by choosing the points whose characteristics differ least from the expected plane characteristics, e.g. the point whose local plane is most similar to the dominant plane parameters. The latter can be obtained by histogram analysis [Dorninger and Pfeifer, 2008] or in Hough space [Oude Elberink and Vosselman, 2009]. Awrangjeb and Fraser [2014] select the midpoints of the building boundary's edges as seed. Others estimate flatness using a covariance-based local curvature measure[8], or according to the local neighborhood's RMSE from the local plane [Alharthy and Bethel, 2004].

3.2.2 Boundary Creation

Many reconstruction procedures require the building to be delimited by a closed boundary (building footprint) [Maas and Vosselman, 1999, Dorninger and Pfeifer, 2008]. Such a boundary may be required also for individual segments, in the form of a closed polygonal boundary polygon, or only by identifying unconnected boundary points[9].

Boundary points and boundary tracing

Connected boundary line segments can be determined from a triangulation of the segment points. Edges which belong to only one triangle are defined as boundary edges [Maas and Vosselman, 1999,

[7] which is especially important when using adaptive thresholds

[8] Sun and Salvaggio [2013] and Sampath and Shan [2010] determine a curvature measure based on the ratio of the local neighborhood's largest eigenvalue to the sum of all three eigenvalues.

[9] Individual boundary points may be needed for fitting step edges to neighboring boundary points of adjacent segments, see Section 3.2.3; tracing of the boundary points to a closed outline is therefore not necessary.

Matei et al., 2008]. Sun and Salvaggio [2013] create segment boundaries by overlying a 2D grid to their segmented point cloud: Each grid edge connecting an empty and an occupied grid cell is chosen as border edge. Very similarly, Zhou and Neumann [2008] define boundaries by tracing the closest LiDAR points to those edges. Rottensteiner [2003] define separation boundary lines between adjacent segments from the Delaunay triangulation: Differently segmented points connected by triangulation edges are boundary points, and the corresponding Voronoi edges form the boundary. Dorninger and Pfeifer [2008], Kada and Wichmann [2012] and Sampath and Shan [2007] use a modified convex hull approach called alpha shapes, in which each next boundary vertix is determined only from the local neighborhood of the previous vertex. If the local neighborhood is determined by a fixed radius, alpha shapes produce only satisfactory results if the point density is regular. Therefore, Sampath and Shan [2007] define the neighborhood with a rectangle whose extents and orientation depend on the along-track and across-track LiDAR sampling characterisitics. [Wang and Shan, 2009] identify unconnected boundary points by creating the convex hull of each point's local neighborhood. If the point is a vertex of this convex hull, it is chosen as a building boundary vertex. Lafarge and Mallet [2012] determine each boundary point based on its distance to the line fitted through its neighborhood.

Boundary regularization

Boundary tracing only delivers irregularly shaped polygons with a high number of polygon vertices and high variability of edge directions. Regularization procedures intend to create simplified polygons by reducing these irregularities to a minimum. The (Ramer-)Douglas-Peucker algorithm, a polygon simplification method, is often used for regularization. Line segments are regularized by connecting the farthest vertices in the irregular polygon, such that the distances of all skipped polygon points from the new line are below a threshold [Sampath and Shan, 2007, Sohn et al., 2012]. Dorninger and Pfeifer [2008] reduce the number of vertices by analyzing the angular deviations of subsequent boundary edges. A new line segment is computed by averaging iteratively the boundary edge directions which are added to the regularized line segment. If the angular deviation of a polygon edge to the regularized line segment is smaller than a threshold, it is added to the line segment. Orthogonality and parallelity to the principal building direction is enforced for those line segments whose angular deviation to one of these conditions is below a threshold [Dorninger and Pfeifer, 2008]. The principal building direction can be chosen parallel to the longest line segment of the building boundary [Rau and Lin, 2011, Dorninger and Pfeifer, 2008].

3.2.3 Modeling Procedures

The goal of modeling procedures is to create closed polyhedral 3D building models consisting of vertices and faces, from a segmented building point cloud.

Heuristic modeling techniques

Traditional approaches analyze the segments' neighborhood relations and create and connect intersection edges and step edges between adjacent roof segments using different heuristics.

Extracting line segments: Simple heuristics neglect step edges and only intersect segment planes with each other and with vertical walls at the building boundary [Maas and Vosselman, 1999]. For

defining step edges between adjacent segments and at the building boundary, many approaches define 3D lines for neighboring boundary points, e.g. by using alpha shapes [Dorninger and Pfeifer, 2008] (Section 3.2.2). Lafarge and Mallet [2012] find 3D line segments by clustering the building boundary points in Hough space. Rottensteiner [2003] and Vosselman and Dijkman [2001] identify step edges by analyzing the distances of the boundary to the corresponding segments' intersection line. If the RMSE of the distances is above a threshold, a step edge is formed by fitting a 2D polygon to the boundary points. Sohn et al. [2008] find the directions of step edge line segments in a brute-force search by implementing an adapted version of the Kirsch's compass filter[10]. [Rau and Lin, 2011] project the building point cloud to 2D planes orthogonal and parallel to the principal building direction. Then, 2D line segments are extracted applying a Hough line detector, and backprojected to 3D space, using the LiDAR points of the corresponding Hough histrogram bin.

Connecting line segments: 3D line segments can be connected to closed models by intersecting their projections on the XY-plane [Dorninger and Pfeifer, 2008]. Rau and Lin [2011] produce constrained Delaunay triangulation from the vertices of 2D projections of their line segments. Triangulation edges which are not separated by line segments are merged. Sohn et al. [2008] use those projections for space decomposition: The building point cloud's rectangular bounding box is iteratively subdivided along the optimal line segments, while maximizing planar homogeneity in the respective sub-spaces.

Structural modeling techniques

Structural approaches estimate 3D vertices by local error minimization. Dual contouring (DC) is such an approach: After overlying a regular 2D or 3D grid to the building point cloud, local surface planes are sampled at the vertices of this grid Fiocco et al. [2005] and Zhou and Neumann [2010][11] The building model's 3D vertices are estimated by minimizing Quadratic Error Functions (QEF) which are constructed from this grid data. The vertices are connected according to their adjacency in the sampling grid.

Lafarge and Mallet [2012] apply a Markov-Random-Field energy minimization approach for label propagation: The building point cloud is transformed to a label image according to the segmentation. Then, the labels are propagated while considering previously extracted 3D line segments as restrictions. Finally, closed 2D polygons are extracted from the resulting label image and polyhedral building structures are created by projection into 3D using the line segments, and by triangulation of the vertices.

Building models from structual approaches are typically represented by more 3D polygon faces than those of heuristic approaches. This is due to the fact that the latter usually assume and enforce planarity of the roof segments, while structural modeling allows also non-planar shapes, which are represented by triangulation. Building models from strucutal methods attain a better fit to the input data, but may require a high regularization effort, depending on the applications' demands concerning model simplicity.

[10] Kirsch's compass filter is an edge detection method from image processing

[11] Zhou and Neumann [2010] additionally sample the boundary lines between adjacent roof height layers. A detailed description of this metho is given in chapter 4.

3.2.4 Model Regularization

Model regularization procedures correct and reduce the 3D line segments of the building model to increase model simplicity. Dorninger and Pfeifer [2008] and Zhou and Neumann [2010] achieve directional correction of the 3D line segments by enforce orthogonality and parallelity of the step edges to the principal building boundary. Additionally, the building model's level of detail can be reduced by merging those 3D model's vertices which are close to each other.

[Sohn et al., 2012] combine both intentions (directional correction and model simplification) by a "Minimum Description Length (MDL)" approach: For each two adjacent line segments in the building model, different model hypotheses are created by shifting the central vertex to different positions[12] along both lines. Such a position may also coincide with one of the other two vertices in this situation. The description length (DL) of each hypothesis includes a measure for the fit of the hypothesis to the data, and a measure for the model's simplicity[13]. The hypothesis with minimum DL is chosen for correction.

3.3 Model-Driven Reconstruction

Most model-driven reconstruction approaches decompose the building point cloud into different regions, because only a minor part of the buildings in a city (40 to 50 % in Western European cities [Haala and Kada, 2010]) can be represented with one simple parametric model.

3.3.1 Pointcloud Decomposition

The building point cloud can be partitioned using roof segmentation as described in section 3.2.1 [Verma et al., 2006], or in terms of larger regions, where one region is assumed to be representable by one of the parametric building model of the catalogue. Footprint decomposition can be achieved e.g. by using previously detected step edges [Vosselman and Dijkman, 2001], or by analyzing the building boundary's line segments [Kada and McKinley, 2009, Haala et al., 1998]. You et al. [2003] require a user-input to determine the footprint regions and to select the appropriate model; only the model parameters are determined automatically.

3.3.2 Parametric Models

The model catalogue has to contain the most probable roof shapes in a parametric description, such as flat roofs, shed roofs[14], gabled roofs[15] and hipped roofs[16] [Kada and McKinley, 2009, Haala and Brenner, 1999, Vosselman and Dijkman, 2001, Zhang et al., 2012]. The catalogue can also contain models for corners where basic roof shapes connect in a right angle or in a T-shape Kada and McKinley [2009]. [You et al., 2003] compose their models from standard computer graphic shapes such as planes, cubes, polyhedra, cylinders, spheres and ellipsoids.

[12] The positions are found according to a compass filter, see Section 3.2.3.
[13] Considering the number of vertices, the variability of line directions and angles between ajacent line segments.
[14] Shed roof: One sloped roof plane.
[15] Gable roofs: Two orthogonal, sloped roof planes intersecting at the ridge.
[16] Hipped roofs: Four orthogonal, sloped roof planes of wich two intersecting at the main ridge.

3.3.3 Model selection and parameter fitting

Kada and McKinley [2009] select the model by computing the percentage of points whose local normal[17] is similar to the catalogue model. Haala et al. [1998] estimate the model parameters by least squares minimization of the DEM pixel's vertical distances to the model. Verma et al. [2006] determine model parameters by a RANSAC fitting procedure. For determining footprint extensions, building orientation and roof type, Maas and Vosselman [1999] compute height-weighted invariant moments from the point cloud, as well as from rasterized versions of the parametric models. Additional building superstructures (dormers and chimneys) are fit to those points whose distances to the model is larger than a threshold.

Verma et al. [2006] and Oude Elberink and Vosselman [2009] define a topology graph in which neighboring roof segments are represented as vertices connected by edges. These edges are labeled according to the relations of the segments' plane normals to each other. Equally, small graphs are created for parametric models of the segment junctions. The graphs are matched to each other by a brute-force search.

3.4 Evaluation Techniques

Evaluation methods can be categorized into detection-based methods and geometric methods. Both techniques can be used to evaluate results from segmentation and reconstruction.

3.4.1 Detection-based evaluation

Detection-based evaluation measures can be derived using object-based and pixel-based approaches [Awrangjeb and Fraser, 2014].

Object-based evaluation methods count the number of detected (True Positives TP) and missed (False Negatives FN) estimated building segments and reference segments [Rutzinger et al., 2009]. False Positives (FP) are estimated building segments which have no correspondence with any reference. Those correspondences can be identified according to the overlap percentage[18] of the estimated segment's boundary with the reference segment's boundary [Rottensteiner et al., 2012a] or by checking whether the center point of the detected segment is located within the reference segment's boundary [Pfeifer et al., 2007, Rutzinger et al., 2009]. A minimum limit on the segment area area can be set for counting TP, FP and FN [Rottensteiner et al., 2012a].

Pixel-based appraoches rasterize the estimated segments and their references into images and determine TP, FN and FP according to corresponding pixels. Pixel-based approaches have the advantage to evaluate more accurately, because misalignments between the estimated and reference segments at the building boundaries influence the evaluation measures. However, reliable evaluation results are only produced if perfectly coregistered reference data is available Rutzinger et al. [2009].

Detection-based evaluation parameters are:

[17] The local normal of each point can be determined as explained in Section 3.2.1.

[18] The *ISPRS Test Project on Urban Classification and 3D Building Reconstruction* defines true positives by a 50% overlap.

- Completeness, detection rate or producer's accuracy [Rutzinger et al., 2009, Schuster and Weidner, 2003, Pfeifer et al., 2007, Shufelt, 1999]: $C_m = TP/(TP + FN)$
- Correctness or user's accuracy [Rutzinger et al., 2009, Pfeifer et al., 2007] : $C_r = TP/(TP+FP)$
- Quality rate [Rutzinger et al., 2009, Schuster and Weidner, 2003, Shufelt, 1999]: $Q = TP/(TP+FP+FN)$
- False alarm rate [Schuster and Weidner, 2003]: $P_fa = FN/(TP + FN)$
- Branch factor [Schuster and Weidner, 2003, Shufelt, 1999]: $P_b = FN/TP$
- Type 2 error [Schuster and Weidner, 2003]: $\beta = FP/(TP + FN)$

Purely object-based evaluation parameters are the number of oversegmented segments [Geibel and Stilla, 2000, Rottensteiner et al., 2012a], the number of undersegmented segments [Geibel and Stilla, 2000, Rottensteiner et al., 2012a] and the number of segments which are both over- and undersegmented [Rottensteiner et al., 2012a].

The disadvantage of object-based evaluation is to be uniquely detection-based, i.e. the accuracy of the building models' line segments is not evaluated.

3.4.2 Geometric evaluation

Geometric evaluation measures represent the accuracy of the building models in planar (horizontal) directions and in heights. Tarsha-Kurdi et al. [2007] evaluate different building reconstruction methods against references by computing the differences in ridge height, eaves height, segment planes' slope angles, and in the horizontal footprint extents. This evaluation technique is limited to simple building shapes[19].

Horizontal accuracy can be estimated from the RMSE of all estimated roof segments' polygon points to the closest reference segment edges in the XY-plane, or viceversa (computing the horizontal RMSE of all reference segments' polygon points to their closest estimated segment edges) [Zhang et al., 2012]. [Rutzinger et al., 2009] Vertical accuracy can be obtained by computing the vertical distances of segment polygons to the model faces[20]

If no reference data is available, the estimated building models can be compared to the input data by computing the orthogonal distances between LiDAR points to their closest segments [Oude Elberink and Vosselman, 2009, Zhou and Neumann, 2010] and viceversa [Oude Elberink and Vosselman, 2009].

[19] Tarsha-Kurdi et al. [2007] only use simple gable roofs for evaluating the different methods.

[20] Vertical distances can also be computed in a pixel-based approach after converting the building models and the reference to a DEM [Rutzinger et al., 2009,?].

4 Proposed Workflow

4.1 Overview of the Workflow

Inputs to the proposed workflow (Fig. 4.1) are a single pulse return LiDAR pointcloud of an urban scene - preferably the first pulse return -, and a corresponding orthoimage with red and near-infrared band. The LiDAR point cloud is required to be aquired during one acquisition "strip", in order to avoid abrupt point density changes within a building region. For presegmentation (Section 4.2), building boundary polygons (footprints) are assumed to be available. For roof segmentation (Section 4.3), a robust triangulation-based region growing technique is developed, using adaptive thresholds, inspired by Nurunnabi et al. [2012]. Segment boundary polygons are created by a convex hull collapsing approach (Section 4.4). The 2.5D Dual Contouring approach as proposed by Zhou and Neumann [2010] is adapted for building modeling (Section 4.5).

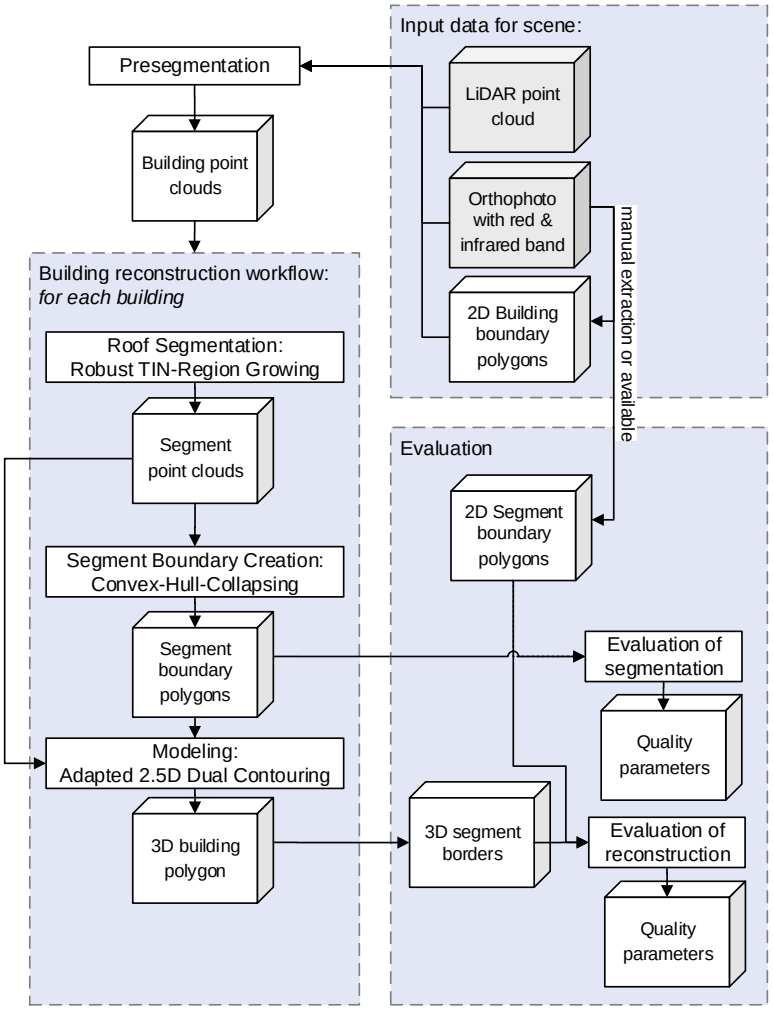


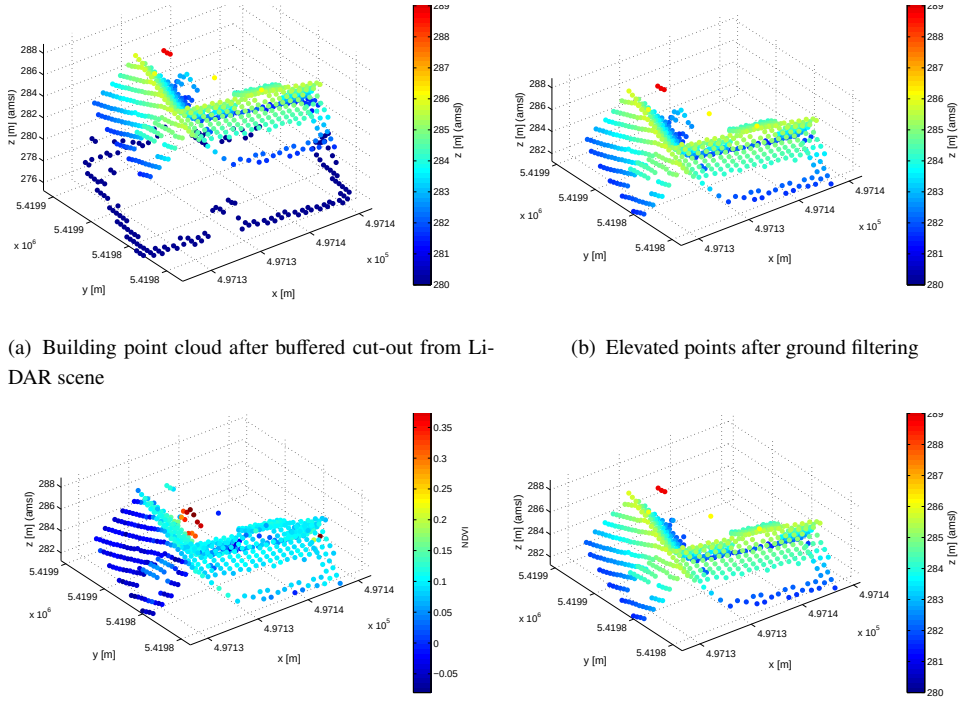
Figure 4.1: Overview of the proposed workflow; grey boxes are the basic input data from which the other input data can be derived; dotted elements concern only the evaluation;

4.2 Presegmentation

Individual building point clouds are extracted from the LiDAR scene using building footprints $\mathbf{P}_{pre}$. As this selection may contain points on nearby vegetation and terrain, additional filtering procedures are necessary.

For ground filtering, a DTM is created from the rasterized and interpolated LiDAR point cloud (DEM). The DTM is created using a progressive morphological filter similar to Zhang et al. [2003]. The DEM surface is iteratively smoothed by an image opening procedure. At each iteration, all DEM pixels whose distance to the opened DEM (ground distance) is larger than a height threshold h are set to the values of the opened DEM. The height threshold is increased at each iteration by the average of the points with ground distances $< h$. The opening filter size is increased at each iteration, and the procedure stops when the filter is larger than the maximum expected building size. A DTM is created from a linear interpolation of all points whose difference between the original DEM and the smoothed surface is smaller than h.

For vegetation filtering, the orthoimage is used to calculate the NDVI (Normalized Difference Vegetation Index) for each pixel: $NDVI = (IR - R)/(IR + R)$, where IR is the infrared intensity and R is the red intensity of the pixel. LiDAR points falling into an orthoimage pixel with an NDVI higher than a threshold are removed from the set of elevated points (Fig. 4.2).[1]



(a) Building point cloud after buffered cut-out from LiDAR scene

(b) Elevated points after ground filtering

(c) NDVI values of the elevated points

(d) Elevated points after vegetation filtering (final building point cloud)

FIGURE 4.2: Presegmentation procedure. Different colors of the points indicate the point heights above mean sea level (amsl) in a),b) and d), and the NDVI values of the points in c).

[1] The NDVI is high for pixels where vegetation is dominating, because photosynthetic procedures cause the reflective spectrum of plants to show a high discrepancy between infrared and red frequencies.

