UCGE Reports
Number 20374

Department of Geomatics Engineering

Automatic 3D Building Model Generation by Integrating LiDAR and Aerial Images Using a Hybrid Approach

(URL: http://www.geomatics.ucalgary.ca/graduatetheses)

by

Eunju Kwak

JANUARY, 2013

UNIVERSITY OF CALGARY

Automatic 3D Building Model Generation by Integrating LiDAR and Aerial Images Using a Hybrid Approach

by

Eunju Kwak

A THESIS

SUBMITTED TO THE FACULTY OF GRADUATE STUDIES

IN PARTIAL FULFILMENT OF THE REQUIREMENTS FOR THE

DEGREE OF DOCTOR OF PHILOSOPHY

DEPARTMENT OF GEOMATICS ENGINEERING

CALGARY, ALBERTA

JANUARY, 2013

Abstract

The development of sensor technologies and the increase in user requirements have resulted in many different approaches for efficient building model generation. Three-dimensional building models are important in various applications, such as disaster management and urban planning. Despite this importance, generation of these models lacks economical and reliable techniques which take advantage of the available multi-sensory data from single and multiple platforms. Therefore, this research develops a framework for fully-automated building model generation by integrating data-driven and model-driven methods as well as exploiting the advantages of images and LiDAR datasets.

The building model generation starts by employing LiDAR data for building detection and approximate boundary determination. The generated building boundaries are then integrated into a model-based image processing strategy, because LiDAR derived planes show irregular boundaries due to the nature of LiDAR point acquisition. The focus of the research is generating models for the buildings with right-angled-corners, which can be described with a collection of rectangles (e.g., L-shape, T-shape, U-shape, gable roofs, and more complex building shapes which are combinations of the aforementioned shapes), under the assumption that the majority of the buildings in urban areas belong to this category. Therefore, by applying the Minimum Bounding Rectangle (MBR) algorithm recursively, the LiDAR boundaries are decomposed into sets of rectangles for further processing. At the same time the quality of the MBRs are examined to verify that the buildings, from which the boundaries are generated, are buildings with right-angled-corners. These rectangles are preliminary model primitives. The parameters that define the model primitives are adjusted using detected edges in the imagery through the least-squares adjustment procedure, i.e., model-based image fitting. The level of detail in the final Digital Building Model is based on the number of recursions during the MBR processing, which in turn are determined by the LiDAR point density. The model-based image fitting refines the search space and resolves the matching ambiguities in multiple images, which results in higher quality boundaries. This research thus develops an approach which not only automates the building model generation, but also improves the accuracy of the building model itself.

Acknowledgements

It would be difficult to overstate my gratitude to my supervisor Dr. Ayman Habib for his continuous support and valuable advice during my PhD. He is a great academic role model due to his enthusiasm and generosity. This thesis would have been impossible without his guidance.

I would like to express my sincere gratitude to my committee, Dr. Derek Lichti, Dr. Naser El-Sheimy, Dr. Mryka Hall-Beyer, and Dr. Costas Armenakis for their assistance and valuable feedback on this thesis.

I would like to acknowledge the scholarship support I received during my studies from the University of Calgary and Northwest Geomatics, and also GEOmatics for Informed DEcisions (GEOIDE) for the support on this research work.

I am indebted to my colleagues in the Digital Photogrammetry Research Group (DPRG) with whom I shared unforgettable moments. I am also very grateful to my special friends in the Geomatics Engineering Department. Special thanks to the members of trailer H, who dragged me to social events and helped me to get rid of the panic and stresses of academia. Many thanks to my Korean friends that I met in Calgary who caringly supported me through the difficult times and shared my successes. I cannot forget to thank to my friends in Korea who provided me with emotional support even though they were far away. There are too many people that I need to thank who cannot be listed here, but I will never forget their unconditional love and help.

Finally, yet most importantly, I would like to express my heartfelt thanks to my beloved parents, Leedo Kwak and Hyesuk Sohn, for their love, trust and support for me. I would like to also thank my two brothers, my grandma and grandpa, who always believed in me and loved me.

Thanks to God for your wisdom, strength and guidance. He provided me all the necessary tools that I needed during the hardships.

To God

And to my beloved parents

Leedo Kwak and Hyesuk Sohn

Table of Contents

List of Tables

List of Figures and Illustrations

List of Abbreviations

Abbreviations	Definition
CSG	Constructive Solid Geometry
DBM	Digital Building Models
DEM	Digital Elevation Models
DTM	Digital Terrain Models
EOP	Exterior Orientation Parameters
EuroSDR	European Organization on Spatial Data Research
FN	False Negative
FP	False Positive
GPS	Global Positioning System
GSD	Ground Sampling Distance
INS	Inertial Navigation System
IOP	Interior Orientation Parameters
LiDAR	Light Detection and Ranging
LoD	Levels of Detail
LPD	Local Point Density
LSA	Least-squares adjustment