4.3 Robust TIN-based Region Growing Segmentation

4.3.1 Overview

In this work, a region growing (RG) based segmentation technique was chosen, after having verified the disadvantages of surface fitting segmentation techniques compared to region-based approaches[2].

The proposed segmentation method is based on the assumption that LiDAR points at the borders of two intersecting roof segments (e.g. in a gable roof) are more likely to be shared by both segments than to belong to just one of these segments. This assumption presumes the use of dense first pulse LiDAR data, where the first pulse indicates the highest object (e.g. ridge lines) in the LiDAR footprint. In contrast to most existing RG approaches, the proposed segmentation method is based on the growing of triangles of a Triangulated Irregular Network (TIN) instead of points. TIN-based RG minimizes the spaces in between the segmented regions by allowing for a double (or multiple) assignment of points, if the triangles this point is a part of are segmented differently.

A major challenge of TIN-based RG is that the triangle normals vary locally due to variations in point density and vertical point accuracy [Awrangjeb and Fraser, 2014]. Even with constant vertical point accuracy, the triangle normals vary according to inhomogeneous point density within the building point cloud. Therefore, robust region growing criteria have to be applied.

Robust TIN-based RG segmentation (RTINS) (Fig. 4.4) starts with a Delaunay triangulation of all roof points (see Section 3.2.1). Triangles whose normal elevation is larger than a threshold $T_{elevation}$[3], or whose longest triangle edge is longer than the expected maximum regular point spacing Sp_{max} (Section 4.3.2), are excluded from the RG procedure as outliers.

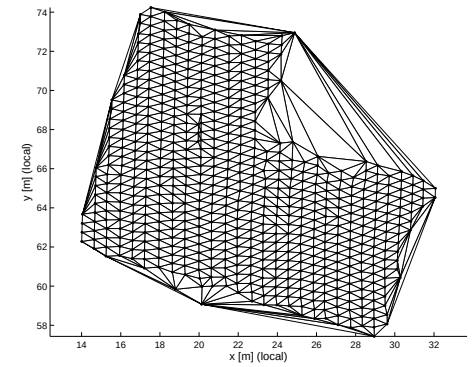
A seed is selected according to a minimum local unevenness factor LUF (Fig. 4.3), which is determined for each triangle as the weighted standard deviation of the triangle normals in the local neighborhood N_l (Section 4.3.2). Higher weights are given to larger triangles, in order to compensate for the sensibility of the normal variation to the variation in point density.

$$LUF_t = \sum_{k}^{K_t} \left(\frac{A_k}{\sum\limits_{k} A_k} \cdot mean \begin{pmatrix} n_{x,k} - \bar{n}_x \\ n_{y,k} - \bar{n}_y \\ n_{z,k} - \bar{n}_z \end{pmatrix} \right), \tag{4.1}$$

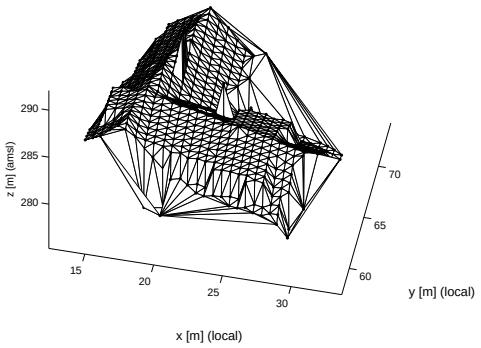
where A_k is equal to the area and $\mathbf{n}_k = \begin{pmatrix} n_{x,k} & n_{y,k} & n_{z,k} \end{pmatrix}^T$ is the normal vector of the k-th of K_t triangles in the neighborhood of the t-th triangle; $\bar{n}_x$, $\bar{n}_y$ and $\bar{n}_z$ are the means of all $n_{x,k}$, $n_{y,k}$, and $n_{z,k}$, respectively; $k = (1, ..., K_t)$.

[2] The implementation of a RANSAC-based segmentation showed that surface-fitting segmentation requires a high post-processing effort for segmentation correction, see Fig. 3.2

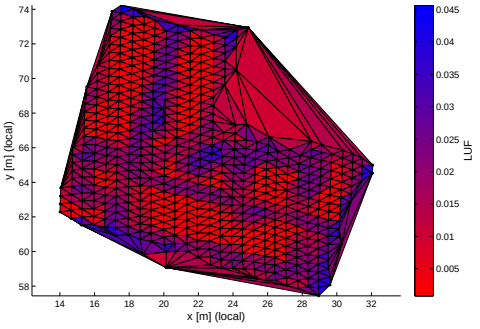
[3] $T_{elevation} = 35°$, according to experiments and visual interpretation. The procedure is not very sensitive to $T_{elevation}$, because even if triangles are erroneously not deteted as outliers, they are usually not include into any region during the RG process.



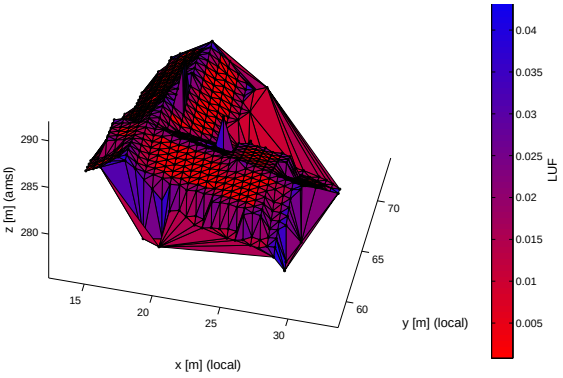
(a) Delaunay triangulation of the LiDAR points, top view



(b) Delaunay triangulation of the LiDAR points, side view



(c) Local unevenness factors (LUF) of the triangles in the TIN, top view



(d) Local unevenness factors (LUF) of the triangles in the TIN, side view

FIGURE 4.3: Local unevenness factors (LUF) of the triangles in the TIN. Each vertex in the triangulation corresponds to one LiDAR point.

A region is grown until no more neighboring triangle fulfills the region growing criteria defined in section 4.3.3. If the region does not fulfill either of the two conditions that

- the region has to contain a minimum number of triangles $T_{\#T}$ and
- the area of the region has to be larger than a threshold T_{area},

it is discarded. Then, a new seed is determined and the segmentation continues until the minimum LUF of the remaining triangles exceeds a threshold. A final post-processing step corrects for improbable segmentation results.

4.3.2 Neighborhood, plane normals and point pattern

The *Large Neighborhood* N_l of a triangle is defined as all triangles sharing at least one point with the examined triangle. The *Small Neighborhood* N_s of a triangle is defined as all triangles sharing an edge, i.e. two points, with the examined triangle, i.e. the number of neighbors in N_s is 3 at maximum.

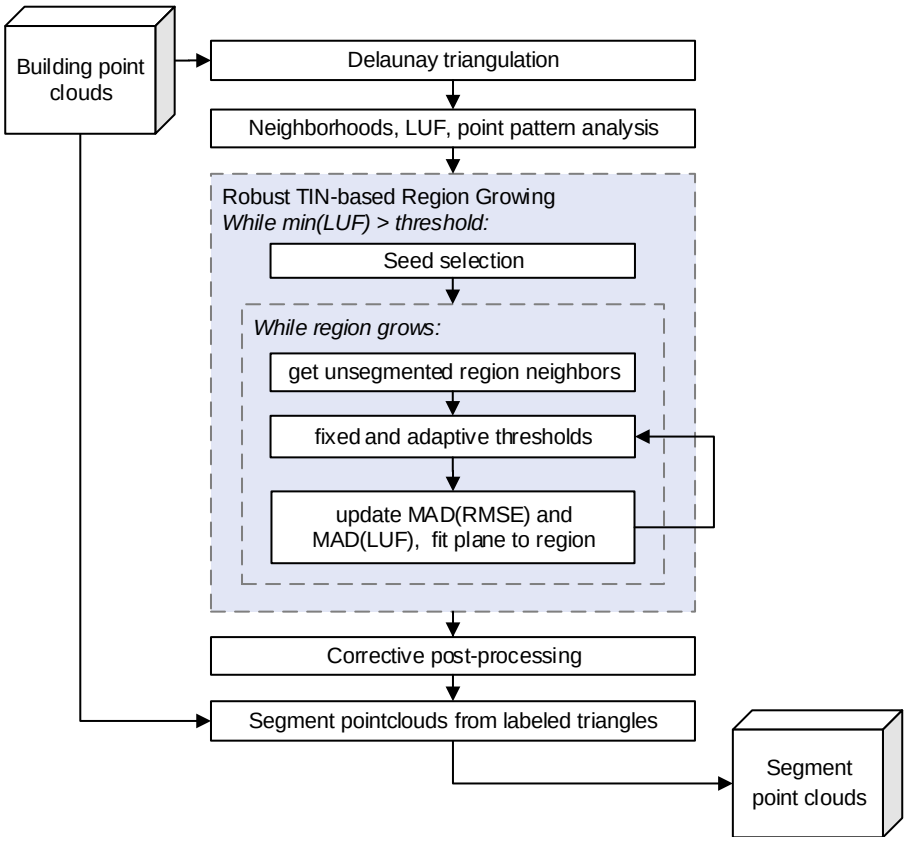


FIGURE 4.4: Overview of the segmentation

In all cases where a plane is described (e.g. the triangles' planes), the Hesse normal form of the plane equation is used:

$$(\mathbf{n} \times \mathbf{p}) + d = n_x x + n_y y + n_z z + d = 0, \tag{4.2}$$

where $\mathbf{n} = \begin{pmatrix} n_x & n_y & n_z \end{pmatrix}$ is the plane normal and $\mathbf{p} = \begin{pmatrix} x & y & z \end{pmatrix}$ is a point on the plane. If $|\mathbf{n}| = 1$, d is the distance of the plane from the origin.

The minimum and maximum regular point spacings Sp_{min} and Sp_{max} are estimated from the distribution of the lengths of those triangle edges, whose midpoints are inside the building boundary polygon $\mathbf{P}_{pre}$:

$$Sp_{max} = median(L_{max,\mathbf{P}_{pre}}) + a_{mad,Sp} \cdot mad(L_{max,\mathbf{P}_{pre}}) \tag{4.3}$$

$$Sp_{min} = median(L_{min,\mathbf{P}_{pre}}) + a_{mad,Sp} \cdot mad(L_{max,\mathbf{P}_{pre}}), \tag{4.4}$$

where $a_{mad,Sp}$ is chosen between 1 and 3, and $L_{max,\mathbf{P}_{pre}} = \left\{L_{max} | mean(L_{max}) \in \mathbf{P}_{pre}\right\}$, and $L_{min,\mathbf{P}_{pre}} = \left\{L_{min} | mean(L_{min}) \in \mathbf{P}_{pre}\right\}$, are the longest and shortest edges of their respective triangles.

4.3.3 Region Growing Criteria

Each unsegmented triangle within the neighborhood N_l of a all triangles in segment S is tested for one fixed and two adaptive thresholds[4].

- *Fixed threshold:* The angular difference $\Delta\Phi$ between the candidate triangle's normal $\mathbf{n}_c$ and at least one of the neighboring segment triangles' normal $\mathbf{n}_s$ shall be smaller than an angular threshold $T_{\Delta\Phi}$:

$$\Delta\Phi = \arccos\left(\mathbf{n}_c^T \cdot \mathbf{n}_s\right) < T_{\Delta\Phi} \tag{4.5}$$

- *Adaptive threshold:* The $RMSE_c$ of the candidate's triangle points from the plane which is fitted to the current segment S shall be smaller than a threshold T_{RMSE}:

$$RMSE_c < T_{RMSE} = median(d_{i\in S}) + a_{mad,RMSE} \cdot MAD(d_{i\in R}), \tag{4.6}$$

  where $a_{mad,RMSE}$ is chosen between 1 and 3, and $d_{i\in S}$ are the distances of all LiDAR points of the region to the plane fitted to the region.
- *Adaptive threshold:* The deviation of the candidate's LUF_c from the current average $mean(LUF_{i\in S})$ of all triangles in the segment S' shall be smaller than an adaptive threshold $T_{\Delta LUF}$:

$$\Delta LUF = LUF_c - mean(LUF_{i\in S}) < T_{\Delta LUF} \tag{4.7}$$

$$T_{\Delta LUF} = median(LUF_{i\in S}) + a_{mad,LUF} \cdot MAD(LUF_{i\in S}), \tag{4.8}$$

  where $a_{mad,\Delta LUF}$ is chosen between 1 and 3, and $LUF_{i\in S}$ are the local unevenness factors of all points which are already in the segment.

4.3.4 Segmentation Correction

Undesired segmentation results are situations where triangles of one segment "protrude" into another segment, i.e. where the dominant label of a triangle's small neighborhood N_s is not equal to the triangle's label. This is corrected by reassigning the respective triangles to the locally most dominant label, which is determined according to their number of occurrence. In case that non-assigned neighbors are dominating, the triangle's label is removed.

4.4 Segment Boundary Polygons

4.4.1 Convex hull collapsing

Segment boundary polygons are created by iteratively collapsing the convex-hull of the segment points, where the decision for collapsing considers the point pattern (Section 4.3.2). The polygon's edges are iteratively subdivided by adding an additional point to the border polygon, if the length of the line segment to be corrected (L_{ls}) is longer than an adaptive threshold T_{refine} (Fig. 4.5). This

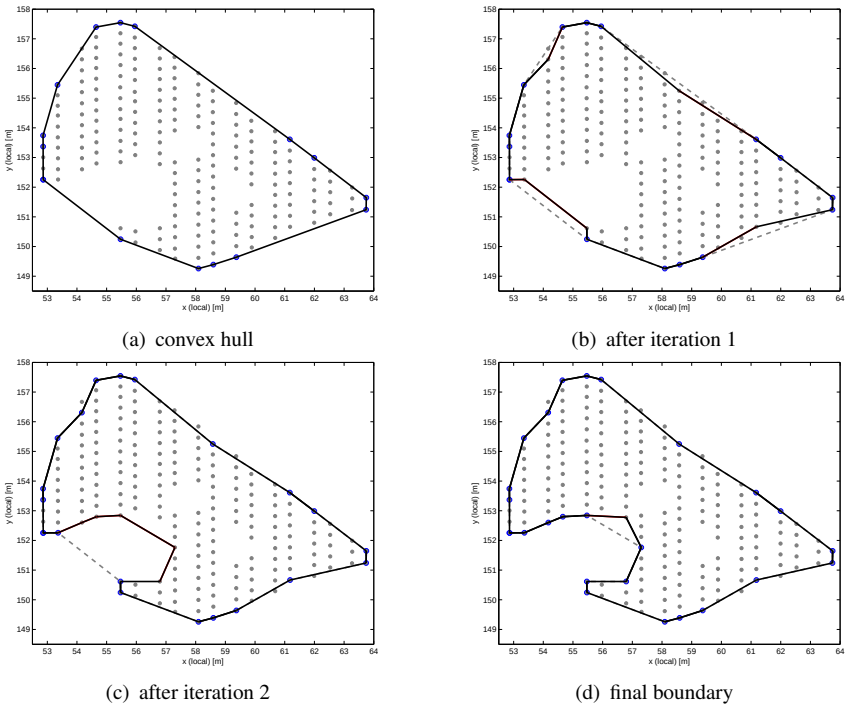
[4] The adaptive thresholds are inspired by Nurunnabi et al. [2012]: To test whether a candidate belongs to a group, the tested value's deviation from the average value of the group shall not exceed a threshold. This threshold is defined as the sum of the group's values' median and a weighted MAD of the group's values (see section 3.2.1).

threshold is computed from the minimum angular difference ϑ of the two possible correcting line segments' directions to the main sampling direction $Sdir_{min}$. $Sdir_{min}$ is the direction of minimum regular point spacing SP_{min} and determined in section 4.4.2.

The intention is to set T_{refine} equal to Sp_{min}, if the angle ϑ is small, and to set T_{refine} equal to Sp_{max}, if the angle ϑ is large. If $\vartheta = 0°$, the next LiDAR point along the boundary is expected at a distance equal to Sp_{min}. Additionally it is assumed that $Sdir_{min}$ and $Sdir_{max}$ are perpendicular, such that if $\vartheta = 90°$, the next LiDAR point along the boundary is expected at a distance equal to Sp_{max}. This approach ensures the boundary to sample the building point cloud more detailed along $Sdir_{min}$, while allowing the border polygon to "cross" the pointcloud in other directions using longer edges.

$$T_{refine} = a_{T_{refine}} \cdot \left[Sp_{min} + (Sp_{max} - Sp_{min}) \cdot \cos(\vartheta)\right], \tag{4.9}$$

where a factor $a_{T_{refine}} = 2$ is introduced to compensate for missing data points.



(a) convex hull (b) after iteration 1

(c) after iteration 2 (d) final boundary

FIGURE 4.5: Iterative convex hull collapsing applied on one roof segment. Dark line: collapsed boundary resulting after the respective iterations. Dotted lines: collapsed boundary resulting from the previous iteration

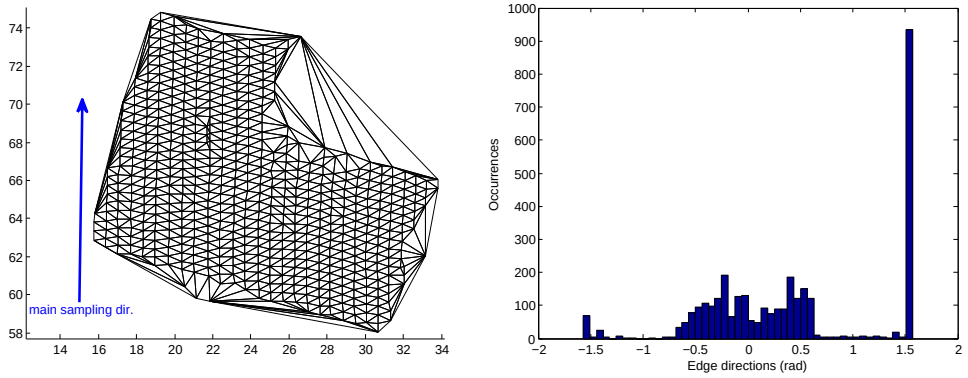
The LiDAR point which minimizes a balanced distance measure D_p is chosen for correcting the boundary line segments. D_p is computed from the points' orthogonal distances from the line segment (d_l) and its orthogonal distances from the line segment's normal, passing through the line segment's center (d_{nl}):

$$D_p = w * d_l + d_{nl}, \tag{4.10}$$

where experiments have shown that a weighting factor between $w = 1$ and $w = 1.5$ delivers satisfactory results.

4.4.2 Main sampling direction

Assumed that the LiDAR pointcloud is acquired by a line scanner with two main regular point spacing directions $Sdir_{min}$ and $Sdir_{max}$, which are orthogonal. The main sampling direction $Sdir_{min}$ is determined from a histogram analysis of all triangle edges of the Delaunay triangulation which are inside the building boundary polygon $\mathbf{P}_{pre}$. Triangle edges connecting closely and regular sampled points have a smaller directional variation, such that the histogram shows a clear peak (Fig. 4.6). The histogram bin center with the highest associated histogram value is chosen as the main sampling direction.



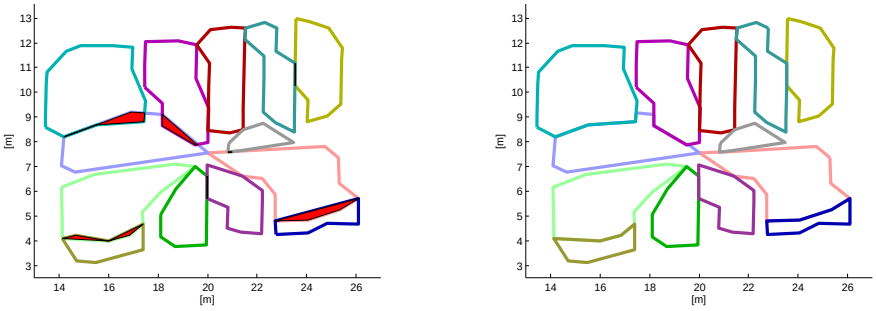
(a) Delaunay triangulation of the building point cloud, arrow indicating $Sdir_{min}$

(b) Histogram of directions of triangulation edges, bincenter with highest value selected as most dominant sampling direction

FIGURE 4.6: Identification of minimum regular sampling direction

4.4.3 Segment boundary separation

Subsequent reconstruction steps require the individual segments to be clearly separated from each other without overlaps. Therefore, overlapping areas between two segment border polygons are removed by cropping one of the respective border polygons, depending on the dominant segmentation label of all LiDAR points inside the overlap. In case there are no points inside the overlap or if the dominant label cannot be determined clearly, the polygon sharing the lower number of border polygon vertices with the overlap area's polygon is reduced. In case the number of vertices is equal for both polygons, the segment border polygon with the larger area is cropped (see Fig. 4.7).



(a) Boundaries of different segments shown in differnt colors (colors arbitrary). Overlapping areas shown in red.

(b) Corrected segment boundaries, where the segment boundaries have been cropped by the overlaps.

FIGURE 4.7: Correction of segment boundaries to avoud overlaps

4.5 Modeling: Adapted 2.5D Dual Contouring

4.5.1 Overview

This work presents an adapted and improved version of the 2.5D dual contouring algorithm of Zhou and Neumann [2010]. The principle of 2.5D dual contouring is introduced in section 4.5.2. A 2D grid is overlain on the segmented point cloud. For each grid cell, a Quaratic Error Function (QEF) is constructed from the local grid data, which is obtained from the LiDAR points near the grid's vertices and cells (Section 4.5.3). For each grid cell, the minimization of the QEF results in one or more 3D vertices of the polyhedral building model (Section 4.5.4). For reducing the number of 3D vertices, QEF solutions are iteratively merged using a quadtree (Section 4.5.5). Finally, the 3D vertices are connected according to their adjacency in the quadtree to form polygon faces of the building model (Section 4.5.6).

Input to the proposed workflow is a detailed segmentation of the building point cloud (see Section 4.3). This is the main difference to the method of Zhou and Neumann [2010], whose input point cloud is segmented only into height layers, i.e. the boundaries of their segments are always at step edges. All further differences are indicated in the detailed description.

4.5.2 2.5D Dual Contouring

2D Case: Dual Contouring (DC) is a method originating in computer vision[5] [Ju et al., 2002]. The 2D case of DC is illustrated by the determination of a 2D boundary polygon between two or more segments. For estimating the boundary polygon's vertices, a 2D grid is overlain on the segmented point cloud in the XY plane (Fig. 4.9 a)). For each grid cell containing data from more than one segment, local boundary lines are computed for each segment pair S_k and S_l (Fig. 4.9 b)). For each cell, an optimal vertex $\hat{\mathbf{X}}_{\mathbf{2D}} = \left(\hat{x}, \hat{y}\right)^T$ is found by minimizing its distances to all boundary lines which intersect with the grid cell's borders (Fig. 4.9 c)). This minimization problem is expressed with the

[5]DC is developed from the Extended Marching Cubes (EMC) method

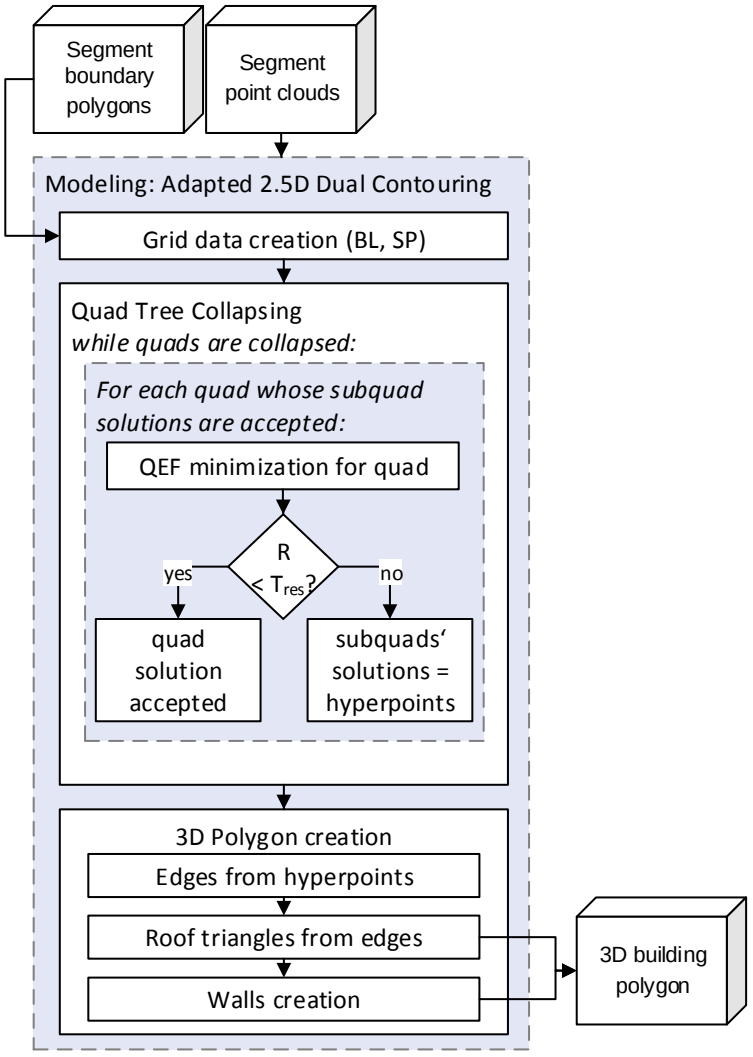


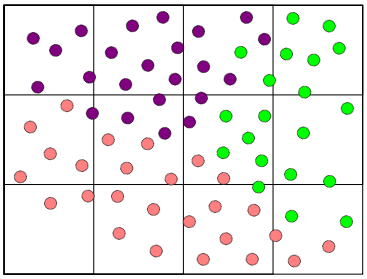
FIGURE 4.8: Workflow of the building modeling procedure

2D-QEF:

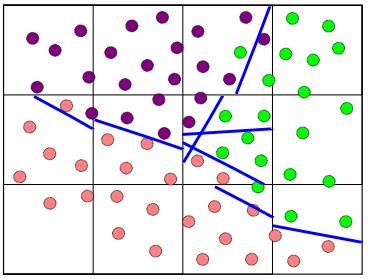
$$\hat{\mathbf{X}}_{2D} = \underset{\mathbf{X}_{2D}}{\operatorname{argmin}} \left\{ \sum_{k=1}^{M-1} \sum_{l=2, l\neq k}^{M} \sum_{i=1}^{I_{k,l}} \left[\mathbf{n}_{i_{(k,l)}}^{T} \left(\mathbf{p}_{i_{(k,l)}} - \begin{pmatrix} x \\ y \end{pmatrix} \right) \right]^2 \right\}, \tag{4.11}$$

where $I_{k,l}$ is the number of boundary lines between the k-th and the l-th segment, $k,l \in [1,...,M]$, $\mathbf{n}_{i_{(k,l)}}$ is the normal of the i-th boundary line between S_k and S_l, and $\mathbf{p}_{i_{(k,l)}}$ is a point on that boundary line, and x and y are horizontal coordinates on the x-y-plane.

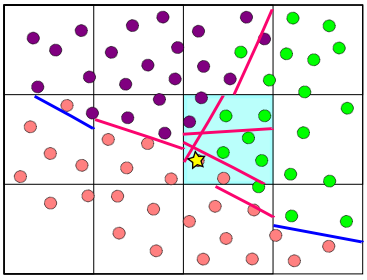
After an optimal vertex is computed for each grid cell, vertices which are adjacent in the grid are connected to a polygonal line (Fig. 4.9d). Grid cells which share a grid edge are considered adjacent. Analogously, a 3D DC approach computes 3D boundary vertices by dividing a 3D point cloud space into voxels and connects them to polygon faces Ju et al. [2002].



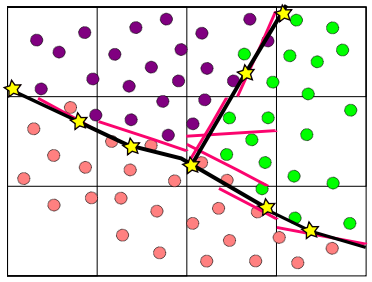
(a) 2D-grid overlain on a segmented point cloud; differently colored points are segmented to differnt regions



(b) Blue lines: local boundary lines for each pair of segments in a cell



(c) 2D-QEF minimization in the blue cell: All local boundary lines which are valid for this cell are plotted in pink. Valid boundary lines are those which have a contact-point / interection point with the cell border. The yellow star is the QEF solution, i.e. one optimal vertex of the polygonal boundary line.



(d) After computing 2D-QEF solutions for all cells, the solutions (yellow stars) are connected to the polygonal boundary line (black line) according to their adjacency in the grid. solutions

FIGURE 4.9: Illustration of the DC approach in a 2D case

2.5D Case: The idea of 2D DC is expanded to 2.5D case for building reconstruction: Corresponding to the optimal boundary line's vertices, the goal is to determine *hyperpoints* by QEF minimization. Hyperpoints are defined by one x- and one y-coordinate, and one or more z-coordinates, depending on whether a step edge or intersection edge is represented (Fig. 4.10). Each z-coordinate corresponds to one vertex of the 3D building polygon. Additionally to local 2D boundary lines (**BL**), local 3D

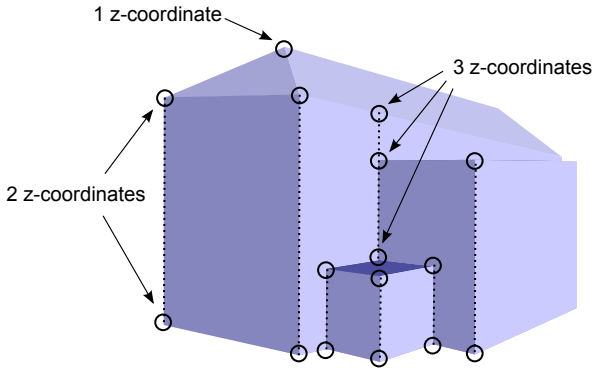


FIGURE 4.10: Hyperpoints at intersection edge (one z-coordinate) and step edges (more than one z-coordinate)

surface planes (**SP**) are determined for each vertex of the 2D grid. Each **SP** is given by a normal **m**

and a point $\mathbf{q}$. The estimated hyperpoint $\hat{\mathbf{X}}$ has a minimum 2D distance $E_{2D}(\mathbf{X})$ to the **BL**, and a minimum 3D distance $E_{3D}(\mathbf{X})$ to the **SP**. The hyperpoint $\hat{\mathbf{X}} = \begin{bmatrix}\hat{x} & \hat{y} & \hat{z}_1 & \hat{z}_2 & \dots & \hat{z}_M\end{bmatrix}^T$ with k z-coordinates is computed by expanding equation 4.11 to [Zhou and Neumann, 2010]:

$$\begin{aligned}\hat{\mathbf{X}} &= \underset{\mathbf{X}}{\operatorname{argmin}} \{E_{2D}(\mathbf{X}) + E_{3D}(\mathbf{X})\} \\ &= \underset{\mathbf{X}}{\operatorname{argmin}} \left\{ \sum_{k=1}^{M-1} \sum_{l=2, l\neq k}^{M} \sum_{i_{(k,l)}}^{I_{(k,l)}} \left[\mathbf{n}_{i_{(k,l)}}^T \left(\mathbf{p}_{i_{(k,l)}} - \begin{bmatrix} x \\ y \end{bmatrix} \right) \right]^2 + \sum_{k=1}^{M} \sum_{j_k=1}^{N_k} \left[\mathbf{m}_{j_k}^T \left(\mathbf{q}_{j_k} - \begin{bmatrix} x \\ y \\ z_k \end{bmatrix} \right) \right]^2 \right\}, \end{aligned} \quad (4.12)$$

where $\mathbf{X} = \begin{bmatrix}x & y & z_1 & z_2 & \dots & z_M\end{bmatrix}^T$. $\mathbf{n}_{i_{(k,l)}}$ is the i-th normal of a local separation line $\mathbf{BL}_{k,l}$ between the segments S_k and S_l, and $\mathbf{p}_{i_{(k,l)}}$ is point on this line. $\mathbf{m}_{j_k}$ is the j-th normal of the local surface plane $\mathbf{SP}_k$ of segment S_k, and $\mathbf{q}_{j_k}$ is a point on this plane. The QEF is solved in matrix form by least squares adjustment[6]:

$$\hat{\mathbf{X}} = \underset{\mathbf{X}}{\operatorname{argmin}} \left\{ (\mathbf{AX} - \mathbf{b})^T (\mathbf{AX} - \mathbf{b}) \right\} = \underset{\mathbf{X}}{\operatorname{argmin}} \{\mathbf{AX} - \mathbf{b}\}, \quad (4.13)$$

where $\mathbf{A}$ is the model matix, and $\mathbf{b}$ is the vector of observations. The first column of $\mathbf{A}$ contains the x-components of $\mathbf{n}_{i_{(k,l)}}$ and of $\mathbf{n}_{j_k}$, the second column contains the correponding y-components, and the $2+k$-th column contains the z-components of $\mathbf{n}_{j_k}$. Each i and j indicates a separate row of $\mathbf{A}$. The remaining entries in $\mathbf{A}$ are padded with zeros [Zhou and Neumann, 2010]. The estimated vector $\hat{\mathbf{X}}$ has one z-coordinate $\hat{z}_k$ for each segment S_k. $\mathbf{b}$ contains the dot products $(\mathbf{n}_{i_{(k,l)}}^T \cdot \mathbf{p}_{i_{(k,l)}})$ and $(\mathbf{m}_{j_k}^T \cdot \mathbf{q}_{j_k})$.

4.5.3 Grid Data Generation

Local surface planes (SP)

For each vertex of the 2D grid, a local 3D surface plane $\mathbf{SP} = [\mathbf{m}^T, \mathbf{q}^T]^T$ is determined, where $\mathbf{m} = [m_x, m_y, m_z]^T$ is the plane's normal, and $\mathbf{q} = [q_x, q_y, q_z]^T$ is a point on the plane. $\mathbf{q}_{xy} = [q_x, q_y]^T$ is equal to the grid vertex. $\mathbf{S}$ is associated with a segment label $l_{\mathbf{SP}}$ according to the topology between $\mathbf{q}_{xy}$ and the segment boundary polygons[7]: Assuming that all segment boundary polygons are entirely inside the building footprint and that they are clearly separated from each other (not overlapping), there are three possible relations between $\mathbf{q}_{xy}$ and the segment boundary polygons:

- If the $\mathbf{q}_{xy}$ is inside the boundary polygon of segment S_k, $l_{\mathbf{SP}} = k$.
- If the $\mathbf{q}_{xy}$ is inside the building footprint, but outside all segment boundary polygons, the $l_{\mathbf{SP}}$ is chosen according to the closest segment boundary polygon.
- If the $\mathbf{q}_{xy}$ is outside the building footprint, $l_{\mathbf{SP}}$ is set to ground: $l_{\mathbf{SP}} = 0$.

$\mathbf{m}$ is determined as the weighted average of the K nearest[8] triangles' normals, considering only those triangles which are segmented to S_k. Triangles with larger area get higher weights. p_z of is chosen as

[6] Zhou and Neumann [2010] solve equation 4.13 using a QR-decomposition, improving computational efficiency due to using floats as data types while reaching the accuracy of doubles [Ju et al., 2002].

[7] as described in 4.4

[8] Distances are determined according to the triangle centers' 2D distances from the grid point. The smaller K, the stronger are the variances of adjacent m.

the mean of the chosen triangle centers' z-coordinates.

Local boundary lines (BL)

Each grid cell has one **SP** computed at each of its four cell vertices. For each pair of differently labeled $\mathbf{SP}_k$, $\mathbf{SP}_l$, a 2D local boundary line $\mathbf{BL}_{k,l} = [\mathbf{n}^T, \mathbf{p}]^T$ is determined, where $\mathbf{n} = [n_x, n_y]^T$ is the line's normal, and $\mathbf{p} = [p_x, p_y]^T$ is a point on that line.

Valid data zone: Each $\mathbf{BL}_{k,l}$ is estimated from the corresponding segments' (S_k and S_l) LiDAR points whose projection to the XY-plane is within the a buffered[9] zone around the considered cell (data zone). If there are no such points in data zone, the K nearest corresponding points are chosen. The buffer guarantees an overlap to adjacent data zones, such that $\mathbf{BL}_{k,l}$ can also be found if it is alined with the cell border. If the boundary line intersects with the grid cell's border edges, a point **p** is determined as this intersection. Otherwise, the boundary line is discarded.

Zhou and Neumann [2010] determine one **BL** for each grid edge which connects differently labeled **SP**, i.e. their data zone is twice the size in vertical extent as in horizontal extent, or viceversa. The underlying assumption that the **BL** intersects with this grid edge is not applicable in the proposed workflow, for two reasons: Firstly, the LiDAR points from which **BL** is computed may be far from the data zone, if **SP** is in between segment boundary polygons. Secondly, in case of a small grid cell size, the resulting **BL** may have no intersection with the grid cell's border edges. In contrast to Zhou and Neumann [2010], the proposed workflow produces **BL** also for those differently labeled **SP** which are opposing diagonally within one cell.

Line estimator: The optimal boundary line $n_x x + n_x y + c = 0$ separates two segments S_k and S_l such that the distances of points of S_l to the line are negative, and the distances of points of S_k to the line are positive.

$$\mathbf{L}_{BL}^T \cdot \mathbf{P}_k = 1, P_k \in S_k \tag{4.14}$$

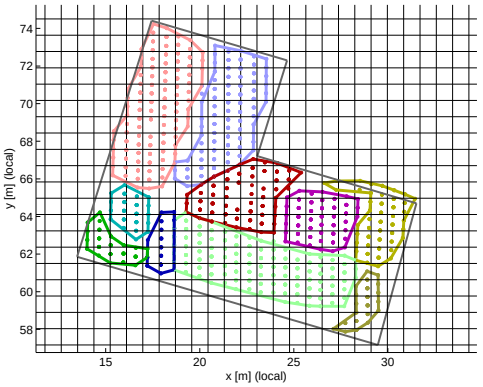
$$\mathbf{L}_{BL}^T \cdot \mathbf{P}_l = -1, P_l \in S_l \tag{4.15}$$

where $\mathbf{L}_{BL} = \begin{bmatrix} n_x & n_y & c \end{bmatrix}^T$ are the line parameters consisting of normal **n** and the distance c to the origin, and $\mathbf{P}_k = \begin{bmatrix} x_k & y_k & 1 \end{bmatrix}^T$ and $\mathbf{P}_l = \begin{bmatrix} x_l & y_l & 1 \end{bmatrix}^T$ are LiDAR points assigned to segment S_k and S_l during the segmentation. With $k = (1, ..., K)$ and $l = (1, ..., L)$, $\mathbf{L}_{BL}$ is found by least squares minimization of

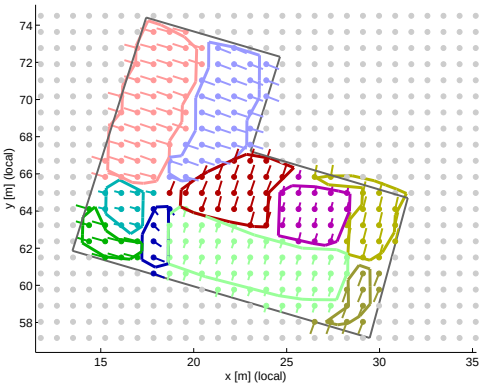
$$\hat{\mathbf{L}}_{BL} = \underset{\mathbf{L}_{BL}}{\operatorname{argmin}} \left\{ \sum_k^K \left((\mathbf{L}_{BL}^T \cdot \mathbf{P}_k) - 1 \right) + \sum_l^L \left((\mathbf{L}_{BL}^T \cdot \mathbf{P}_l) + 1 \right) \right\}. \tag{4.16}$$

For each intersection of the building boundary (footprint) with the 2D-grid, a corresponding **BL** is created. Those line segments of the footprint which have no intersection with the grid edge are extended to intersect with the grid.

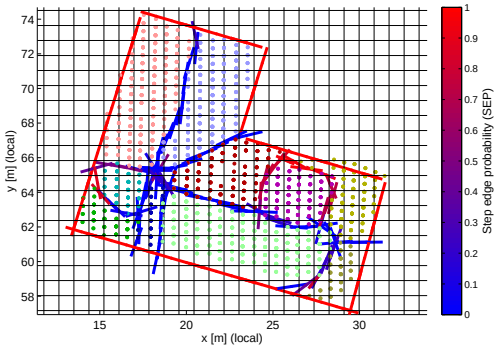
[9] The buffer is chosen such that the data zones overlap by in average 3 LiDAR points. This is computed from the average point cloud density.



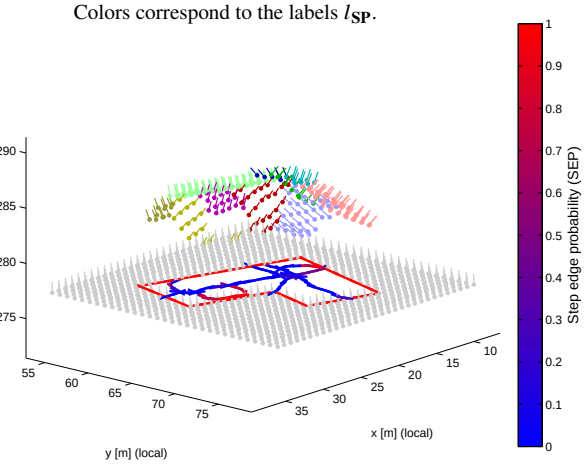
(a) 2D-grid over segmented building LiDAR points (different colors of the points: different segments), the corresponding segment boundaries, and the building footprint (grey line)



(b) Points **q** and normal vectors **m** of the surface planes **SP**, plotted over the segment boundaries. Colors correspond to the labels $l_{\mathbf{SP}}$.



(c) Boundary lines **BL** plotted over the segmented LiDAR points (different colors of the points: different segments). The colors of the **BL** indicate the step edge probabilities.



(d) Combined grid data: Boundary lines **BL** and surface planes $l_{\mathbf{SP}}$, in side view.

FIGURE 4.11: Illustration of the DC approach in a 2D case

4.5.4 QEF Modification and Weighting

Matrix modification for intersection edges

Adjacent segments are not necessarily separated by a step edge, but also intersection edges are possible. In contrast to Zhou and Neumann [2010], who compute one z-coordinate $\hat{z}_k$ for each height layer of $\hat{\mathbf{X}}$, the proposed workflow has to adapt the number of z-coordinates of each hyperpoint depending on intersection and step edges. This is achieved by modifying the matrix $\mathbf{A}$: With H_v describing a group of segments $\{S_k | k \in H_v\}$ which are connected by intersection edges ($SEP < 0.5$), all z-components of the normals $\{\mathbf{m}_{\mathbf{i_k}} | k \in H_v\}$ are merged into one column of $\mathbf{A}$ (see equation 4.19).

The two cases "intersection edge" and "step edge" are distinguished by estimating the step edge probability SEP for each BL. From the LiDAR points which were used for computing the BL, the two closest LiDAR points ($\mathbf{P}_k \bar{\mathbf{P}}_l$) are determined as reference points $\mathbf{P}_{ref,k}$ and $\mathbf{P}_{ref,l}$. The minimum $d_{z,min}$ of three distance measures is computed:

$d_{z,1}$: The height difference between $\mathbf{P}_{ref,k}$ and $\mathbf{P}_{ref,l}$
$d_{z,2}$: The vertical distance of $\mathbf{P}_{ref,k}$ to the plane fitted to S_l.
$d_{z,3}$: The vertical distance of $\mathbf{P}_{ref,l}$ to the plane fitted to S_k.