MBR	Minimum Bounding Rectangle
OEEPE	European Organization for Experimental Photogrammetric Research
QA	Quality Assurance
QC	Quality Control
RMBR	Recursive Minimum Bounding Rectangle
RMSE	Root Mean Squared Error
TN	True Negative
TP	True Positive

Chapter One: **Introduction**

1.1 Motivation

Accurate and up-to-date 3D geo-spatial information, including building models, is quite valuable for several applications. A few examples where accurate digital building models are useful include the following: evaluation of the aftermath of natural disasters, search and rescue, urban planning, environmental studies, telecommunication network planning, 3D surveillance, and simulation for urban terrorism. Considering that the United Nations reported that more than 70% of the global population by 2030 will dwell in urban areas, up-to-date and accurate 3D city models are required to properly plan for and accommodate this urban growth (UN-Habitat, 2012). A survey pertaining to 3D city models, conducted by the European Organization for Experimental Photogrammetric Research (OEEPE), currently the European Organization on Spatial Data Research (EuroSDR), also confirmed the importance of 3D city modeling, especially building models. Ninety-five percent of the people who participated in the survey stated they consider buildings as the most important objects of interest within a city model (Fuchs et al., 1998). Rapid urbanization causes social and environmental problems in developing countries; and access to accurate and current 3D Digital Building Models (DBM), which can be derived in a timely manner and at a reasonable cost, is essential in the planning process. In addition, with more accessible location-based and personal navigation services now available to the general public, the need for automated, realistic, and efficiently-generated 3D models has become more urgent than ever (Brenner, 2005).

Many research efforts have addressed this criticality for automatic and efficient building model generation. In particular, the development of sensor technologies and the increase in user requirements have resulted in many different approaches for efficient building model generation (Rottensteiner et al., 2005; Sohn et al., 2008; Kim and Habib, 2009; Huang et al., 2011). These approaches can be categorized according to the data source used (single vs. multi-source approaches), the data processing strategy (data-driven, model-driven, or hybrid), and the amount of human interaction (manual, semiautomatic, or fully automated) (Vosselman and Maas, 2010).

In terms of the data source, aerial imagery has been, and still is, one of the most common sources to obtain 3D building models based on traditional photogrammetric approaches. While aerial imagery provides reliable results, the low degree of automation during the matching process is its main limitation, especially when occlusions are present (Brenner, 2005). Partial or complete occlusions are common problems for images over urban areas. Thus, most of the current techniques have mainly relied on semi-automatic systems (i.e., recognition and interpretation tasks are performed by an operator) while the modeling and precise measurement tasks are supported through automation. Meanwhile, Light Detection and Ranging (LiDAR) data have emerged as an important source of data for 3D building model generation due to the availability of dense 3D point clouds. While the LiDAR system has the significant advantage of direct acquisition of the 3D coordinates of points (thus, eliminating the need for the matching procedure), the positional nature of LiDAR data collection makes it difficult to derive semantic information from the captured surfaces. Moreover, the quality of the derived boundaries is affected by the point density, which makes LiDAR data insufficient as a stand-alone source.

Due to the limitations of using a single data source, integration of multi-source data has been recommended since it preserves the advantages of the involved datasets (Chen et al., 2004; Rottensteiner et al., 2005; Kim and Habib, 2009; Demir and Baltsavias, 2012). The integration of LiDAR data and aerial imagery, which are the most common sources for building model generation, provides more accurate solutions by combining the accurate height measurements of laser scanners and the planimetric accuracy of aerial images (Awrangjeb et al., 2010). However, despite the progress that has been made in the area of integrating LiDAR data and imagery, there is still a great deal of work to be done; for example, automated processing of the available datasets is still at an early research stage.

In terms of processing strategies for building reconstruction, the most used are data-driven and model-driven approaches (Faig and Widmer, 2000; Vosselman and Maas, 2010). These approaches differ on how much relevant information regarding buildings is incorporated during each process. Data-driven approaches, which are also called bottom-up processes, do not make assumptions regarding the building shapes. Theoretically, one can model any shape of building using data-driven approaches; but, practically, these approaches do not address the question of how to impose constraints and set the rules during the generation process (Brenner, 2005). Due to the complexity of the implementation of the data-driven approaches, model-based approaches (i.e., top-down processes), have been the focus of many researchers. Model-based approaches use the opposite strategy of their data-driven counterparts by assuming and predefining parametric models for the sought-after buildings in advance and update them using the information derived from existing data. The predefined parametric models are referred to as model primitives and they are associated with translation, scale, and rotation parameters.

Complex building models can be constructed by combining small sets of model primitives depending on the desired level of detail (Suveg and Vosselman, 2004; Xie et al., 2009). While robust computation is possible, establishing the representative models requires human interaction (Tseng and Wang, 2003). More specifically, selection of a model primitive and its approximate alignment with the existing data are performed by an operator. Therefore, the mixture of data-driven and model-driven methods (i.e., hybrid approach), which follows the process of hypothesize-and-test, is suggested (Faig and Widmer, 2000; Vosselman and Maas, 2010). Faig and Widmer (2000) briefly described a hybrid approach for building extraction using aerial images. It combines the flexibility of the bottom-up approach and the robustness of the top-down approach.