Assuming a normally distributed error, SEP is computed from the probability function:

$$SEP = \exp\left\{-\frac{d_{z,min}^2}{2 \cdot (T_{step})^2}\right\}, \tag{4.17}$$

where T_{step} is a fixed step edge threshold.

Weighting of grid data

BL-weighting: Minimizing the 3D quadratic error term $E_{3D}(X)$ implies that the solution is approaching a minimum least squares distance to the local surface planes SP. This is only desirable in case of an intersection edge. In case of a step edge, the influence of the **SP** normals **m** on the horizontal position of the solution should be minimal, whereas in case of an intersection edge, the influence of the **BL** normals **n** should be minimal. Zhou and Neumann [2010] only face the second situation, since their BL represent step edges alltogether.

Since **SP** cannot be omitted - they are the only data from which the hyperpoint's z-coordinates can be computed from - the weightings of **BL** normals **n** with respect to the **SP** normals **m** can be increased. The intention is to to balance between **BL** and **SP** depending on the prevailing situation (step edge or intersection edge). Each $\mathbf{m}_{i_{(k,l)}}$ is weighted with $w_{(k,l)}$, which is computed from the maximum $SEP_{i_{(k,l)}}$ in this QEF cell.

$$w_{(k,l)} = F_w\left(SEP_{max} | SEP_{max} = \max_i(SEP_{i_{(k,l)}})\right), \tag{4.18}$$

where F_w is a weighting function. F_w is determined from fitting an exponential function $F_w([0, 0.5, 1]) = [0, 1, w_{max}]$, with the intention to create a close-to-zero weighting for a low SEP, a maximum weighting w_{max} for a high SEP, and a weighting of 1 (equal weighting) in case of a 50 % probability.

SP-scaling: Within one QEF, there are N_k local surface planes $\mathbf{SP}_k$ for segment S_k, and usually $N_k \neq N_l$. As an example, there might be three $\mathbf{SP}_k$ and one $\mathbf{SP}_l$ in an intersection edge situation (only one z-coordinate is estimated for both segments). If all $\mathbf{SP}_{i_k}$ and $\mathbf{SP}_{i_l}$ are weighted equally, the QEF solution does not result in the intersection of segments S_k and S_l, but the intersection of all individual **SP**. If the normals $\mathbf{m}_{i_k}$ are very similar to each other, i.e. close to planar, the solution may be far from the actual intersection line of S_k and S_l.[10] This situation is avoided by dividing each $\mathbf{m}_{i_k}$ by N_k.

[10] In certain cases the **BL** in this QEF have no restrictive effect, e.g. they have low weights, or their normal directions $\mathbf{n}_{i_{k,l}}$ are similar to those of $\mathbf{m}_{i_k}$.

Weighted and modified QEF: The total weighted and modified QEF yields:

$$\hat{\mathbf{X}}_w = \underset{\mathbf{X}}{\operatorname{argmin}} \left\{ \sum_{k=1}^{M-1} \sum_{l=2, l\neq k}^{M} \sum_{i_{(k,l)}}^{I_{(k,l)}} \left[w_{(k,l)} \cdot \mathbf{n}_{i_{(k,l)}} \left(\mathbf{p}_{i_{(k,l)}} - \begin{bmatrix} x \\ y \end{bmatrix} \right) \right]^2 + \sum_{H_v} \sum_{k \in H_v}^{M} \sum_{j_k}^{N_k} \left[\frac{\mathbf{m}_{j_k}}{N_k} \left(\mathbf{q}_{j_k} - \begin{bmatrix} x \\ y \\ z_v \end{bmatrix} \right) \right]^2 \right\}, \quad (4.19)$$

and

$$\hat{\mathbf{X}}_w = \underset{\mathbf{X}}{\operatorname{argmin}} \left\{ (\mathbf{A}_w \mathbf{X} - \mathbf{b}_w)^T (\mathbf{A}_w \mathbf{X} - \mathbf{b}_w) \right\} = \underset{\mathbf{X}}{\operatorname{argmin}} \left\{ \mathbf{A}_w \mathbf{X} - \mathbf{b}_w \right\}, \quad (4.20)$$

where $\mathbf{A}_w$ and $\mathbf{b}_w$ are the weighted matrix and vector equivalents of equation 4.13 (see annex A for the detailed matrix equation). The resulting hyperpoint $\hat{\mathbf{X}}_w = \begin{bmatrix} \hat{x} & \hat{y} & \hat{z}_v & \hat{z}_{v+1} & \dots & \hat{z}_V \end{bmatrix}^T$ has $2+V$ entries: $\hat{\mathbf{X}}_w$ Each $\mathbf{V} = [\hat{\mathbf{X}}_w(1), \hat{\mathbf{X}}_w(2), \hat{\mathbf{X}}_w(2+v)]$ is one vertex of the hyperpoint. All labels $l_{\mathbf{SP}}$ of those **SP** which are in one $2+V$-th column of A, i.e. which are leading to one hyperpoint vertex, are stored in a label vector $\mathbf{l}_v$.

4.5.5 Quad Tree Collapsing

If one hyperpoint is computed for each grid cell, the 3D building model will be represented in unnecessary detail. The goal is to describe the building model with a minimum number of vertices, while fitting it to the data in an optimal manner. Therefore, the number of hyperpoints is iteratively reduced by collapsing the grid using a quadtree: After QEFs are precomputed for all grid cells, four adjacent cells are collapsed to a quadtree cell, if the QEF minimization's residual R_{QEF} is smaller than a fixed threshold R. The collapsing procedure continues in the upper quadtree cell levels until no more quadtree cells can be collapsed.

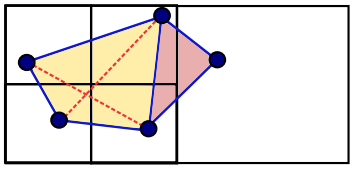
$$R_{QEF} = \mathbf{F}_w^T \mathbf{F}_w = \left(\frac{1}{\mathbf{W}} (\mathbf{A}_w \hat{\mathbf{X}}_w - \mathbf{b}_w) \right)^T \left(\frac{1}{\mathbf{W}} (\mathbf{A}_w \hat{\mathbf{X}}_w - \mathbf{b}_w) \right) < R, \quad (4.21)$$

where $\mathbf{W} = \left[w_{(k,l)}, w_{(k,l+1)}, \dots, w_{(k+1,l)}, \dots, w_{(M,M-1)}, \dots, \frac{1}{N_k}, \frac{1}{N_{k+1}}, \dots, \frac{1}{N_M} \right]^T$ is the vector containing the weightings of each QEF line.

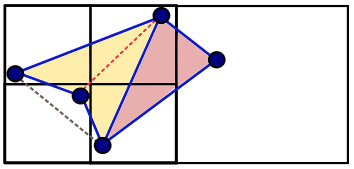
4.5.6 3D Polygon Creation

For each pair of adjacent quadtree cells, one or more 3D edges of the final building model are computed. Each vertex pair of the corresponding hyperpoint pair $\mathbf{V}_u$ and $\mathbf{V}_v$ are connected to a 3D edge, if they share at least one equal label in their associated label vectors $\mathbf{l}_{V_u}$ and $\mathbf{l}_{V_v}$. It is guaranteed that at least one 3D edge will be found. For each SP which is a part of the quadtree cell, the hyperpoint has at least one corresponding label $l \in \{\mathbf{l}_1, \dots, \mathbf{l}_V\}$, and there are at least two SP which are shared by adjacent quadtree cells. This implies, that a closed, hole-free building polygon can be created from the hyperpoint quadtree.

The 3D edges have to be connected to planar polygon faces. These connections result in two possible shapes: A triangle of vertices, and a quad of vertices (Fig. 4.12)



(a) Two possibilities for quad division (red dotted lines).



(b) One possibility for quad division (red dotted line); grey dotted line: no possibility for quad division

FIGURE 4.12: Quads (yellow areas) and triangles (red areas) resulting from connecting adjacent hyperpoints' vertices (dark blue points), in top view

As the polyhedral building model shall only consist of planar faces, each quad of vertices has to be subdivided into two triangles. There are two possibilities for quad division (Fig. 4.12a). If adjacent hyperpoints form a concave quad, there is only one possible division to partition the quad into two triangles (Fig. 4.12a). In case of two possibilities, the quad is divided to fit the LiDAR data optimally: For each division possibility, there are two quad vertices which are only part of one of the resulting triangles. For each of these two vertices, a plane is fitted to the K nearest LiDAR points. The total angular deviation of these planes to their corresponding triangles' planes indicates how well this division fits the data.

For evaluation and plotting, each triangle is associated with a segment label l_d. l_d is chosen among the triangle's vertices' labels $\mathbf{l}_{V_u}$, $\mathbf{l}_{V_v}$, and $\mathbf{l}_{V_w}$ as the most dominant one. If there is no dominant label, l_d is determined from the K nearest[11] LiDAR points. If the number K is not enough to determine a clearly dominant label, K is increased iteratively until such a dominant label can be determined.

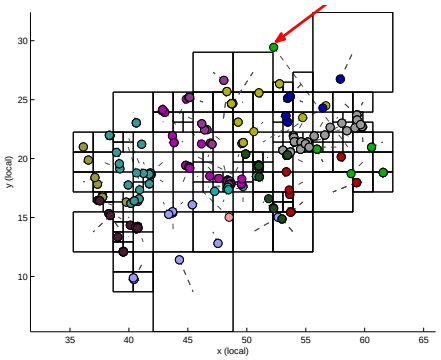
Vertical walls are created at each triangle edge which is not shared by another edge (single edge). If there is another single edge connecting the same two hyperpoints, a wall segment is created between them. Else, a vertical wall is created to ground.

4.5.7 Solution Constraints

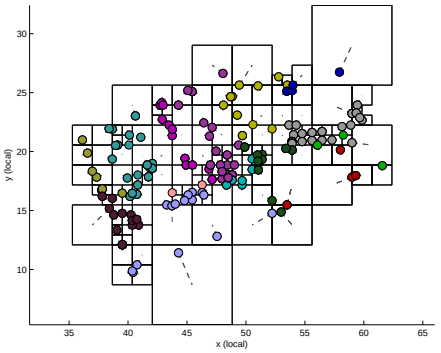
In certain cases, the **SP** within a quadtree cell are nearly coplanar. If in addition the **BL** of this quadtree cell have small weights, their restrictive effect on the solution is very small. Then, the QEF minimization may result in a hyperpoint which lies far outside the quad boundary (Fig. **??** a), resulting in a distorted building model (Fig. **??** c). To avoid this problem, the solution's horizontal position is restricted to the intersection point of horizontal line from the quad region's center to the far solution with the quad boundary[12] The z-coordinates are determined by intersecting the vertical line on the restricted solution with the averaged plane of all corresponding surface samples. The residual is determined as in 4.21, with $\mathbf{F}_w$ being computed from the distances of the respective hyperpoint vertices to the corresponding surface samples' planes.

[11] Distance to the triangle is determined according the distance to one of the triangle edges in the X-Y-plane. The triangle's inlier points are given preference when choosing the K nearest.

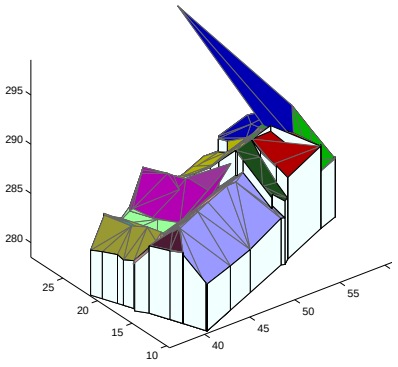
[12] A similar approach is found in Fiocco et al. [2005], who apply a DC procedure in 3D: Fiocco et al. [2005] solve this problem by projecing their QEF solution to the nearest cell face of their voxel raster. In 2.5D-case, this might lead to abrupt height discontinuities, if a solution vertex from a higher segment falls in the midst of a lower region, or viceversa.



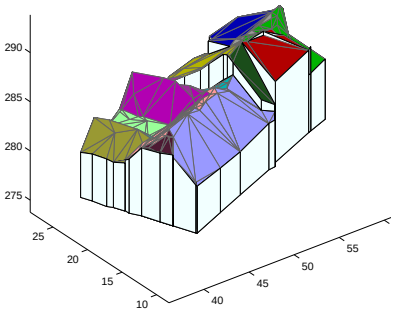
(a) Hyperpoints without constraints. The dotted lines indicate from which quadtree cell the hyperpoints are computed. The red arrow points to a hyperpoint (which is labeled with the green and blue segment), which is far outside its corresponding quadtree cell.



(b) Hyperpoints with constraints. All hyperpoints are enforced to lie within or on the border of their corresponding quadtree cells' borders.



(c) 3D Model without constraints, after the hyperpoints are connected. Obvious deformation due to the wrong hyperpoint.



(d) 3D Model with constraints. No strong deformation visible.

FIGURE 4.13: Effect of solution constraints on the positions of hyperpoints and the resulting 3D model

5 Tests and Evaluation

5.1 Tests Data and Ground Truth

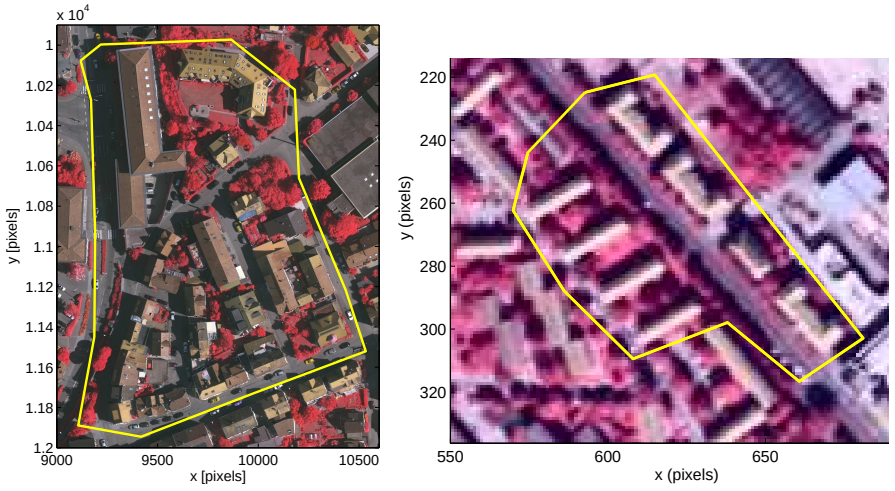
5.1.1 Test Data

Two scenes of different data characteristics and different building complexities are chosen for testing and evaluation. The test scene of higher LiDAR point density, higher number of buildings[1] and higher building complexity is area 1 of the ISPRS benchmark project's test data set in Vaihingen, Germany. The test scene of lower LiDAR point density, lower number of buildings and lower building complexity is a scene in the center of Munich, Germany. Table 5.1 gives an overview of the data characteristics of the LiDAR point clouds and test scenes. In both cases, ground truth was extracted manually from the corresponding ortho-image or from open source online services[2].

Table 5.1: LiDAR input data of Vaihingen and Munich

	Vaihingen	**Munich**
Point denisty	∅ 3.452 points / m^2	2.315 points / m^2
Vertical accuracy	n.a.	n.a.
Buildings	21 buildings, 182 roof segments, complex roof shapes	8 buildings, 21 roof segments, simpler roof shapes
Orthoimage	0.09 m pixel size	2 m pixel size

Fig. 5.1 shows the two test areas in the corresponding orthoimages.



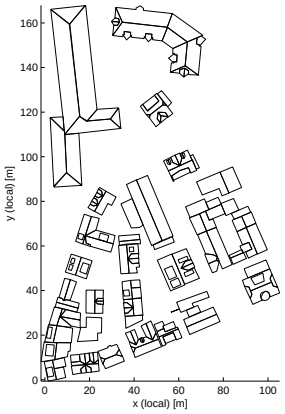
(a) Vaihingen test scene

(b) Munich test scene

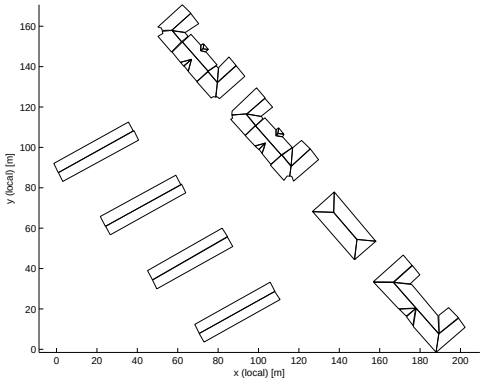
Figure 5.1: Test scenes shown in the corresponding orthoimages

[1] Where each building is defined as a block of connected roof parts which is entirely surrounded by ground.

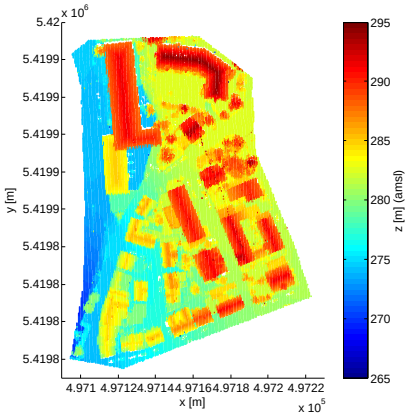
[2] For the Munich data set, ground truth was extracted manually and downloaded from Google Earth, due to higher image resolution than in the available orthoimage.



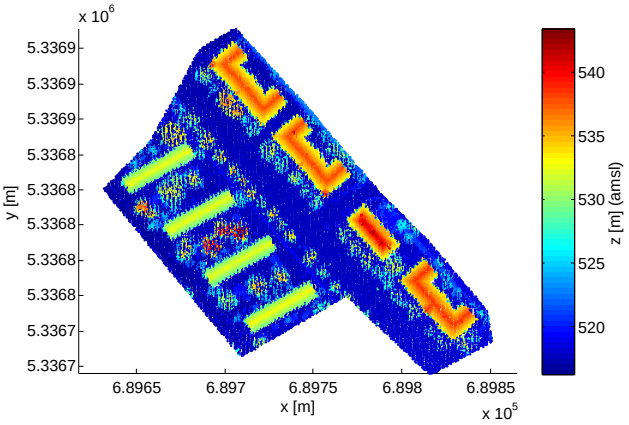
(a) Manually extracted ground truth of Vaihinen test scene



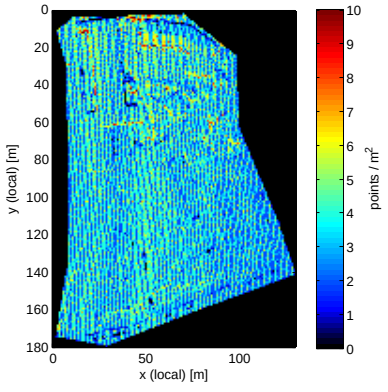
(b) Manually extracted ground truth of Munich test scene



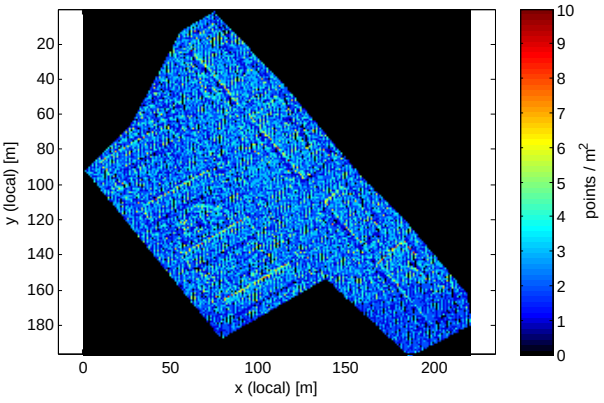
(c) LiDAR point cloud of Vaihingen test scene



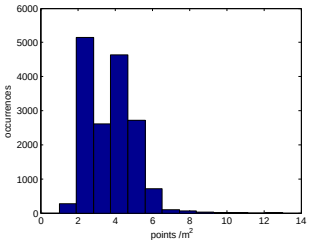
(d) LiDAR point cloud of Munich test scene



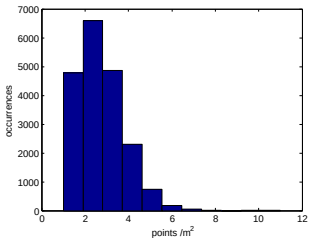
(e) Map of point densities (pixel size 1 m), Vaihingen test scene



(f) Map of point densities (pixel size 1 m), Munich test scene



(g) Histogram of point densities, Vaihingen test scene



(h) Histogram of point densities, Munich test scene

FIGURE 5.2: Ground truths and LiDAR point characteristics of the two test scenes

Fig.5.2 show the reference polygons (ground truth), and an analysis of the LiDAR point density distributions of the test scenes. For each test scene, the data of a single strip is used for reconstruction, in order to avoid abrupt point density changes within one building point cloud. Fig. 5.2 e) and f) show maps of point densities, which are computed by counting the number of pixels falling in the pixel. From Fig. 5.2 e) and f) and their corresponding histograms in Fig. 5.2 g) and h), it is clear that the Vaihingen test scene has a higher point density, but also higher density variations.

5.2 Test

5.2.1 Parameter setting

Through experiments and visual interpretation, the optimal input parameters for segmentation and reconstruction were determined as follows (Table 5.2).