An increasing number of 3D building models have been generated using different methodologies during the last decades. All models are associated with a specific level of detail. The required levels of detail for the resulting models are different, from simple building blocks to detailed façade, depending on their applications. For example, simple block models without unnecessary details can be produced for navigational purposes, especially on mobile devices. However, in large-scale applications that require cultural heritage building documentation, a high level of detail is required. In addition, in some applications, especially in terms of visualization, there is no single representation which suits all needs. Consequently, a range of level of detail is necessary; and producing multiple representations of a building with different levels of detail therefore has been an active research topic. While some researchers have aimed to produce models that are as detailed and realistic as possible, other researchers have tried to simplify and generalize the existing models to coarse representation (i.e., DBM generalization) (Kolbe et al., 2005; Arefi et al., 2008; Guercke et al., 2011).

The Special Interest Group 3D (SIG 3D) presented the standard definition regarding the levels of detail (LoD). Five different LoD are supported, ranging from simple block models to detailed models with interiors. Kolbe et al. (2005) explained the characteristics and accuracies of each level as follows: LoD0 is a 2.5D Digital Terrain Model (DTM); LoD1 is a simple block model; LoD2 includes roof structures and textures; LoD3 is an architectural model; and, lastly, LoD4 includes highly accurate interior structures. Interested readers can refer to Kolbe et al. (2005) and Gröger et al. (2012) for more detailed information about each level.

Since many models can be produced using different approaches from different data, the quality required of the building model needs to be specified to decide if the generated model can be used for certain applications. For instance, many commercial applications are good for navigation purposes, but their level of accuracy is not high enough for design and engineering applications. It should be noted that the aforementioned LoD categories cannot represent the geometric accuracy of the final model so a standard method of evaluating geometric accuracy is crucial. In the current research literature, evaluation results are often missing from the building models. Furthermore, many studies utilized only one or two evaluation indices, such as completeness and correctness, without considering the stochastic properties of the generated DBMs (Vu et al., 2009; Demir and Baltsavias, 2012). Most of the work so far lacks a standard method of assessing the accuracy of a model and cannot determine the reliability of the results. While assessing the quality of an end product is important, at the same time, if the performance of the processing steps and assumptions made can be evaluated at each stage, the amount and the time of human intervention can be minimized. A clear indication of the performance of the approach used will increase the productivity of 3D building model generation.

1.2 Research objectives

As discussed in the previous section, it is obvious that 3D building models are important components, but they still lack economical and reliable techniques for their generation while taking advantage of the available multi-sensory data from single and multiple platforms. The objective of this research work is to establish a framework for fully-automatic building model generation by integrating data-driven and model-driven methods while combining the advantages of images and LiDAR datasets. First, LiDAR data are employed for building detection and approximate boundary determination. Then, the building boundaries generated from LiDAR are integrated into a model-based image processing strategy. The focus of the reconstruction is on complex structures, which are comprised of a collection of rectangles (e.g., L-shape, T-shape, U-shape, gable roofs, and more complex building shapes which are combinations of the aforementioned shapes) under the assumption that the majority of the buildings in urban areas belong to this category. Combinations of rectangles can produce LoD1 and LoD2 with a high level of automation. Figure 1.1 shows the proposed framework for the automatic building model generation starting from the quality assurance/quality control (QA/QC) of the involved dataset. The buildings are detected from LiDAR data automatically, and the building detection performance is evaluated. The building boundaries are decomposed into sets of model primitives and the quality of the primitives is analysed to decide whether or not the proposed approach can reconstruct the buildings in question. Finally, the derived primitives are refined through model-based image fitting, which produces the stochastic properties of the estimated parameters.

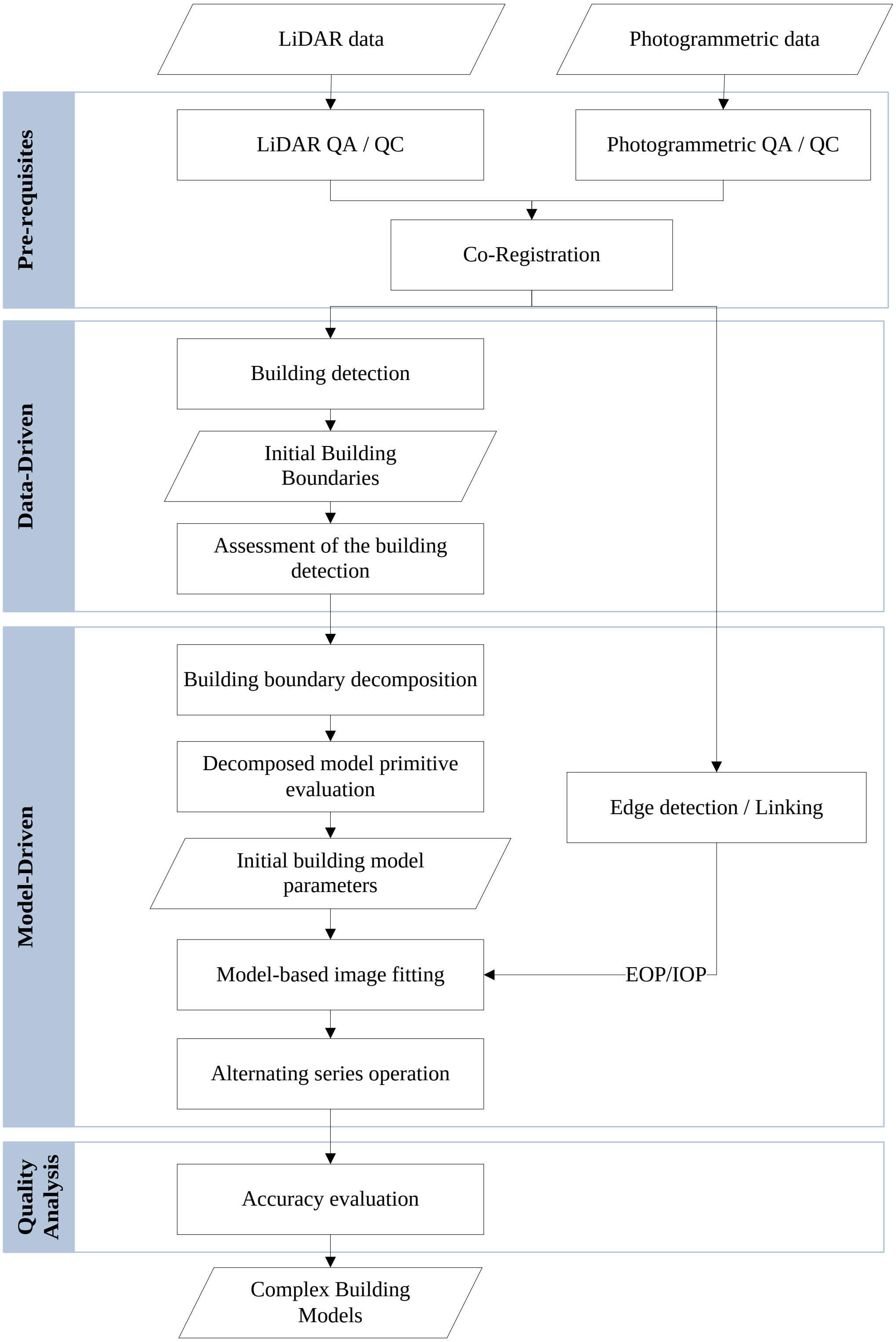


Figure 1.1: Flowchart of the proposed methodology

The objectives of this research work can be summarized as follows:

- Propose a hybrid methodology (i.e., combination of data-driven and model-driven approaches) for automatic building model generation.
- Achieve a high level of automation:
  - Reduce the number of model primitives while being able to model complex buildings automatically.
  - Derive the initial model parameters automatically, thus eliminating the need for human intervention as opposed to what was observed in previous studies.
  - Determine whether or not buildings can be reconstructed automatically using the proposed approach.
- Obtain the model with a high level of accuracy:
  - Increase the accuracy of the final model by combining LiDAR data and imagery.
    - Pursue better horizontal boundary accuracy from photogrammetric images.
    - Maintain the good vertical accuracy from the LiDAR point cloud.
  - Analyze the quality of each processing strategy (i.e., building detection rate from LiDAR data, evaluation of the decomposed models, and geometric accuracy of the final result).
- Generate different levels of details:
  - Aim to produce buildings with details equivalent to LoD2.
  - Generalize the model to LoD1 depending on the application.
- Utilize any number of images and overcome the limitation of partial occlusions in the imagery, which is the most common problem for image-based approaches.

1.3 Thesis outline

This thesis proposes a fully-automatic building model generation method by integrating data-driven and model-driven methods according to the flowchart in Figure 1.1. Each chapter can be summarized as follows:

- Chapter 2 explains a typical process of building extraction and presents several criteria to classify building reconstruction approaches. The literature on existing DBM generation methods is reviewed and different quality evaluation methodologies are also addressed.
- Chapter 3 discusses automatic detection of possible buildings from airborne LiDAR data. The LiDAR-generated building boundaries are good indications of where the buildings are located in the dataset, but the irregular boundaries need to be improved.
- Chapter 4 presents the methodology of automatic LiDAR building boundary decomposition into basic model primitives. The derived model primitives are evaluated to decide whether the buildings are regular in shape or not.
- Chapter 5 introduces the building reconstruction methodology based on model-based image fitting to improve the positional accuracy of building boundaries. The definition of model parameters, edge detection procedures, and how to apply the model-based image fitting sequentially are presented.
- Chapter 6 presents experimental results to demonstrate and evaluate the performance of the proposed methodologies using airborne LiDAR and photogrammetric data.
- Chapter 7 provides the conclusion for the presented research and recommendations for future work.