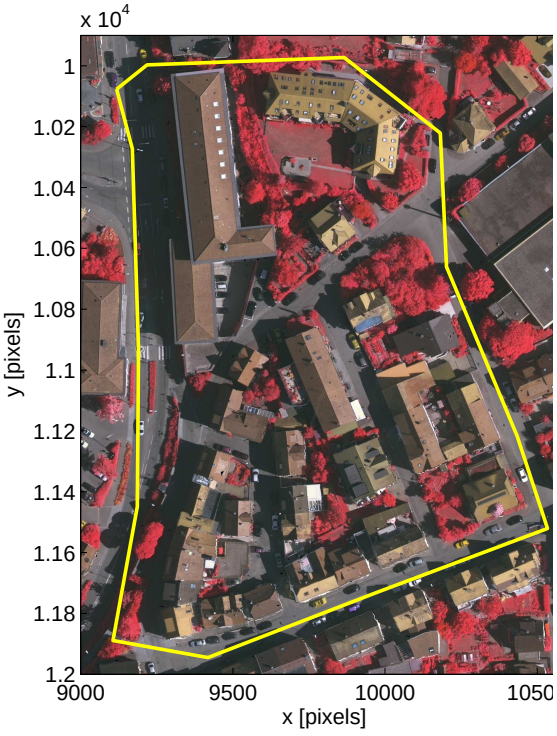
TABLE 5.2: Input parameters

Parameter	Vaihingen test scene	Munich test scene
h for presegmentation	2.5 m	2.5 m
footprint buffer for presegmentation	0.8 m	0.8 m
NDVI threshold for presegmentation	0.3	0.3
segmentation threshold T_θ	12.5°	20°
$a_{mad,LUF}$ for segmentation threshold $T_{\Delta LUF}$	1.5	2
$a_{mad,RMSE}$ for segmentation threshold T_{RMSE}	2	2
grid cell size C	2.5 points / grid cell	3 points / grid cell
residual threshold R	0.8 m	1.2 m
step edge theshold T_{step}	0.2 m	0.3 m

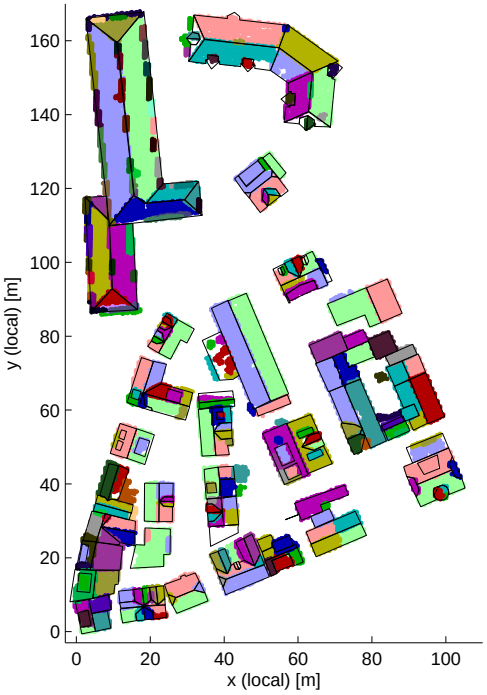
5.2.2 Segmentation results

The following figures show the results for segmentation, plotted over the ground truth. The colored maps in subfigures b) of Fig. 5.4 and 5.3 show the segmented LiDAR roof points, where each color represents a different segment within one building. LiDAR points at segment borders may be segmented to more than one segment, even though this cannot be shown in the plots. Subfigures c) show the boundaries of detected (true positive, TP_r), missed (false negatives, FN) reference segments, correct (true positives, TP_e), and wrong (false positives, FP) estimated segments, as determined from the evaluation process in section 5.4. Subfigures d) show undersegmented and oversegmented regions.

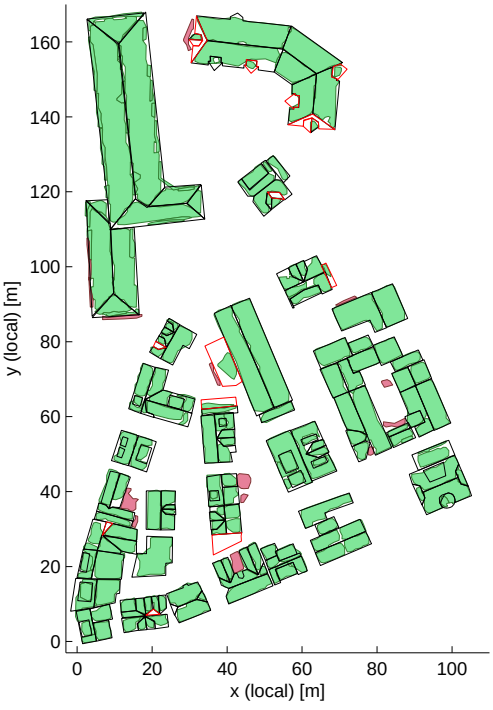
Segmentation of Vaihingen city scene



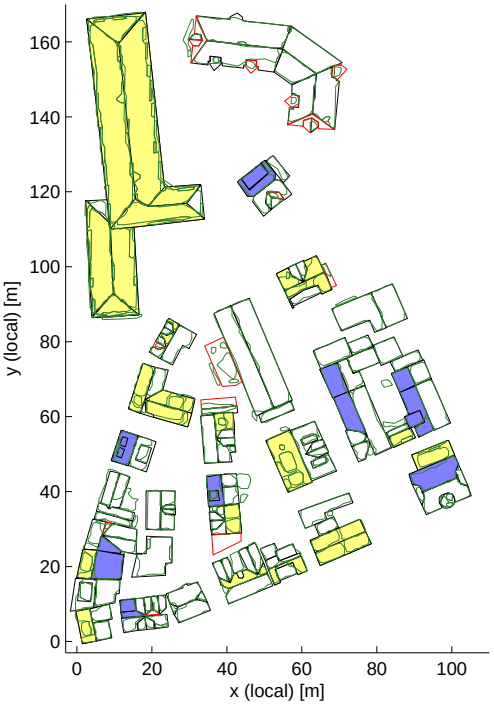
(a) Orthoimage of the scene



(b) Segmented LiDAR points of the scene; different colors indicate different roof planes (colors are chosen according to the sequence of detected segments within each building)



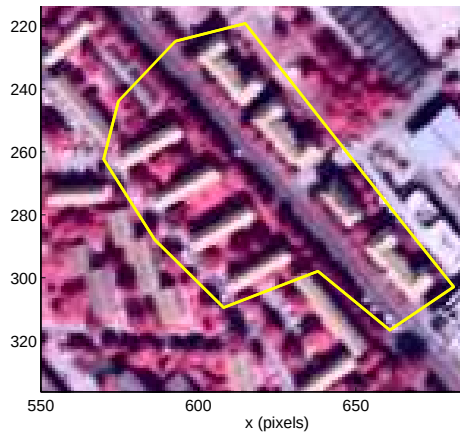
(c) Green areas: True positive estimated segments TP_e; red areas: False positive estimated segments FP; black polygons: true positive (detected) reference segments TP_r; red polygons: false negative (missed) reference segments FN



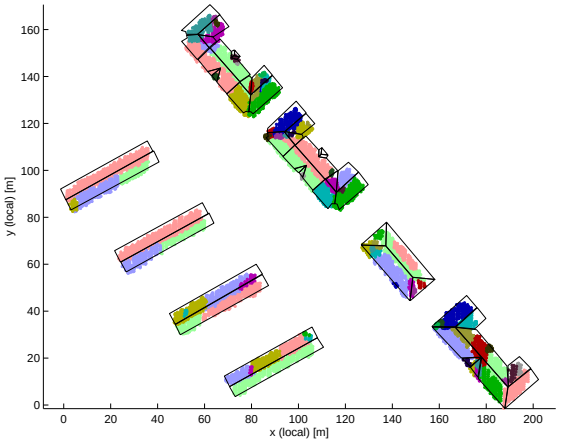
(d) Yellow areas: oversegmented reference segments; blue areas: undersegmented reference segments; black polygons: true positive (detected) reference segments TP_r; red polygons: false negative (missed) reference segments FN; dark green polygons: estimated segments.

FIGURE 5.3: Segmentation results of the Vaihingen test scene

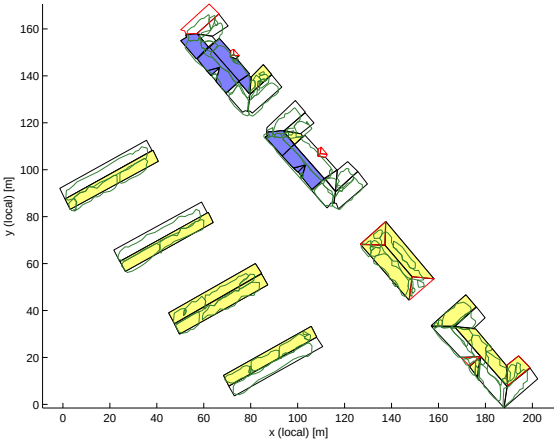
Segmentation of Munich city scene



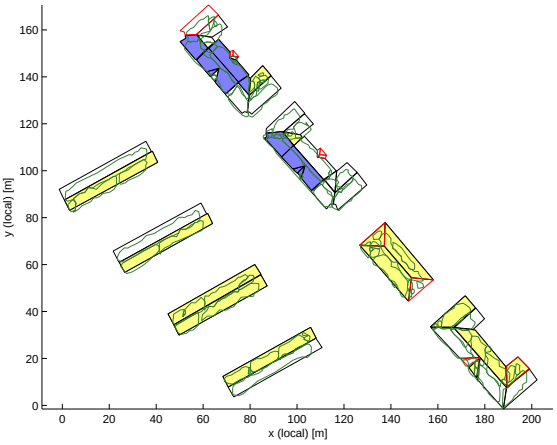
(a) Orthoimage of the scene



(b) Segmented LiDAR points of the scene; different colors indicate different roof planes (colors are chosen according to the sequence of detected segments within each building)



(c) Green areas: True positive estimated segments TP_e; red areas: False positive estimated segments FP; black polygons: true positive (detected) reference segments TP_r; red polygons: false negative (missed) reference segments FN



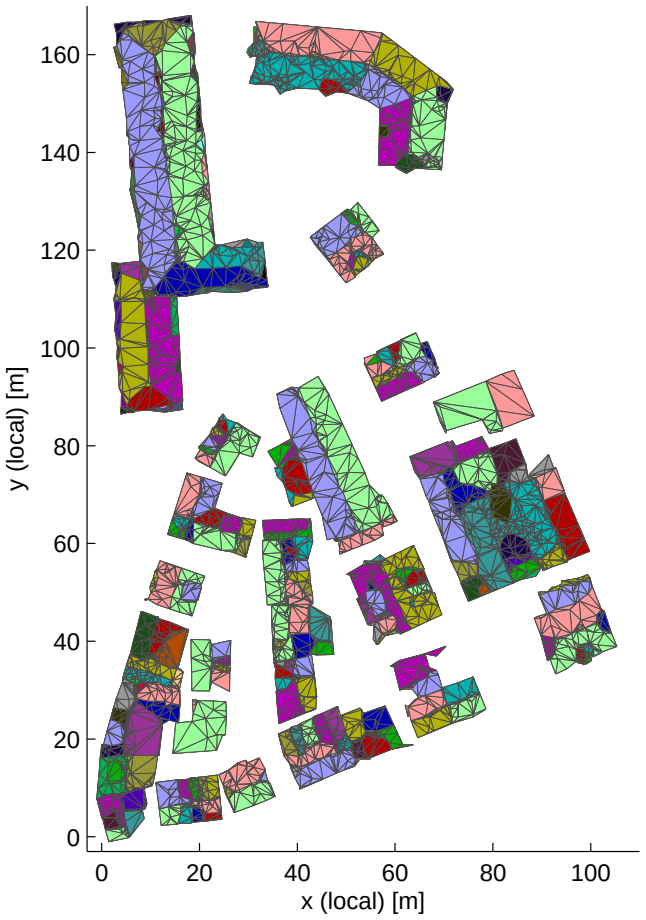
(d) Yellow areas: oversegmented reference segments; blue areas: undersegmented reference segments; black polygons: true positive (detected) reference segments TP_r; red polygons: false negative (missed) reference segments FN; dark green polygons: estimated segments.

FIGURE 5.4: Segmentation results of the Munich test scene

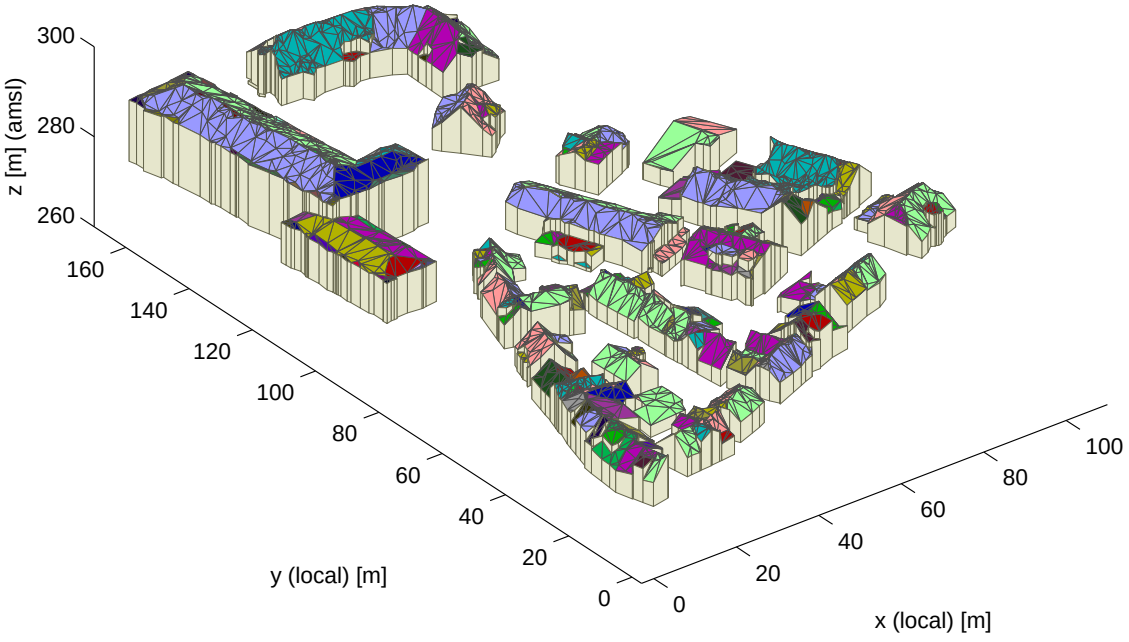
5.2.3 Reconstruction results

The following figures show the results for reconstruction of both test scenes in top view and side view.

3D model of Vaihingen city scene



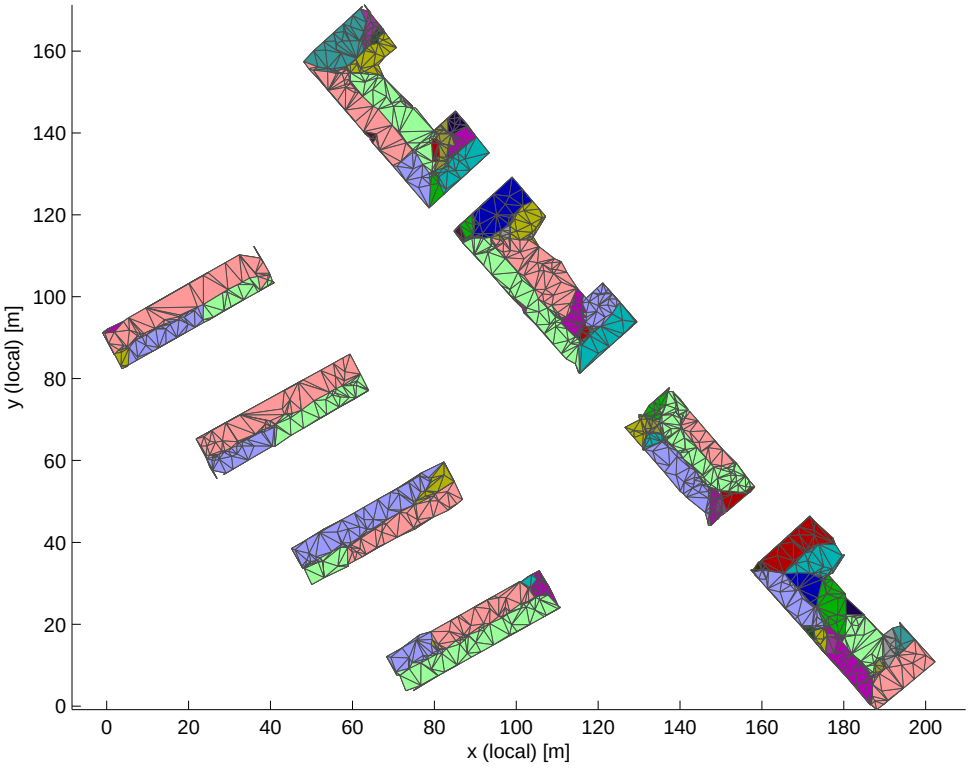
(a) Top view of the reconstructed scene, different colors show differently labeled triangles, indicating different segments (colors are chosen according to the sequence of detected segments within each building)



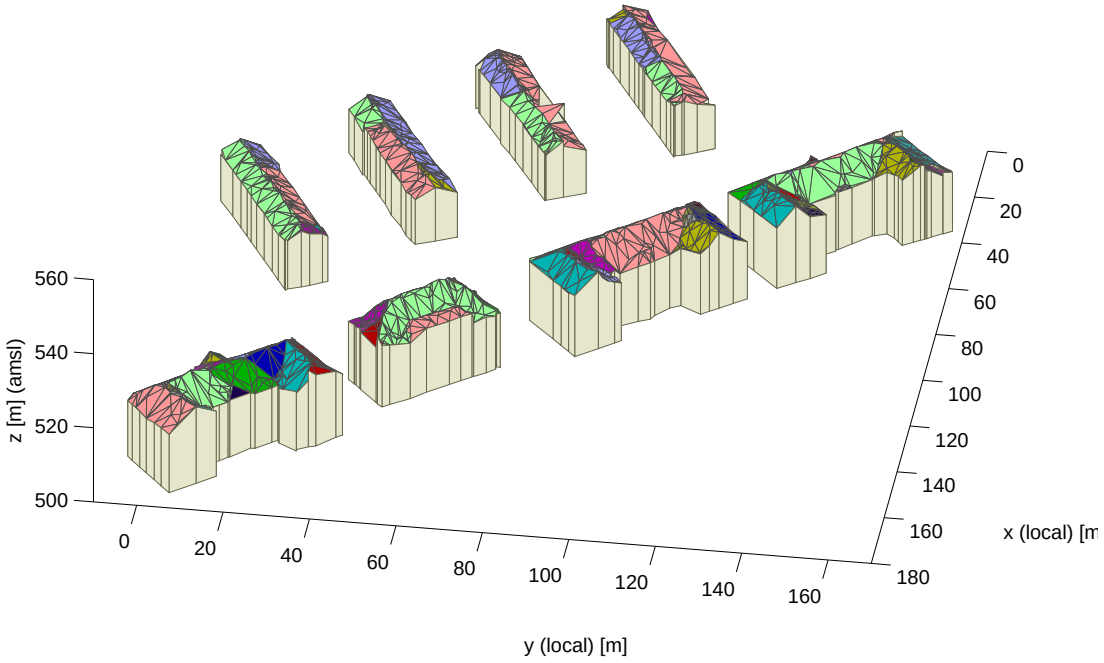
(b) Side view of the reconstructed scene, wall segments (grey areas) are connecting the outer roof boundaries to ground or to other roof segments

FIGURE 5.5: Reconstruction results of the Vaihingen test scene

3D model of Munich city scene



(a) Top view of the reconstructed scene, different colors show differently labeled triangles, indicating different segments (colors are chosen according to the sequence of detected segments within each building)



(b) Side view of the reconstructed scene, wall segments (grey areas) are connecting the outer roof boundaries to ground or to other roof segments

FIGURE 5.6: Reconstruction results of the Munich test scene

5.3 Evaluation Method

Both segmentation and reconstruction results are evaluated for different input parameters, using the object-based evaluation method of the ISPRS Benchmark Project [Rottensteiner et al., 2012a, Rutzinger et al., 2009, Rottensteiner, 2012]. Basis for the evaluation of the segmentation are the 2D boundary polygons as described in section 4.4. Basis for the evaluation of the reconstruction are the 2D-projections of 3D boundary polygons of the roof model segments, which are extracted by tracing the boundary edges of adjacent, equally labeled roof triangles. Eight evaluation parameters are calculated:

1. Completeness $C_m = \frac{TP_r}{TP_r + FN}$, where TP_r (True Positives) and FN (False Negatives) are the numbers of reference polygons whose area is at least $2.5m^2$ and which are overlapping by at least (TP_r) or less than (FN) 50 % with estimated segment polygons.
2. $C_{m,10}$ is computed analogously for segment areas $\geq 10m^2$.
3. Correctness $C_r = \frac{TP_e}{TP_e + FP}$, where TP_d[3] (True Positives) and FP (False Positives) are the numbers of all estimated segment polygons whose area is at least $2.5m^2$ and which are overlapping by at least (TP_d) or less than (FP) 50 % with reference polygons.
4. $C_{r,10}$ is computed analogously for segment areas $\geq 10m^2$.
5. Horizontal (planimetric) accuracy $RMSE_{xy}$: Root mean square error of the minimum 2D distances of the estimated segment's polygon points to their reference, while only distances larger than a threshold $d_{thr} = 3m$ are considered,
6. Number of oversegmented references N_O, counting all reference segments corresponding to more than one TP_d.
7. Number of undersegmenting estimated polygons N_U, counting all estimated polygons corresponding to more than one TP_r.
8. Number of references which are both under- and oversegmented $N_{O\&U}$.