Chapter Two: **Literature review**

2.1 Introduction

The earliest research effort to automatically reconstruct 3D building models can be traced back to not quite two decades ago (Gruen et al., 1995; Haala, 1995). Numerous works have been published about reconstruction methods using various approaches. As mentioned in Chapter 1, the existing building reconstruction approaches can be classified according to the data sources they use (single vs. multi-source approaches); the processing strategy (data-driven, model-driven, or hybrid); and the amount of user interaction (manual, semiautomatic, or fully-automated) (Vosselman and Maas, 2010). In this chapter, the existing literature is discussed in detail, as categorized above, and each approach's advantages and limitations are defined.

2.2 Existing DBM generation methods in terms of data source: Single vs. multi-source approaches

Previous studies, depending on their data sources for building extraction, can be categorized as follows: 1) approaches using only imagery (monocular, stereo, or multiple images); 2) approaches using only LiDAR data; and 3) approaches that integrate multiple sensors.

Reconstructed surfaces using photogrammetric measurements contain rich semantic information and dense positional information along break lines. Therefore, diverse research studies using images to generate DBMs have been implemented. The early stage of the previous work started with monocular imagery (Huertas and Nevatia, 1988; McGlone and Shufelt, 1994). To acquire

3D building information from a single image, walls and shadows are utilized together with the direction of the sun and the camera geometry (Lin and Nevatia, 1998). The main problem of this approach is that the result is dependent on the quality of the detected edges, and the use of shadow information does not produce reliable results. Therefore, stereo/multiple images have been one of the most popular data types for three dimensional reconstructions. 3D features (i.e., lines and planes), which are derived after the matching of conjugate features, are merged and grown to generate building information (Roux and McKeown, 1994; Fischer et al., 1998; Baillard et al., 1999; Kim and Nevatia, 2004). Many of the researchers focused on the improvement of the matching quality and solutions for the occlusion problems. Nonetheless, the automation of the matching problem remains a difficult and unreliable task due to many obstacles, such as relief displacement, different illumination conditions, and the shadow effect when dealing with large scale imagery over urban areas.

LiDAR has become an important source of data for the generation of 3D city models due to the direct acquisition of a reliable and dense 3D point cloud in contrast to the problematic matching process when using images. While LiDAR has the great advantage of direct acquisition of the 3D coordinates of points, the positional nature of LiDAR data collection makes it difficult to derive semantic information from the captured surfaces. The measured points from the LiDAR system do not provide enough information to extract the accurate boundaries of building patches. The probability that the LiDAR points lie on the exact edges of a physical shape is very low due to the discrete and semi-random/irregular nature of such points. Hence, LiDAR data have been known to be an insufficient data source to accurately extract building boundaries. The geometric assumptions of building shapes, such as parallelism, symmetry, flat roof, and straightness of boundaries, have been used to acquire building information, while alleviating the irregular nature of LiDAR points (Weidner and Förstner, 1995; Maas and Vosselman, 1999). However, the research based on these assumptions seriously simplifies the real world without considering the various shapes of man-made structures.

Due to the limitations of using single source data, several researchers have attempted to integrate different data sources. Suveg and Vosselman (2004) generated a 3D model from aerial images by incorporating a 2D GIS ground plan (from a GIS map) as a building hypothesis. The ground plan was used to decompose complex buildings into multiple rectangular parts. The integration of GIS data with LiDAR data was explored by Haala et al.(1998) and Vosselman and Dijkman (2001). Although GIS data can provide additional information that can reduce the search space in the reconstruction, the data could be outdated, lacking information, generalized, or mis-aligned with other data. Throughout many research studies using images and LiDAR data, it is often stated that the photogrammetric and LiDAR technologies have complementary characteristics. The integration of LiDAR data and aerial imagery for DBM generation has gained significant attention by the research community (Rottensteiner and Jansa, 2002; Chen et al., 2004; Ma, 2004; Brenner, 2005; Lee et al., 2008; Awrangjeb et al., 2010; Habib et al., 2011; Hermosilla et al., 2011). Ma (2004) utilized LiDAR data to extract features and to reconstruct initial building models. The initial building models were projected back to aerial images and refined using the edge pixels extracted from the images. Chen et al. (2004) detected building regions by integrating a region-based segmentation and a knowledge-based classification from both LiDAR data and color aerial images. They considered the characteristics of elevation and roughness from LiDAR data and the spectral and texture information from aerial images. To reconstruct the building models, the split-merge-shape method was applied using planar patches from LiDAR data and extracted edges from images. However, this method cannot detect fine levels of details and does not show how to deal with erroneous lines. Awrangjeb et al. (2010) proposed an automatic building detection method using LiDAR data and color orthoimagery. First, the initial building models were determined using the line segments of building masks generated from the LiDAR data and the Normalized Difference Vegetation Index (NDVI) values from the imagery. The complete building models were obtained using the LiDAR masks and YIQ color information. This research showed a high success rate in smooth terrain areas in the absence of buildings of rapidly varying heights. However, it yielded relatively poor geometric positional accuracy and was thereby insufficient for mapping purposes.

Much of the previous work has focused on reconstruction of simple buildings, and DBM generation of complex buildings is still a challenging issue. One of the popular integration approaches is to use LiDAR data as a primary source for building detection and employ the imagery as an additional source. Habib et al. (2011) incorporated stereo aerial images after deriving initial boundaries from LiDAR data. In this case, the quality of the results was dependent on the initial LiDAR boundary result and only two images could be utilized. Therefore, how to integrate the two data sources in a way that their weaknesses can be compensated effectively is an active issue of current research.

2.3 Existing DBM generation methods in terms of reconstruction strategy: Data-driven, model-driven, and hybrid approach

The reconstruction of buildings can be carried out using three different approaches: The data-driven method (bottom-up), the model-driven method (top-down), or the hybrid method (Mayer, 1999; Vosselman and Maas, 2010). The differences between the aforementioned three approaches are manifested in how much relevant information about the buildings is implemented during each process.

Data-driven approaches do not make assumptions regarding building shapes, in most cases using generic models. They have the advantage of being able to model any shape of building, but at the same time the limited knowledge of the models makes its implementation complex. Therefore, it leaves the question of how constraints and sets of rules should be imposed on rectangularity, parallelism, etc. (Brenner, 2005). Henricsson and Baltsavias (1997) presented an example of the data-driven approach using aerial imagery. First, they detected the region of interest manually from aerial imagery and derived the 3D lines by matching the extracted 2D lines. Their final polyhedral model was reconstructed by using the co-planarity constraint and color information. This method produces good results but cannot reconstruct buildings in densely built-up areas and is not fully-automated. One of the common data-driven approaches utilizes segmentation of planar rooftops on LiDAR data. Sampath and Shan (2010) reconstructed polyhedral building roofs using LiDAR data based on segmentation. After the boundaries were established, regularization was attained with multiple topological constraints (i.e., parallel and perpendicular line pairs). Kim and Habib (2009) generated complex polyhedral building models by integrating photogrammetric and LiDAR data based on the data-driven approach. LiDAR data were utilized to generate the building hypotheses and initial boundaries while stereo aerial images were used to extract straight lines based on the initial LiDAR boundary information. Like other data-driven methods, the extracted lines were subjected to a matching process with additional constraints for building shapes.

The model-based approach uses parametric building models which are used as a hypothesis and verifies the model using information derived from the existing data. The majority of buildings in rural areas can be approximated by parameterized standard roof shapes. Therefore, the model-based approach predefines different basic building models in a database, from which the best fit model to the data is selected (Suveg and Vosselman, 2004; Xie et al., 2009). The reconstruction of complex buildings is reduced into the construction of the basic primitives and then they are combined afterwards. Therefore, they are decomposed into the basic roof shapes (Kada and McKinley, 2009). Model-based methods are commonly implemented in a semi-automatic manner and use information from images. While the model-image fitting is solved automatically, the selection of a target model and an initial alignment of the model to the image are carried out manually by an operator. Various existing model-based approaches can be reviewed based on the categories of employed models and reconstruction strategies. Vosselman and Veldhuis (1999) presented a semi-automatic model-based method using Constructive Solid Geometry (CSG) models. The selection of a model and the approximate alignment were carried out by a human operator and precise alignment of the model with the image was done by automatic fitting using grey value gradients. Tseng and Wang (2003) established a practical approach for model-based building extraction using aerial images. They tackled the two key issues of model-based building reconstruction: how to establish a set of representative models and how to develop an algorithm to adjust the model using imagery. The models were selected and positioned interactively by a human operator and the least-squares model-image fitting algorithm proposed by Lowe (1991) was used to adjust the model pose and shape parameters. The model results demonstrated decent horizontal accuracy, but the vertical accuracy is low because of the small base-height ratio of the utilized stereo images. Also, the fitting process exhibited poor results if the bottom edges of the building were unclear or occluded.

The mixture of data-driven and model-driven methods which follow the process of hypothesize-and-test is called the hybrid approach (Faig and Widmer, 2000; Vosselman and Maas, 2010). Faig and Widmer (2000) briefly described a hybrid approach for building extraction using aerial images, which combined the flexibility of the bottom-up approach and the robustness of the top-down approach. The hybrid approach could be one of the solutions to automate building model generation, thereby reducing the numerous interpretation problems of data-driven methods and the dependence on operators in model-based methods.