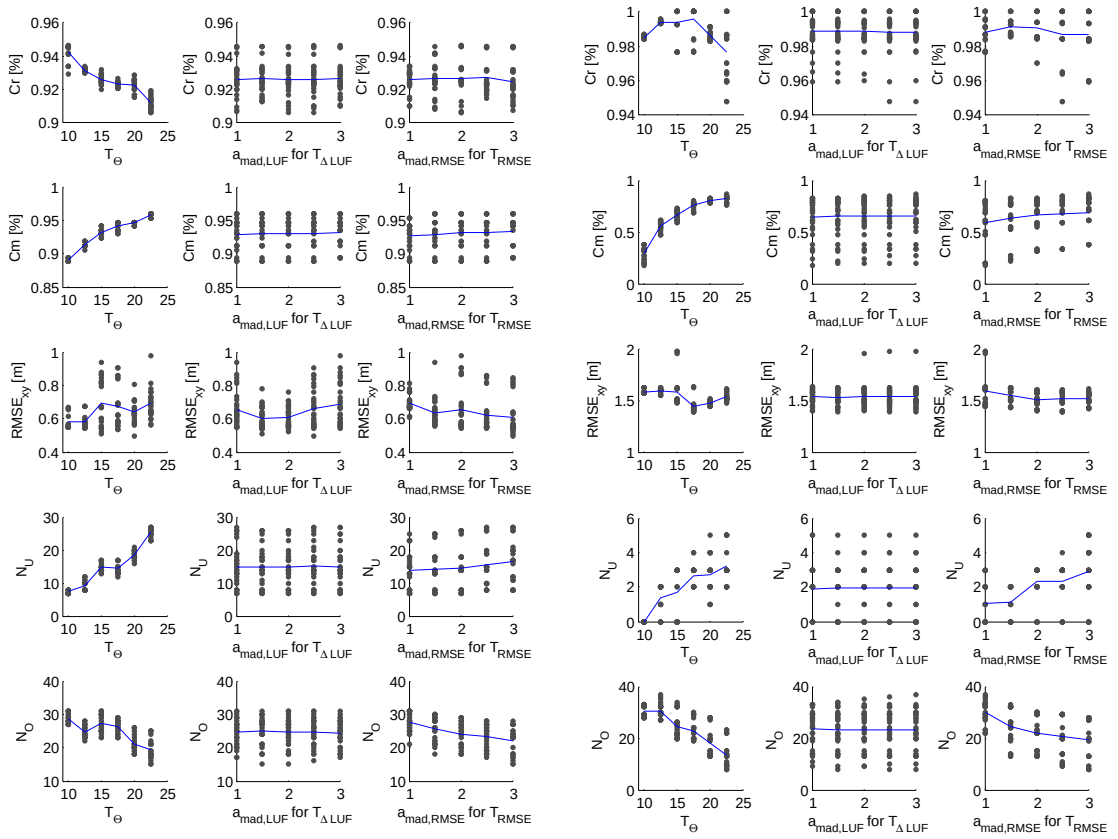
5.4 Evaluation of Segmentation

The impact of the most important input parameters on the evaluation parameters is analysed in detail for both data sets. The most important input parameters for segmentation are the relative fixed threshold $T_{\Delta\Phi}$, the adaptive thresholds T_{RMSE} and $T_{\Delta LUF}$. The absolute fixed thresholds $T_{elevation}$ and $T_{L_{max}}$ are not analysed in detail, as they apply only to few triangles.

As shown Fig. 5.7, the $T_{\Delta\Phi}$ has the strongest influence on correctness C_r, completeness C_m, oversegmentation N_O and undersegmentation N_U. Completeness and undersegmentation increase strongly with increasing $T_{\Delta\Phi}$, while correctness, oversegmentation and $RMSE_{xy}$ decrease. This effect is stronger in the Munich test scene than in the Vaihingen test scene.

[3] The distinction between TP_r and TP_d is described in Rottensteiner et al. [2005].

Discrete parameter testing:



(a) Vaihingen test scene

(b) Munich test scene

FIGURE 5.7: Influences of the input parameters for segmentation on the evaluation parameters. The blue lines connect the average evaluation parameters for the respective input parameters.

Similarly to $T_{\Delta\Phi}$, a higher T_{RMSE} can delimit oversegmentation. But in contrast to $T_{\Delta\Phi}$, a higher T_{RMSE} does not result in a strong increase in undersegmentation. These effects of T_{RMSE} and $T_{\Delta LUF}$ are stronger in the Vaihingen test scene. In the Munich test scene, the effects are T_{RMSE} and $T_{\Delta LUF}$ are more similar to the one of $T_{\Delta\Phi}$, but in alleviated forms. In both data sets, $T_{\Delta LUF}$ is the parameter with only very small impact on the evaluation parameters, and only slightly stronger in the Vaihingen test scene than in the Munich test scene.

Assuming that the evaluation parameters indicate the quality of the segmentation[4], the optimal segmentation parameters would produce best scores for all evaluation parameters. An analysis of the mutual dependencies of the evaluation parameters (Fig. 5.8) has shown that such an evaluation output is not possible for any constellation of input parameters. For illustrating the mutual dependencies of evaluation parameters, two quality measures have been computed: $Qm_{1,best}$ is supposed to maximize

[4] The suitability of the evaluation to indicate segmentation and reconstruction quality is discussed in 7.

both completeness C_m and correctness C_r, and $Qm_{2,best}$ is supposed to minimize both undersegmentation N_U and oversegmentation N_O.

$$Qm_{1,best} = \arg\max\left\{\frac{C_m}{\max(C_m)} + \frac{C_r}{\max(C_r)}\right\} \tag{5.1}$$

$$Qm_{2,best} = \arg\min\left\{\frac{N_U}{\max(N_U)} + \frac{N_U}{\max(N_U)}\right\} \tag{5.2}$$

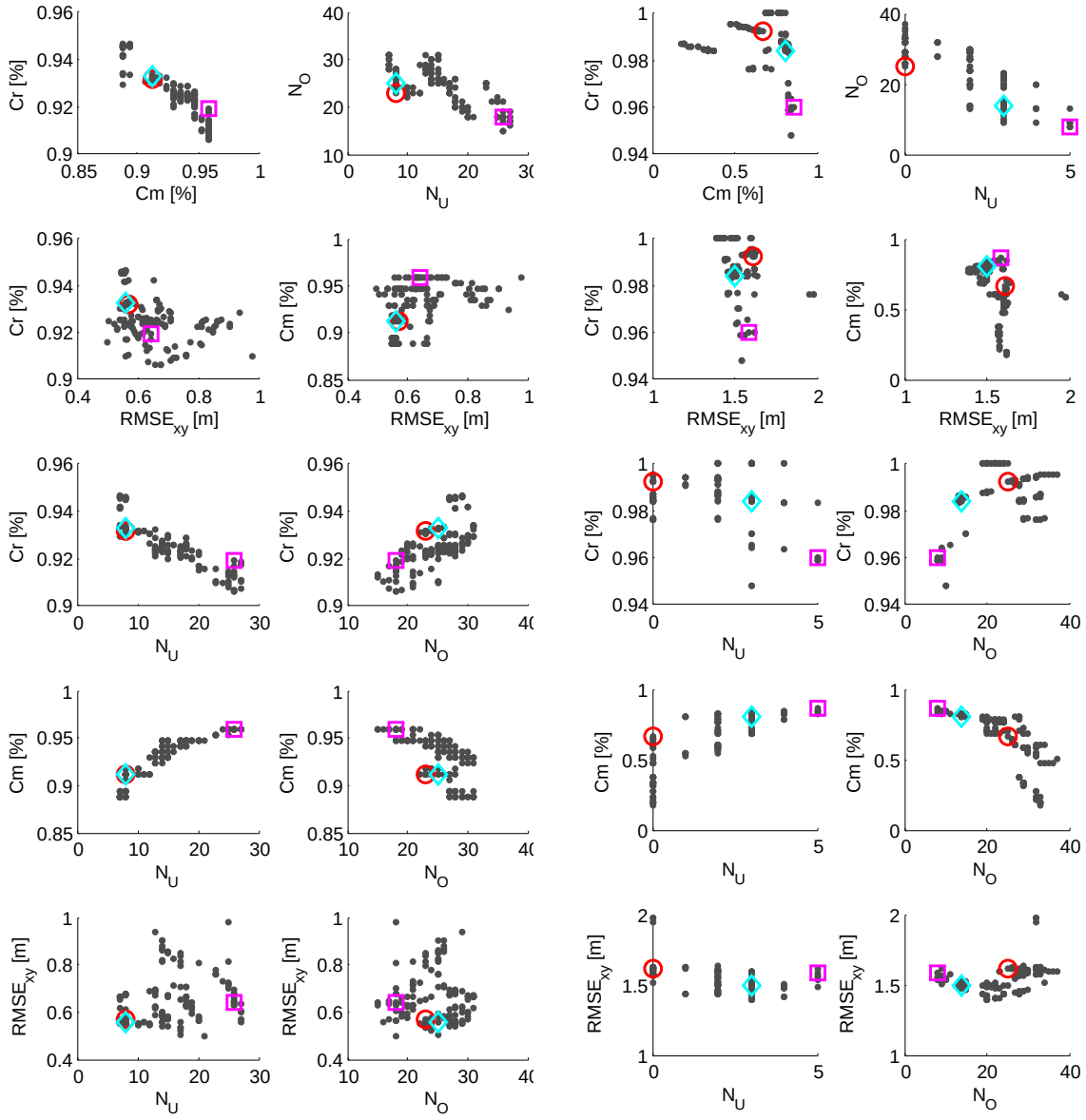
From Fig. 5.8 it is obvious that maximizing completeness and correctness ($Qm_{1,best}$, pink square) does not imply low over- and undersegmentation values, just as minimizing over- and undersegmentation ($Qm_{2,best}$, red circle) does not result in a high completeness and correctness. The evaluation results of the previously determined optimal input parameters are shown by the blue diamond.

TABLE 5.3: Evaluation results for segmentation

Test scene	$\mathbf{C_m}$	$\mathbf{C_r}$	TP_r	FN	TP_e	FP	$\mathbf{N_U}$	$\mathbf{N_O}$	$\mathbf{N_{U\&O}}$	$RMSE_{xy}(std)$
Vaihingen, best visual	91.18	93.27	155	15	194	14	8	25	0	0.5596 (0.4250)
Vaihingen, $Qm_{1,best}$	95.88	91.95	160	7	163	14	26	18	4	0.6425
Vaihingen, $Qm_{2,best}$	91.18	93.20	192	15	155	14	8	23	0	0.5708
Munich, best visual	80.39	98.57	69	9	41	1	3	15	1	1.4608 (0.8612)
Munich, best $Qm_{1,best}$	86.27	96.00	44	7	48	2	5	8	0	1.5864
Munich, best $Qm_{2,best}$	66.67	99.26	34	17	135	1	0	25	0	1.6113

TABLE 5.4: Evaluation results for segmentation, compared to increasing the area threshold to 10 m^2

Test scene	$\mathbf{C_m}$	$\mathbf{C_r}$	$\mathbf{C_{m,10}}$	$\mathbf{C_{r,10}}$
Vaihingen, best visual	91.18	93.27	94.96	97.27
Munich, best visual	80.39	98.57	87.80	100



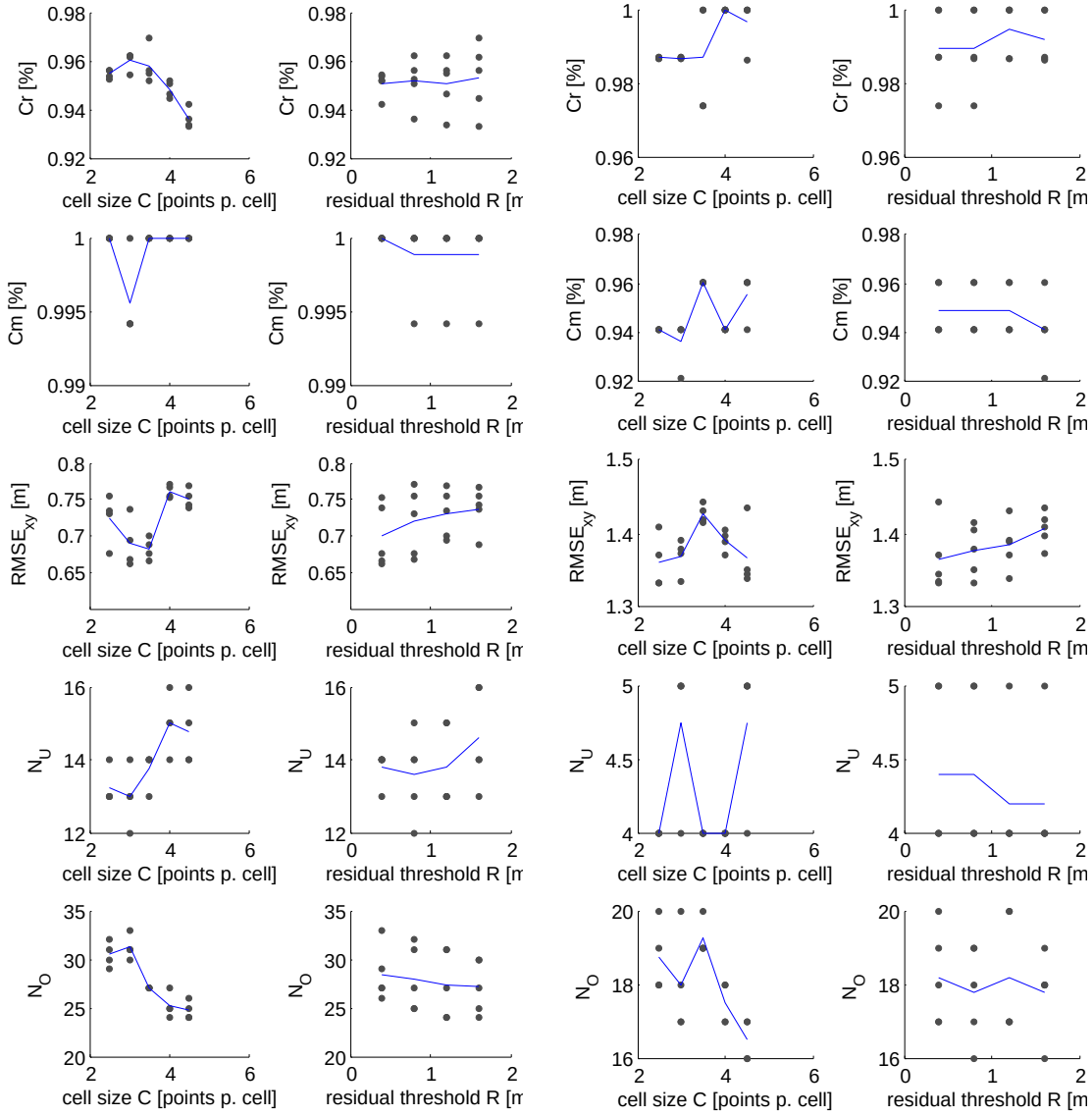
(a) Vaihingen test scene

(b) Munich test scene

FIGURE 5.8: Mutual dependencies of the evaluation parameters. Pink square: results according to inputs maximizing $Qm_{1,best}$; Red circle: results according to inputs minimizing $Qm_{2,best}$; Blue diamond: results of visually determined best input parameters;

5.5 Evaluation of Reconstruction

The effect of input parameters on evaluation parameters is tested also for reconstruction. The cell size C of the dual contouring grid, and the residual threshold R for quadtree collapsing, are identified as the most important input parameters.



(a) Discrete parameter testing for Vaihingen test scene

(b) Discrete parameter testing for Munich test scene

FIGURE 5.9: Influences of the input parameters for reconstruction on the evaluation parameters. The blue lines connect the average evaluation parameters for the respective input parameters.

As seen in Fig. 5.9 a) (discrete input parameter testing for the Vaihingen test scene), a small cell size C improves completeness, correctness, and undersegmentation, and deteriorates $RMSE_{xy}$ and oversegmentation. The residual threshold R has a similar effect to the one of C, but in alleviated form. The evaluation results in Fig. 5.9 b) have a very discrete character, due to the fewer roof segments of the Munich test scene, and are therefore less suited for interpetation.

An analysis of the mutual dependencies of the evaluation parameters to each other, analog to Fig. 5.8, has been carried out[5]. The result of this analysis is very similar to the one for segmentation:

[5] The resulting plots are very similar to the one in Fig. 5.8 and are therfore not shown additionally in this thesis.

Maximizing completeness and correctness ($Qm_{1,best}$) does not correlate with minimizing over- and undersegmentation, just as minimizing over- and undersementation ($Qm_{2,best}$) does not imply high completeness and correctness.

TABLE 5.5: Evaluation results for reconstruction

Test scene	$\mathbf{C_m}$	$\mathbf{C_r}$	TP_r	FN	TP_e	FP	$\mathbf{N_U}$	$\mathbf{N_O}$	$\mathbf{N_{U\&O}}$	$RMSE_{xy}(std)$
Vaihingen, best visual	100	95.35	170	0	205	10	13	32	1	0.7187 (0.5782)
Vaihingen, $Qm_{1,best}$	100	96.98	170	0	193	6	23	25	4	0.7609
Vaihingen, $Qm_{2,best}$	100	95.54	170	0	193	9	7	31	2	0.6637
Munich, best visual	94.12	98.67	48	3	74	1	4	17	0	1.3916 (0.8488)
Munich, $Qm_{1,best}$	96.08	100	49	2	72.6	0	4.6	17.8	2.4	1.3769
Munich, $Qm_{2,best}$	94.12	98.61	48	3	71	1	4	16	1	1.4349

TABLE 5.6: Evaluation results for reconstruction, compared to increasing the area threshold to 10 m^2

Test scene	$\mathbf{C_m}$	$\mathbf{C_r}$	$\mathbf{C_{m,10}}$	$\mathbf{C_{r,10}}$
Vaihingen, best visual	100	95.35	100	95.38
Munich, best visual	96.08	98.67	100	100

Table 5.7 shows different evaluation results for a reconstruction with weighted QEFs (BL-weighting and SP-scaling) and with unweighted QEFs. N_H is the number of hyperpoints which are estimated for the whole scene.

TABLE 5.7: Influence of BL weighting and SP scaling on evaluation parameters, for Vaihingen (VH) and Munich (MUC) test scene

QEF weighting	$\mathbf{C_m}$	$\mathbf{C_r}$	TP_r	FN	TP_e	FP	$\mathbf{N_U}$	$\mathbf{N_O}$	$\mathbf{N_{U\&O}}$	$RMSE_{xy}$	N_H
VH: both	100	95.35	170	0	205	10	13	32	1	0.72	3251
VH: only BL-weighting	100	95.39	170	0	207	10	14	32	1	0.71	3943
VH: only SP-scaling	99.41	95.69	169	1	200	9	13	28	1	0.67	3207
VH: none	100	95.71	170	0	201	9	13	28	1	0.68	3467
MUC: both,	94.12	98.67	48	3	74	1	4	17	0	1.39	1251
MUC: only BL-weighting	94.12	98.72	48	3	77	1	4	20	2	1.37	1560
MUC: only SP-scaling	94.12	98.59	48	3	70	1	5	17	1	1.38	1219
MUC: none	94.12	98.61	48	3	71	1	5	17	1	1.35	1305

6 Discussion

6.1 Discussion of the Segmentation

The robust TIN-base region-growing segmentation (RTINS) is designed to distinguish roof segments even at smooth intersection edges, which are common in complex roof segments. Especially shed roof segments[1] and dormers propose challenges, as they intersect with the underlying basic building shape in obtuse angles.

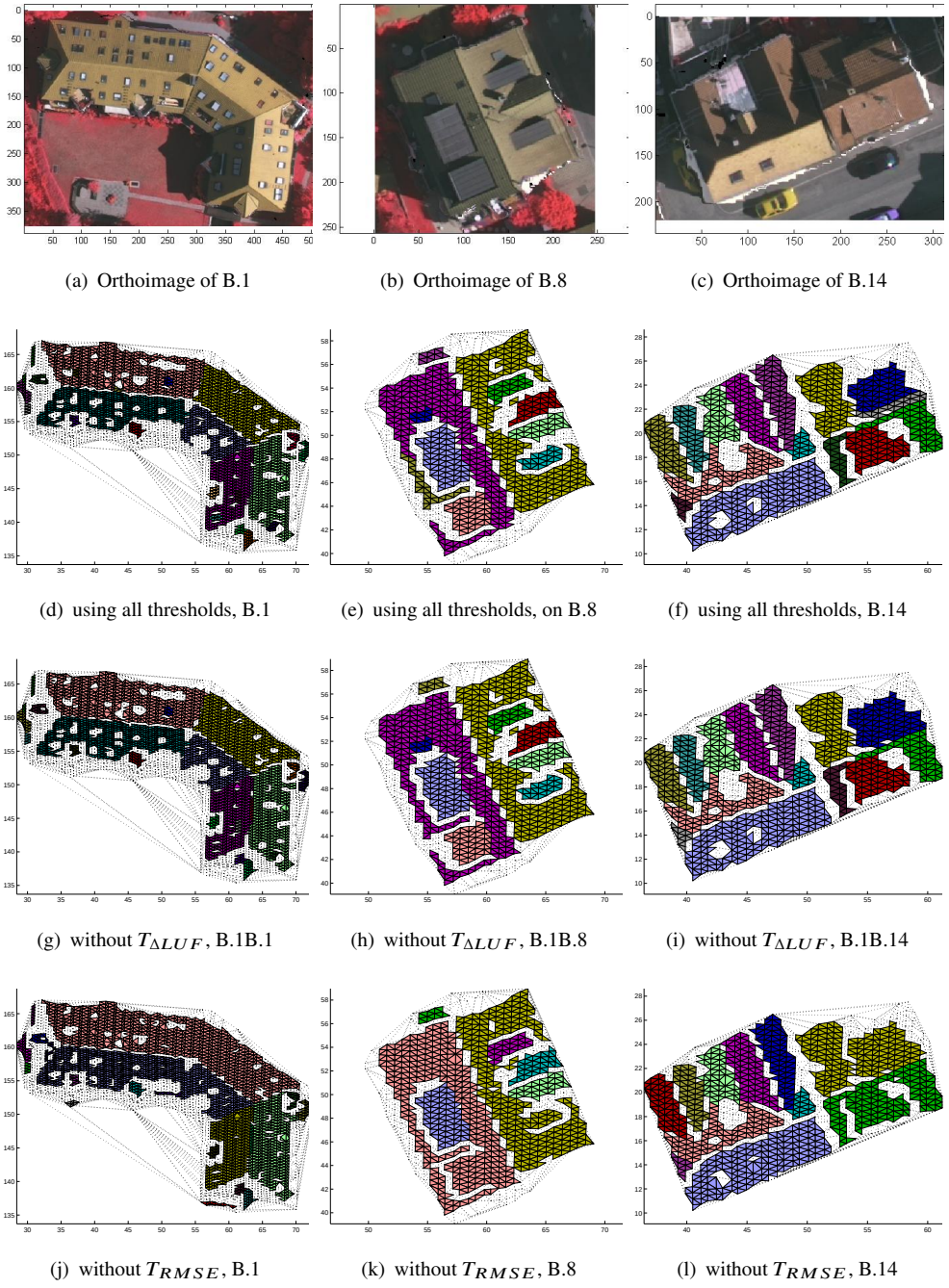
A fixed angular deviation threshold T_Θ is not enough for segmenting complex roof height layers, because regions will generally grow over smooth edges. Therefore, adaptive thresholds have been introduced, which are supposed to allow the region to grow, until abrupt relative changes in curvature ($T_{\Delta LUF}$) or in the points' distances to the region's fitted plane (T_{RMSE}) occur. Both adaptive thresholds are independent from point density variations (T_{RMSE}) or compensate for density variations by introducing corresponding weights into their calulation ($T_{\Delta LUF}$). All of the region growing criteria are computed relative to the region characteristics, which is an important advantage, because also non-planar roof segments can be segmented.