2.4 Existing DBM generation methods in terms of degree of automation: Semi-automatic or fully-automatic

Fully-automated building model generation is certainly one of the ultimate goals in the photogrammetric and computer vision community. Many fully-automated approaches have been proposed in the past decade (Karner et al., 2006; Lafarge et al., 2008; Zebedin et al., 2008), but all of them are more or less practical only for some special cases. Lafarge et al. (2008) presented an automatic building extraction method from Digital Elevation Models (DEMs) acquired from satellite images based on an object approach. This approach yielded satisfactory results considering that satellite images were employed and a high level of automation was used. However, the roof shapes were constrained to only symmetric two-planes (i.e., gable roof) because they used the median axis of the connected polygon set as the rooftops, which does not allow for the representation of some details. Zebedin et al. (2008) proposed an automatic building reconstruction method by fusing feature- and area-based information from aerial images. The 3D line features detected by matching multiple images were used to generate segments of buildings, which were combined with a set of geometric primitives obtained from the height data and a building mask. In order to take advantage of the available information and to generate a detailed building model, a global optimization algorithm based on Graph Cuts was used during the fusion process. This approach is able to reconstruct roof shape with planes and domes and different levels of details. Yet, it left the investigation of gable roofs as a future work.

Developing a fully-automated process for the generic model is still a challenging task for many reasons. The task of building extraction depends on the data type, scale used, the complexity of the object, and the required level of detail. The applied building detection algorithms to date tend to fail whenever a new situation is encountered. Semi-automated approaches therefore are currently attracting more attention with the need for precise, reliable, and complete 3D data.

The semi-automated approach generally lets humans decide which building model should be used and the computer performs the model-image fitting automatically (Suveg and Vosselman, 2000; Tseng and Wang, 2003). Humans can perform high level tasks much more reliably than computers, and computers do low-level tasks faster than humans; and this cooperation could make semi-automated building extraction practically valuable. Human operations such as detecting the error and editing of the building models often slows down the whole building model generation process. However, if it is possible to show the performance of the utilized approach (i.e., success rate), the amount and the time of human intervention can be minimized. Kaartinen and Hyyppä (2006) evaluated the quality, accuracy, and feasibility of various building extraction techniques utilizing reference points and raster ground plans. Reference points are used to analyze the accuracy of the location, length, and roof inclination of the modeled buildings. Raster data can be used to compute the total relative building area and total relative shape dissimilarity (Henricsson and Baltsavias, 1997). Gruber et al. (2008) evaluated the geometric accuracy of 3D building models using an error matrix and a statistical analysis of the height differences. Awranjeb et al. (2010) developed a performance evaluation system using 15 evaluation indices in three categories (i.e., object-based evaluation, pixel-based evaluation, and geometric evaluation). However, evaluation results are often missing from the published work of building detection; and many studies only used one or two evaluation indices, such as completeness and correctness, without considering the stochastic properties of the generated DBMs (Demir and Baltsavias, 2012; Vu et al., 2009). Elberink and Vosselman (2011) focused on the internal quality of 3D building models when independent reference data are missing. They assessed the quality of the input data, which affected the final accuracy, and analyzed the correctness of the assumptions made during the processing.

Most of the work thus far lacks a standard method for assessing the accuracy and reliability of the results. An appropriate performance evaluation system during the processing could increase the productivity of 3D model generation by clearly indicating the performance of the used approach. Table 2.1 summarizes the different building modeling approaches according to the above mentioned three categories.

Table 2.1: Summary of different building modeling approaches according to data source, reconstruction strategy, and level of automation

	Data source	Authors	Reconstruction strategy	Level of Automation
1	Aerial images	(Gruen and Wang, 1998)	Data-driven	Semi-automatic
2	Aerial images	(Baillard et al., 1999)	Data-driven	Automatic for polyhedral model with constraints
3	Aerial images/ GIS map	(Suveg and Vosselman, 2000)	Model-driven	Semi-automatic (manual partitioning into building parts)
4	LiDAR	(Brenner et al., 2001)	Data-driven	Automatic (required: DSM and set of rules)
5	Aerial images	(Tseng and Wang, 2003)	Model-driven	Manual selection of building and alignment (interactive); Automatic model-image fitting
6	Aerial images/ cadastral map	(Flamanc et al., 2003)	Hybrid	Semi-automatic
7	DEM from satellite images	(Lafarge et al., 2008)	Model-driven	Automatic with constraints
8	Aerial images	(Zebedin et al., 2008)	Data-driven	Automatic with height, building mask, 3D line input (acquired from image matching)
9	LiDAR/aerial images	(Kim and Habib, 2009)	Data-driven	Automatic for polyhedral model
10	Aerial images	(Hammoudi and Dornaika, 2011)	Model-driven	Automatic for polyhedral model

Chapter Three: **Building detection using LiDAR data**

3.1 Introduction

As mentioned in Chapter 1, LiDAR systems have been extensively utilized in generating building models due to their rapid acquisition of dense point clouds. The original LiDAR point cloud does not convey semantic information about the nature of the surfaces (i.e., one cannot tell which points belong to buildings, vegetation, roads, etc.). Therefore, the data needs to undergo a processing stage to extract useful information, buildings in this case. To automatically extract buildings from LiDAR data, buildings need to be separated from other objects such as ground and vegetation, and this task is referred to as building detection process (Vosselman and Maas, 2010). Given the fact that buildings are above ground and are mainly comprised of planar rooftops, this preliminary information can be incorporated to automatically discern buildings from other objects. Therefore, in the present research, LiDAR data segmentation and ground/non-ground classification procedures are performed to differentiate buildings from ground and non-planar objects.