Even though the T_Θ influences completeness and correctness strongest, T_{RMSE} is very important for the segmentation of complex roof height layers. Without T_{RMSE}, a basic roof segment will grow into shed dormers, if only one neighboring triangle of the basic roof structure is similar enough to one in the shed dormer (Fig. 6.1k) and l)). An advantage of the T_{RMSE} in comparison with $T_{\Delta LUF}$ is, that T_{RMSE} can decrease oversegmentation without increasing undersegmentation strongly (Fig. 5.7).

Using $T_{\Delta LUF}$ in addition to T_{RMSE} enforces the effect to stop region growing (RG) at smooth edges (Fig. 6.1 f)). From Fig. 5.7 is concluded that $T_{\Delta LUF}$ has only a small influence on the segmentation result. Its main contribution is to prevent the region to grow into small superstructures, which are only slighly elevated above the underlying basic building structure (Fig. 6.1 g), i)). Therefore, the usage of $T_{\Delta LUF}$ is only recommended if such small superstructures are of importance (e.g. for solar potential analysis), because $T_{\Delta LUF}$ can also be responsible for oversegmentation in certain cases (Fig. 6.1 h)).

All effects of the RG criteria on the evalution parameters are more obvious for the Vaihingen test scene, which has a significantly higher number of roof planes and whose evaluation parameters are therefore considered as more reliable. Different optimal input parameters for the two different test scenes are determined by visual interpretation of the results. As the two data sets are very different in data characteristics, it is proposed to perform an analysis of the correlation of data characteristics with the optimal input parameters in future work.

[1] Shed roof segments are flat superstructures which intersect one side with the underlying building structure, and form step edges on the other sides (e.g. in Fig. 6.1 b) and c)), and are very common in western european regions.



(a) Orthoimage of B.1 (b) Orthoimage of B.8 (c) Orthoimage of B.14

(d) using all thresholds, B.1 (e) using all thresholds, on B.8 (f) using all thresholds, B.14

(g) without $T_{\Delta LUF}$, B.1B.1 (h) without $T_{\Delta LUF}$, B.1B.8 (i) without $T_{\Delta LUF}$, B.1B.14

(j) without T_{RMSE}, B.1 (k) without T_{RMSE}, B.8 (l) without T_{RMSE}, B.14

FIGURE 6.1: Influence of region growing criteria on segmentation, shown in the local coordinate system of the scene.

Apart from the proposed adaptive thresholds, a unique feature of RTINS is the growing of TIN-triangles instead of points. This allows LiDAR points at the border of intersecting segments to be segmented to two or more different regions. 29.84 % of the segment border points in the Vaihingen test scene and 31.37 % of those in the Munich test scene are segmented to more than one region. The advantage of multiple point segmentation is that local boundary lines between segments can be determined more accurately than if each point is assigned to only one segment. In the latter case, the space between adjacent segments' border points forms a gap in the segmentation. For such gaps, local

boundary lines are estimated along their center. This assumption is automatically more correct when segmenting triangles: Gaps will only occur if none of the border points of one region is close enough to the other region to be segmented for that segment, too. Especially for the proposed modeling algorithm it is important to determine local boundaries accurately. RTINS is therefore a good match to the proposed modeling procedure. However, it can cause oversegmentation, if the gaps between intersecting roof segments are interpreted as individual regions.

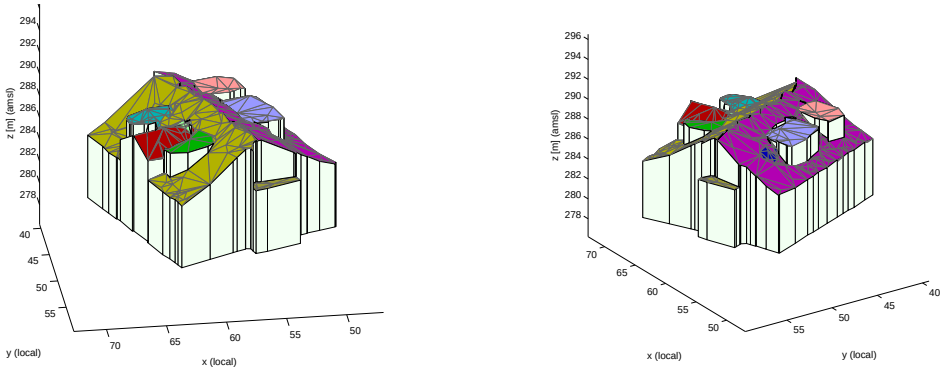
Comparing the evaluation results with those of [Awrangjeb and Fraser, 2014], who apply the same evaluation procedure on the Vaihingen test scene, shows that the proposed segmentation oversegments the point cloud stronger (Table 6.1). This leads to the conclusion that the RTINS method is very sensible to edges in the point cloud. With a similar $RMSE_{xy}$ and clearly better values for completeness C_m, correctness C_r, and undersegmentation N_U, RTINS outperforms the method of [Awrangjeb and Fraser, 2014] clearly. As oversegmentation is not deteriorating the modeling results as much as undersegmentation (Section 6.2), the RTINS technique is very well suited for the proposed modeling algorithm.

TABLE 6.1: Comparison of semgentation results with existing method, Vaihingen test scene

Segmentation methods	$\mathbf{C_m}$	$\mathbf{C_r}$	$\mathbf{C_{m,10}}$	$\mathbf{C_{r,10}}$	$\mathbf{N_U}$	$\mathbf{N_O}$	$\mathbf{N_{U\&O}}$	$RMSE_{xy}(Std.)$
Proposed	91.18	93.27	94.96	97.27	8	25	0	0.56 (0.43)
Awrangjeb & Fraser, 2014	76.4	83.3	84.4	84.9	42	6	7	0.41

6.2 Discussion of the Reconstruction

The unique feature of the proposed reconstruction method is using a detailed segmentation as input to a 2.5D dual contouring approach. For modeling complex superstructures, the QEF for hyperpoint estimation has been modified, such that intersection edges and step edges can be created between segments from one height layer[2] (Fig. 6.2).



(a) Model of building 8, with SP scaling, but without BL weighting (view1)

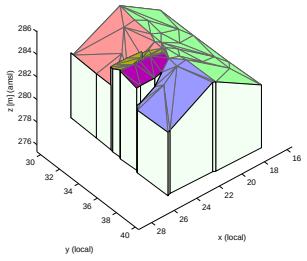
(b) Model of building 8, with SP scaling, but without BL weighting (view2)

FIGURE 6.2: Detailed model of a building. Different segments are modeled in detail, segments on one height layer are separeted by both step edges and intersection edges.

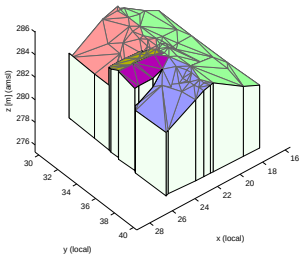
[2] Shed roof segments and dormers can only be modeled accurately with both intersection edges and step edges to the underlying basic roof segment.

Depending on whether step edges or intersection edges are estimated, the weights of the local boundary lines **BL** for QEF construction are adapted with respect to the local surface planes **SP** (BL-weighting). Additionally, the **SP** are scaled according to the number of equally labeled **SP** in the QEF (SP-scaling). Both weightings have the purpose of improving the hyperpoints' accuracy. Comparing weighted reconstruction with a non-weighted reconstruction (Table 5.7) reveals that the weightings have no significant effect on the evaluation results. However the number of hyperpoints N_H is influenced: While BL-weighting increases N_H, SP-scaling reduces N_H such that, if both modifications are applied, the results have fewer hyperpoints than without any modifications. Fig. 6.3 shows that effects of BL-weighting and SP-scaling on a building model are comparatively small. At the building borders, the number of hyperpoints is reduced, such that a simpler polygon is created.

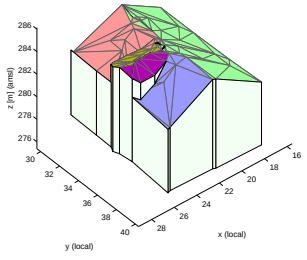
The main purpose of BL-weighting is to restrict hyperpoints by the local boundary lines for step edges, and to minimize the boundary lines' restrictive effect for intersection edges. This can lead to a more regularized building model, if the **SP** along a roof ridge line are close to coplanar. But if those **SP** along a ridge line vary strongly, BL-weighting impedes the restrictive effect of potentially parallel **BL**, such that a non-weighted solution may result in the more regularized model. I.e., the impact of BL-weighting concerning model simplicity depends on the variability of **SP** of a (planar) roof segment within one QEF cell.



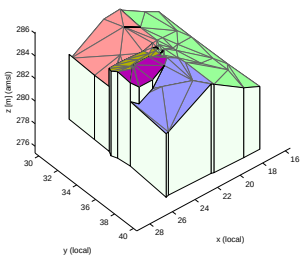
(a) BL-weighting & SP-scaling: Especially at the building boundary, the number of hyperpoints are reduced to a minimum.



(b) only BL-weighting: without SP scaling, the building has more hyperpoints and therefore more polygon faces



(c) only SP-scaling: without BL weighting, the roof is more jagged, because local boundary lines restrict the hyperpoints at intersection edges

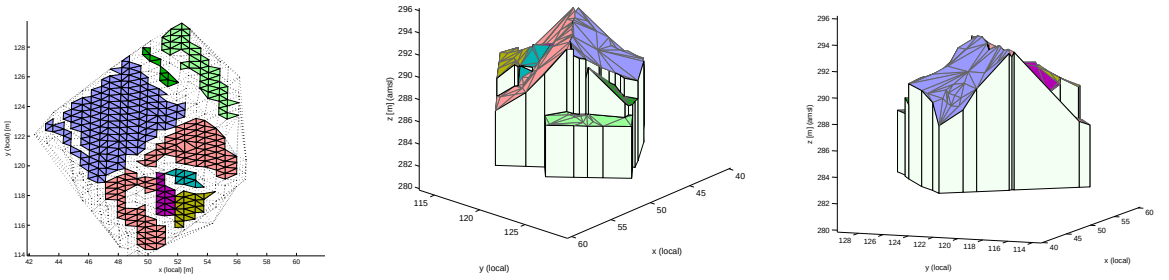


(d) no QEF weighting: higher number of hyperpoitns at the boundaries, roof ridge lines are slightly more jagged, and the building has slightly more polygon faces

FIGURE 6.3: Effects of QEF weighting on building modeling

This variability is influenced by the number of LiDAR points K which are considered for **SP** calculation, and by the data characteristics. It is proposed for future work to perform an analysis of the dependencies between K and data characteristics, and the contribution of BL-weigthing for model simplicity.

Prerequisite for QEF modification is a detailed roof segmentation. The example of Fig. 6.4 illustrates the behaviour of the modeling method in case of undersegmentation. Where the roof is undersegmented (blue segment), the modeling method has a similar behaviour as the approach of Zhou and Neumann [2010], who can create step edges only between different height layers. Even though no step edge is created in case of wrong segmentation, 2.5D dual contouring is able to approximate the roof form in detail by a detailed roof triangulation. Representing roof segments by triangulation has the advantage to model roof segments which are not necessarily planar, and to attain a high fit of the model to the data, at the expense of model simplicity.



(a) Undersegmentation at the blue segment: opposite of the dark green segment, a similar separate segment should have been identified.

(b) Reconstruction, step edge between the blue and dark green segment

(c) Reconstruction, no step edge can be modeled, because no different segments are identified

FIGURE 6.4: Effect of QEF modification and detailed segmentation: Where the roof height layer is undersegmented, no step edges can be created. However, the roof is approximated to fit the input point cloud in detail.

The models' grade of detail can be chosen flexibly by adapting the input parameters accordingly. The smaller the grid size C and the smaller the residual threshold R, the more accurate is the result. Fig. 5.9 shows that the $RMSE_{xy}$ improves with decreasing either of the two parameters. However, the building is represented with more hyperpoints, and therefore more roof triangles, when C and R are chosen small (Fig. 6.5). Therefore, 2.5D dual contouring requires a trade-off between the simplicity of the model (low number of hyperpoints), and the accuracy and grade of detail of the model. Additionally it has to be considered, that the computation time increases exponentially with the number of grid cells $N_{c/grid}$, which has to be $2^(2n)$ for creating a quadtree. Changing the cell size slightly smaller can lead to a jump in $N_{c/grid}$ from 2^{2n} to $2^{2(n+1)}$, leading to an exponential increase in the number of computations.

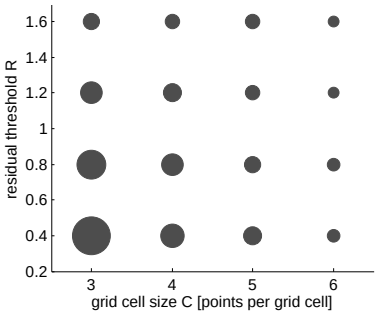
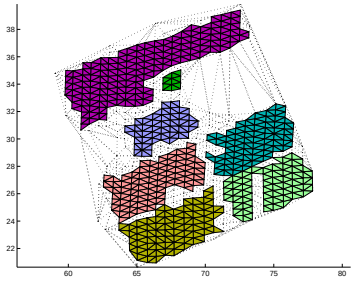
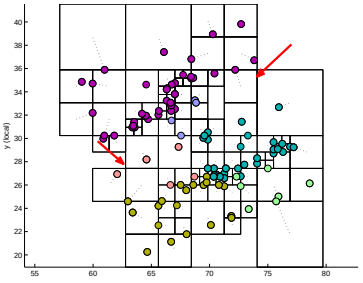


FIGURE 6.5: Influences of reconstruction parameters on the number of hyperpoints (Munich test scene). The radius of the grey circles corresponds to the number of hyperpoints resulting from the corresponding reconstruction parameters C and R.

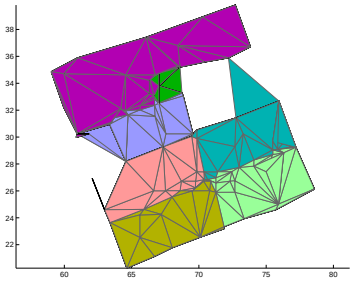
The building model's 3D edges are constructed from connecting adjacent hyperpoints in the quadtree. If buildings are not compact, but have concave forms, this can lead to erroneous edges (Fig. 6.6, right arrow in subfigure b)). These errors can be corrected in future work by checking whether the resulting edge's meanpoint is close to the building footprint. Other errors occur, when hyperpoints, which shall be connected, are not adjacent in the quadtree (Fig. 6.6, left arrow in subfigure b)), which may result in deformations or stand-alone "walls" in the model. This problem could be solved by performing a rotation of the building point cloud before the modeling procedure, such that it is aligned with the main building direction. As building edges are assumed to be mostly orthogonal and parallel to the main bulding direction, there are fewer cases where hyperpoints from non-adjacent quadtree cells have to connect to an edge.



(a) Segmentation of the building point cloud; differently colored triangles are segmented to different regions (colors are arbitrary)



(b) Collapsed quadtree with hyperpoint solutions (differently colored points, colored according to the segment they are labled with). The red arrows indicate positions phere neighboring quadtree cells' hyperpoints are erroneously connected (right arrow), or where hyperpoints are erroneously not connected, because they are no neighbors in the quadtree (left arrow).



(c) 3D model resulting from erroneous hyperpoint connection (top view). In the right part of the building, a wrong triangle is created from erroneously connecting adjacent hyperpoints. In the left part, a triangle is missing, because the corresponding hyperpoints are not adjacent in the quadtree.

FIGURE 6.6: Errors due to connecting hyperpoints from neighboring quad regions

Another problem resulting from quadtree collapsing is the creation of unnecessary hyperpoints for cells which cannot be collapsed to a quad. Simple roof segments, which could be accurately modeled by few hyperpoints, will then be represented by more hyperpoints than required (Fig. 6.7 b)). As

an example, for creating only the four necessary hyperpoints of a planar segment with rectilinear outlines, the quadtree can collaps until only four quadtree cells remain for this region (Fig. 6.7 a)). If however the segment is adjacent to smaller segments (Fig. 6.7b)), the additional hyperpoints for modeling the other segments impede further quadtree collapsing. The problem is, that for each non-collapsed quadtree cell, an own hyperpoint is estimated, which results in unnecessary hyperpoints (red points in Fig. 6.7 b)).

(a) Rectilinear flat segment represented by four hyperpoints (blue points), in a maximally collapsed quadtree; quadtree cells plotted with black lines; dotted lines indicate the centers of the quadtree cells from which the hyperpoints are computed

(b) Additional adjacent rectilinear segments represented by more hyperpoints. The quadtree cannot collaps, because the addional hyperpoints have to be created, such that also unnecessary hyperpoints are created for the other cells of the non-collapsed quad (red points); dotted lines indicate the centers of the quadtree cells from which the hyperpoints are computed

FIGURE 6.7: Illustration of the creation of unnecessary hyperpoints due to complex building roofs. In the right image, the quadtree cannot collaps further, because additional hyperpoints have to be estimated for additional segments. However, also in those un-collapsed quadtree cells where is no additional segment, an additional hyperpoint is created (red points).

As the building models are created from a triangulation of the hyperpoints, additional hyperpoints also increase the number of triangles, such that building models can look jagged and irregular. It is therefore recommended for future work to enhance aesthetics and simplicity of the models by means of a regularization. Planimetric regularization can be achieved by correcting the x-y-position of hyperpoints which connect nearly (planimetrically) orthogonal edges. Unnecessary hyperpoints which connect nearly parallel edges can be removed, and corresponding wall segments can be merged. The number of roof polygon faces can be reduced by merging hyperpoints with similar horizontal positions, and by iteratively merging adjacent, nearly coplanar roof triangles, while correcting the vertical positions of the corresponding hyperpoint vertices accordingly. As an alternative to regularization, the quadtree cells for hyperpoint computation could be further merged with adjacent cells, which are not necessarily inside the same quadtree cell.

Comparing adapted 2.5D dual contouring with the reconstruction approaches of the ISPRS benchmark (Table 6.2) reveals, that the proposed method outperfroms all other approaches in completeness and undersegmentation. Comparing C_m with $C_{m,10}$ and C_r with $C_{r,10}$ reveals that the proposed method is robustly detecting segments of any size. High oversegmentation is explained by the sensibility of the segmentation procedure to smooth edges (Section 6.1), but oversegmentation does not influence the quality of the building models such as undersegmentation (Fig.6.4). Considering high

standard deviation of the $RMSE_{xy}$ (0.58), the proposed method's results are comparable to the other methods concerning horizontal accuracy.

TABLE 6.2: Comparison of reconstruction results to other methods, Vaihingen test scene

Method	$\mathbf{C_m}$	$\mathbf{C_r}$	$\mathbf{C_{m,10}}$	$\mathbf{C_{r,10}}$	$\mathbf{N_U}$	$\mathbf{N_O}$	$\mathbf{N_{U\&O}}$	$RMSE_{xy}$
Proposed	100	95.35	100	95.38	13	32	1	0.72
Rau & Lin, 2011	86.7	98.9	86.7	99.3	36	10	3	0.66
Oude Elberink & Vosselman, 2009	60.8	94.6	58.5	94.0	26	16	17	0.91
Oude Elberink & Vosselman, 2011	65.3	97.3	63.3	97.3	38	0	3	0.94
Xiong, s. Rottensteiner et. al, 2012	76.0	94.5	72.9	95.1	40	2	2	0.84
Dorninger & Pfeifer, 2008	72.2	96.7	77.7	96.5	42	7	6	0.79
Sohn et al., 2008	88.2	98.5	89.9	98.2	36	5	14	0.75

The high completeness can be explained by the fact, that the proposed reconstruction method fills the gaps between the footprint polygon and the segment boundary polygons with grid data from the closest roof segments. Even for unsegmented regions inside the building footprints, the modeling algorithm constructs roof parts using the corresponding grid data. Therefore, each reference roof segment within the building boundary has a corresponding model roof part, leading to high completeness. The modeling algorithm is therefore based on two assumptions. First, that the respectively closest segments can be extended to unsegmented regions within the building footprints. The second assumption is, that the footprints are correct.

Footprints don't necessarily need to be available beforehand, but can be extracted by automated building detection, which is proposed as as a future improvement of this work[3]. Including automatic building detection, the proposed procedure is a fully automatic building reconstruction workflow. Except for the step edge threshold, no assumptions are needed for estimating the 3D vertices of the model.

Comparing the evaluation results for the two different data sets Munich and Vaihingen (Table 5.5) leads to the conclusion, that the proposed reconstruction workflow is independent of data characteristics and ground truth, and shows very high robustness for different input parameters.

6.3 Discussion of the Evaluation Method

From analyzing the effect of different input parameters on evaluation results (Fig. 5.8) can be seen that the $RMSE_{xy}$ is correlated with oversegmentation. An estimated segment corresponding to an oversegmented reference has boundary vertices which have no corresponding reference boundary (Fig. 6.8). Nevertheless, these vertices' distances to their closest reference boundary are taken into account for $RMSE_{xy}$ calculation. The ISPRS benchmark project uses hereby a threshold of 3m, leading to a high standard deviation $RMSE_{xy}$ in case of oversegmentation. With such a high standard deviation, the results in $RMSE_{xy}$ are not comparable any more. Therefore it is suggested to future

[3] As DTM production by morphological filtering is already part of this work, building point clouds can be extracted from the elevated data (after ground filtering) by a connected component analysis. Elevated connected components which are large enough, are identified as buildings.

benchmark projects to evaluate only polygon vertices whose closest reference boundary is closer than any other estimated segment boundary.

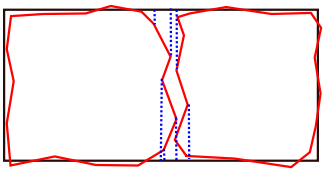


FIGURE 6.8: Illustration of correlation between RMSE and oversegmentation. Red polygons: estimated segments' polygons; black polygon: reference polygon; dotted blue lines: problematic distances of polygon vertices to their closest reference polygon segment, due to oversegmentation.

Also the object-based calculation of C_m, and C_r is criticized, as the size of the detected segments (TP_r) is not considered, leading to different C_m for different tested areas [Rottensteiner et al., 2012a]. Both C_m, and C_r are correlated with over- and undersegmentation (Fig. 5.8): N_O is increasing, respectively N_U is decreasing the number of correct segments TP_e, which determines C_r. The correlation of undersegmentation with C_m is explained by the fact, that some very small segments are only detected if larger regions are allowed to grow into them, implying both a larger C_m and larger N_U.

Furthermore it has to be considered, that ground truth was extracted manually from the orthoimage and is therefore not equal with the ground truth used by the ISPRS benchmark project. Discrepancies can result from different criteria for delineating segments. Where a roof part may be described with one segment in one ground truth, there may be two segments in the other ground truth (Fig. 6.9, arrow "a"), which can lead to different results for oversegmentation $\mathbf{N_O}$ and undersegmentation $\mathbf{N_U}$.

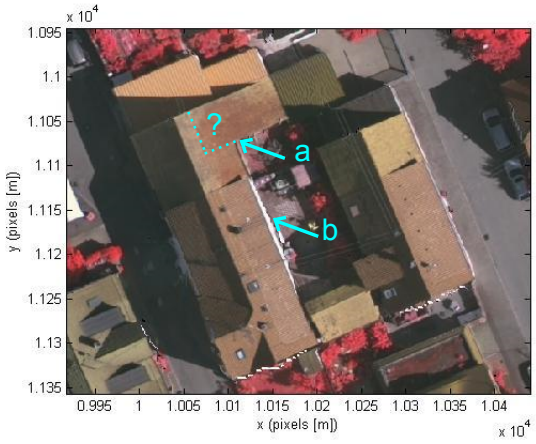


FIGURE 6.9: Problems in determining reference segments from an orthoimage. Arrow "a" points to a roof segment which could be seen as two different roof segments, or as only one, depending on individual decision criteria. Arrow "b" points to a white wall which is visible below the roof structure due to projection inaccuracies.

Additional ground truth errors may occur due to projection inaccuracies[4] in the orthophoto (Fig. 6.9, arrow "b"), which may result in wrong reference segments. For better comparability of the proposed

[4] Projection inaccuracies can happen at steep acquisition's viewing angles, at roof segments' height discrepancies and at different steepness.

method with other approaches, the reconstruction results of the Vaihingen test scene are submitted to the IPSRS benchmark project for a second evaluation.

7 Conclusion

The proposed workflow was developed to create detailed building models for complex roof shapes, which fit the input data in a precise manner. A 2.5D dual contouring method was adapted such that step edges and intersection edges can be created between roof segments within one roof height layer.

Prerequisite to the proposed reconstruction method is a robust segmentation technique which can identify segment delineations even at smooth edges. Evaluation and testing have shown that the proposed robust TIN-based region growing segmentation (RTINS) is very sensitive to smooth edges. In contrast to other methods, RTINS results in significantly less undersegmentation, and improves accurate boundary estimation, which is essential for the proposed modeling algorithm.

The main contribution of this work is the modification and weighting of the Quadratic Error Function (QEF) used by the 2.5D dual contouring algorithm for estimating vertices of the building model. The basic difference of the proposed modeling procedure to the approach of Zhou and Neumann [2010] is the use of a detailed roof segmentation as input, instead of only height layers. The QEF was modified such that it results in one or more building vertices at the same x-y-coordinates, depending on whether intersection edges or step edges shall be computed. Therefore, columns of the QEF matrix equation have been rearranged according to local step edge probabilities (QEF modification). Additionally, the local grid data from which the QEF matrix equation's rows are constructed are weighted and scaled (QEF weighting). Considering that "the model must be adequate both in terms of the solution attained and the cost to attain the solution" [Haala et al., 1998], the contribution of QEF weighting for improving the model accuracy is considered small compared to the additional effort. However, QEF modification is indispensable for detailed modeling of complex roofs, and is therefore regarded as the main contribution of this work. Building models which are created with adapted 2.5D dual contouring are very detailed and can fit the input data in a very precise manner. Depending on the chosen input parameters, a user can produce results at the required level of detail. As this work does not assume each roof plane to be planar, each segment is represented by a triangulation. A trade-off between the required level of detail and the simplicity of the model has to be made. Subsequent model regularization is recommended, as buildings are represented by a high number of vertices, and model simplicity can be further improved.

Testing and evaluating two different data sets has shown that the proposed segmentation and reconstruction methods are very robust against varying point cloud characteristics and building complexity. A detailed analysis of the effects of different input parameters attest to the robustness of the presented methods for changes in input parameters. Thanks to the ISPRS benchmark test, the performance of the designed workfow could be compared to existing methods. Evaluation results have shown that both segmentation and reconstruction procedures outperform other methods, with outstanding values for completeness and undersegmentation, wich are considered as the most important quality criteria.

The transparency in the research field depends very much on using standardized evaluation methods and equal data sets. It is suggested to future benchmark projects to provide open-source evaluation software, test data and ground truths, in order to facilitate the assessment of research results in the field of 3D building reconstruction from airborne LiDAR point clouds.

A Appendix

Weighted and modified QEF matrix equation:

$$
\mathbf{A}_w\hat{\mathbf{X}} - \mathbf{b}_w = \begin{pmatrix}
w_{k,k+1}\cdot n_{i,x}^{k,k+1} & w_{k,k+1}\cdot n_{i,y}^{k,k+1} & 0 & 0 & \dots \\
w_{k,k+1}\cdot n_{i+1,x}^{k,k+1} & w_{k,k+1}\cdot n_{i+1,y}^{k,k+1} & 0 & 0 & \dots \\
\dots & \dots & \dots & \dots & \dots \\
w_{k,k+2}\cdot n_{i,x}^{k,k+2} & w_{k,k+2}\cdot n_{i,y}^{k,k+2} & 0 & 0 & \dots \\
w_{k,k+2}\cdot n_{i+1,x}^{k,k+2} & w_{k,k+2}\cdot n_{i+1,y}^{k,k+2} & 0 & 0 & \dots \\
\dots & \dots & \dots & \dots & \dots \\
w_{k+1,k+2}\cdot n_{i,x}^{k+1,k+2} & w_{k+1,k+2}\cdot n_{i,y}^{k+1,k+2} & 0 & 0 & \dots \\
w_{k+1,k+2}\cdot n_{i+1,x}^{k+1,k+2} & w_{k+1,k+2}\cdot n_{i+1,y}^{k+1,k+2} & 0 & 0 & \dots \\
\dots & \dots & \dots & \dots & \dots \\
m_{j,x}^{k} & m_{j,y}^{k} & m_{j,z}^{k} & 0 & \dots \\
m_{j+1,x}^{k} & m_{j+1,y}^{k} & m_{j+1,z}^{k} & 0 & \dots \\
\dots & \dots & \dots & \dots & \dots \\
m_{j,x}^{k+1} & m_{j,y}^{k+1} & m_{j,z}^{k+1} & 0 & \dots \\
m_{j+1,x}^{k+1} & m_{j+1,y}^{k+1} & m_{j+1,z}^{k+1} & 0 & \dots \\
\dots & \dots & \dots & \dots & \dots \\
m_{j,x}^{k+2} & m_{j,y}^{k+2} & 0 & m_{j,z}^{k+2} & \dots \\
m_{j+1,x}^{k+2} & m_{j+1,y}^{k+2} & 0 & m_{j+1,z}^{k+2} & \dots \\
\dots & \dots & \dots & \dots & \dots
\end{pmatrix}
\begin{pmatrix} x \\ y \\ z_v \\ z_{v+1} \\ \dots \end{pmatrix} -
$$

$$
\begin{pmatrix}
w_{k,k+1}\left(n_{i,x}^{k,k+1}\cdot p_{i,x} + n_{i,y}^{k,k+1}\cdot p_{i,y}\right) \\
w_{k,k+1}\left(n_{i+1,x}^{k,k+1}\cdot p_{i+1,x} + n_{i+1,y}^{k,k+1}\cdot p_{i+1,y}\right) \\
\dots \\
w_{k,k+2}\left(n_{i,x}^{k,k+2}\cdot p_{i,x} + n_{i,y}^{k,k+2}\cdot p_{i,y}\right) \\
w_{k,k+2}\left(n_{i+1,x}^{k,k+2}\cdot p_{i+1,x} + n_{i+1,y}^{k,k+2}\cdot p_{i+1,y}\right) \\
\dots \\
w_{k+1,k+2}\left(n_{i,x}^{k+1,k+2}\cdot p_{i,x} + n_{i,y}^{k+1,k+2}\cdot p_{i,y}\right) \\
w_{k+1,k+2}\left(n_{i+1,x}^{k+1,k+2}\cdot p_{i+1,x} + n_{i+1,y}^{k+1,k+2}\cdot p_{i+1,y}\right) \\
\dots \\
m_{j,x}^{k}\cdot q_{j,x}^{k} + m_{j,y}^{k}\cdot q_{j,y}^{k} + m_{j,z}^{k}\cdot q_{j,z}^{k} \\
m_{j+1,x}^{k}\cdot q_{j+1,x}^{k} + m_{j+1,y}^{k}\cdot q_{j+1,y}^{k} + m_{j+1,z}^{k}\cdot q_{j+1,z}^{k} \\
\dots \\
m_{j,x}^{k+1}\cdot q_{j,x}^{k+1} + m_{j,y}^{k+1}\cdot q_{j,y}^{k+1} + m_{j,z}^{k+1}\cdot q_{j,z}^{k+1} \\
m_{j+1,x}^{k+1}\cdot q_{j+1,x}^{k+1} + m_{j+1,y}^{k+1}\cdot q_{j+1,y}^{k+1} + m_{j+1,z}^{k+1}\cdot q_{j+1,z}^{k+1} \\
\dots \\
m_{j,x}^{k+2}\cdot q_{j,x}^{k+2} + m_{j,y}^{k+2}\cdot q_{j,y}^{k+2} + m_{j,z}^{k+2}\cdot q_{j,z}^{k+2} \\
m_{j+1,x}^{k+2}\cdot q_{j+1,x}^{k+2} + m_{j+1,y}^{k+2}\cdot q_{j+1,y}^{k+2} + m_{j+1,z}^{k+2}\cdot q_{j+1,z}^{k+2} \\
\dots
\end{pmatrix}
\tag{A.1}
$$

Bibliography

AG, L. G. (2014). Airborne lidar: Higher productivity with best-in-class lidar technology. http://www.leica-geosystems.com/en/airborne-lidar-86814.htm, [14-08-2014].

Alharthy, A. and Bethel, J. (2004). Detailed building reconstruction from airborne laser data using a moving surface method. *International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences*, 34(B3):213–218.

Ameri, B. and Fritsch, D. (2000). Automatic 3d building reconstruction using plane-roof structures. *In Proc. of the ASPRS 2015 Annual Conference*.

Awrangjeb, M. and Fraser, C. S. (2014). Automatic segmentation of raw lidar data for extraction of building roofs. *Remote Sensing*, 6(5):3716–3751.

Baltsavias, E. P. (1999a). Airborne laser scanning: basic relations and formulas. *ISPRS Journal of Photogrammetry and Remote Sensing*, 54(2):199–214.

Baltsavias, E. P. (1999b). Airborne laser scanning: existing systems and firms and other resources. *ISPRS Journal of Photogrammetry and Remote Sensing*, 54(2):164–198.

Brenner, C. (2000). Towards fully automatic generation of city models. *International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences*, 33(B3/1):84–92.

Brenner, C. (2001). City models - automation in research and practise. *Photogrammetric Week*, pages 149–158.

Brenner, C. (2005). Building reconstruction from images and laser scanning. *International Journal of Applied Earth Observation and Geoinformation*, 6(3):187–198.

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

Fiocco, M., Bostrom, G., Gonçalves, J. G., and Sequeira, V. (2005). Multisensor fusion for volumetric reconstruction of large outdoor areas. *In Proc. of the Fifth IEEE International Conference on 3-D Digital Imaging and Modeling (3DIM)*, pages 47–54.

Geibel, R. and Stilla, U. (2000). Segmentation of laser altimeter data for building reconstruction: different procedures and comparison. *International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences*, 33(B3/1):326–334.

Haala, N. and Brenner, C. (1999). Extraction of buildings and trees in urban environments. *ISPRS Journal of Photogrammetry and Remote Sensing*, 54(2):130–137.

Haala, N., Brenner, C., and Anders, K. H. (1998). 3d urban gis from laser altimeter and 2d map data. *International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences*, 32:339–346.

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

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

Ju, T., Losasso, F., Schaefer, S., and Warren, J. (2002). Dual contouring of hermite data. *ACM Transactions on Graphics (TOG)*, 21(3):339–346.

Kada, M. and McKinley, L. (2009). 3d building reconstruction from lidar based on a cell decomposition approach. *International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences*, 38(3), W4:47–52.

Kada, M. and Wichmann, A. (2012). Sub-surface growing and boundary generalization for 3d building reconstruction. *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, I-3.

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

Kolbe, T. H. (2012). Exchange and storage of virtual 3d city models. http://www.citygml.org, [29-04-2012].

Lafarge, F. and Mallet, C. (2012). Creating large-scale city models from 3d-point clouds: A robust approach with hybrid representation. *International Journal of Computer Vision*, 99(1):69–85.

Maas, H. G. and Vosselman, G. (1999). Two algorithms for extracting building models from raw laser altimetry data. *ISPRS Journal of Photogrammetry and Remote Sensing*, 54(2):153–163.

Matei, B. C., Sawhney, H. S., Samarasekera, S., Kim, J., and Kumar, R. (2008). Building segmentation for densely built urban regions using aerial lidar data. *In Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1–8.

Meng, X., Currit, N., and Zhao, K. (2010). Ground filtering algorithms for airborne lidar data: A review of critical issues. *Remote Sensing*, 2(3):833–860.

Morgan, M. and Tempfli, K. (2000). Automatic building extraction from airborne laser scanning data. *International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences*, 33(B3/2):616–623.

Nurunnabi, A., Belton, D., and West, G. (2012). Robust segmentation in laser scanning 3d point cloud data. *In Proc. of the IEEE International Conference on Digital Image Computing Techniques and Applications (DICTA)*, 1:1–8.

Oude Elberink, S. (2008). Problems in automated building reconstruction based on dense airborne laser scanning data. *International Archives of Photogrammetry, Remote Sensing and Spatial Information Sience*, 37(B3):93–98.

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

Perera, S. N., Nalani, H. A., and Maas, H. G. (2012). An automated method for 3d roof outline generation and regularization in airbone laser scanner data. *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, I-3:281–286.

Pfeifer, N., Rutzinger, M., Rottensteiner, F., Muecke, W., and Hollaus, M. (2007). Extraction of building footprints from airborne laser scanning: Comparison and validation techniques. *In Proc. of the IEEE Urban Remote Sensing Joint Event*, pages 1–9.

Poullis, C. and You, S. (2009). Automatic reconstruction of cities from remote sensor data. *In Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 1:2775–2782.

Rau, J. Y. and Lin, B. C. (2011). Automatic roof model reconstruction from als data and 2d ground plans based on side projection and the tmr algorithm. *ISPRS Journal of Photogrammetry and Remote Sensing*, 66(6):13–27.

Rottensteiner, F. (2003). Automatic generation of high-quality building models from lidar data. *IEEE Computer Graphics and Applications*, 23:42Ű50.

Rottensteiner, F. (2012). Isprs test project on urban classification and 3d building reconstruction: Evaluation of object detection results. http://www2.isprs.org/commissions/comm3/wg4/results.html, [02-07-2014].

Rottensteiner, F., Sohn, G., Jung, J., Gerke, M., Baillard, C., Benitez, S., and Breitkopf, U. (2012a). The isprs benchmark on urban object classification and 3d building reconstruction. *ISPRS Annals of Photogrammetry, Remote Sensing and Spatial Information Sciences*, I-3:293–298.

Rottensteiner, F., Sohn, G., Jung, J., Gerke, M., Baillard, C., Benitez, S., and Breitkopf, U. (2012b). The isprs benchmark on urban object classification and 3d building reconstruction. http://www2.isprs.org/commissions/comm3/wg4/results.html, [04-08-2014].

Rottensteiner, F., Trinder, J., Clode, S., and Kubik, K. (2005). Using the dempster-shafer method for the fusion of lidar data and multi-spectral images for building detection. *Information Fusion*, 6(4):283–300.

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

Sampath, A. and Shan, J. (2007). Building boundary tracing and regularization from airborne lidar point clouds. *Photogrammetric Engineering and Remote Sensing*, 73(7):805–812.

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

Schuster, H. F. and Weidner, U. (2003). A new approach towards quantitative quality evaluation of 3d building models. *In Proc. of the ISPRS Commission IV Joint Workshop on Challenges in Geospatial Analysis.*

Shufelt, J. A. (1999). Performance evaluation and analysis of monocular building extraction from aerial imagery. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 21(4):311–326.

Sohn, G., Huang, X., and Tao, V. (2008). Using a binary space partitioning tree for reconstructing polyhedral building models from airborne lidar data. *Photogrammetric Engineering and Remote Sensing*, 74(11):1425–1438.

Sohn, G., Jwa, Y., Jung, J., and Kim, H. (2012). An implicit regularization for 3d building rooftop modeling using airborne lidar data. *ISPRS Annals of Photogrammetry, Remote Sensing and Spatial Information Sciences*, I-3:305–310.

Sun, S. and Salvaggio, C. (2013). Aerial 3d building detection and modeling from airborne lidar point clouds. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 6(3):1440–1449.

Tarsha-Kurdi, F., Landes, T., and Grussenmeyer, P. (2007). Hough-transform and extended ransac algorithms for automatic detection of 3d building roof planes from lidar data. *International Archives of Photogrammetry, Remote Sensing and Spatial Information Systems*, 36:407–412.

Tarsha-Kurdi, F., Landes, T., and Grussenmeyer, P. (2008). Extended ransac algorithm for automatic detection of building roof planes from lidar data. *The photogrammetric journal of Finland*, 21(1):97–109.

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

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

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

Wang, J. and Shan, J. (2009). Segmentation of lidar point clouds for building extraction. *In Proc. of the Annual Conference of the American Society for Photogrammetry and Remote Sensing*, pages 9–13.

Wang, R. (2013). 3d building modeling using images and lidar: a review. *International Journal of Image and Data Fusion*, 4(4):273–292.

Wehr, A. and Lohr, U. (1999). Airborne laser scanning - an introduction and overview. *ISPRS Journal of Photogrammetry and Remote Sensing*, 54(2):68–82.

You, S., Hu, J., Neumann, U., and Fox, P. (2003). Urban site modeling from lidar. *In Proc. of the International Conference on Computational Science and Its Applications (ICCSA)*, pages pp. 579–588.

Zhang, K., Chen, S. C., Whitman, D., Shyu, M. L., Yan, J., and Zhang, C. (2003). A progressive morphological filter for removing nonground measurements from airborne lidar data. *IEEE Transactions on Geoscience and Remote Sensing*, 41(4):872–882.

Zhang, K., Yan, J., and Chen, S. C. (2006). Automatic construction of building footprints from airborne lidar data. *IEEE Transactions on Geoscience and Remote Sensing*, 44(9):2523–2533.

Zhang, W., Chen, Y., Yan, K., Yan, G., and Zhou, G. (2012). Primitive-based 3d building reconstruction method tested by reference airborne data. *International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, 39(B3):373–378.

Zhou, Q. Y. and Neumann, U. (2008). Fast and extensible building modeling from airborne lidar data. *In Proc. of the 16th International Conference on Advances in Geographic Information Systems (ACM GIS)*.

Zhou, Q. Y. and Neumann, U. (2010). 2.5 d dual contouring: a robust approach to creating building models from aerial lidar point clouds. *In Proc. of the European Conference on Computer Vision (ECCV)*, pages pp. 115–128.