Segmentation is a process of grouping the points that share similar properties according to some criteria (i.e., points that belong to the same plane in this case) (Vosselman et al., 2004). After the points that belong to planar objects are separated and aggregated from LiDAR data, the terrain and off-terrain points will be classified. Based on the results, building hypotheses will be generated and initial building boundaries for the detected buildings will be obtained, which will be further enhanced during the model-based image fitting process. Figure 3.1 illustrates the flowchart of the proposed LiDAR building detection procedure.

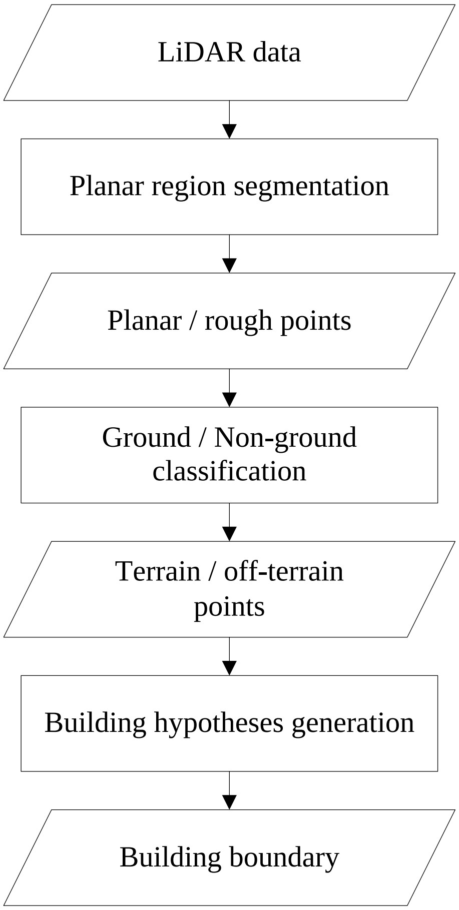

LiDAR data
Planar region segmentation
Planar / rough points
Ground / Non-ground classification
Terrain / off-terrain points
Building hypotheses generation
Building boundary

Figure 3.1: Flowchart of the building detection process

Section 3.2 will explain the planar region segmentation process, and Section 3.3 will illustrate the ground/non-ground classification. Lastly, Section 3.4 will discuss how to evaluate the building detection results.

3.2 Planar segmentation

The purpose of planar region segmentation is to group neighboring points that belong to the same plane. The most common segmentation techniques can be categorized into two approaches: 1) region growing in the spatial domain and 2) clustering in the parameter (attribute) space.

Region growing methods group adjacent points into segments based on the homogeneity of the surface defined by the grouped points. Several researchers utilized region growing methods to detect planar surfaces from LiDAR data. Region growing methods start from detecting a seed region and expand the seed region according to the closeness of the candidate points to the defined plane by the seed region (Lee and Schenk, 2001; Chen et al., 2004). The two main limitations of this approach are that the segmentation performance depends on the seed region selected and that the similarities of the attributes are compared only locally (Besl and Jain, 1988). While region growing is implemented in the spatial domain, the second approach, the clustering method, segments the points in the parameter space. These methods are dependent on the quality of the computed attributes; therefore, the segmentation attributes that best represent each surface should be identified. Filin and Pfeifer (2006) adopted clustering techniques using the slopes of the normal vector to the best fitting plane defined by a point and its neighboring points. Kim and Habib (2009) utilized the normal distances to the fitted planes from two pre-defined origins. However, the clustering technique requires substantial computation with the increase in the number of attributes and also does not consider the spatial-domain connectivity of the points.

Since each approach has its own advantages and limitations, many researchers have proposed different segmentation methodologies in an effort to overcome the aforementioned limitations (Besl and Jain, 1988; Vosselman and Dijkman, 2001; Filin and Pfeifer, 2006; Kim and Habib, 2009). However, most of the suggested approaches assume a uniform point distribution within the LiDAR data, which is not the case in reality. Depending on the type of acquisition platform (i.e., ground-based, airborne, or unmanned vehicle), the number of overlapping strips, and the properties of the surface, variations in the point density can be present within the dataset (Vosselman and Maas, 2010). Lari et al. (2011) proposed a new parameter-domain segmentation approach, which considers varying point densities in the dataset. In this research, the main focus will be buildings with planar rooftops which constitute the majority of buildings in a given environment. Therefore, the segmentation methodology proposed by Lari et al. (2011) will be employed to detect the planar objects that are potential buildings. The following paragraphs will discuss the proposed segmentation procedures in detail.

3.2.1 Neighborhood definition and local point density estimation

The proposed methodology starts with organising the LiDAR point cloud using a kd-tree to speed up the rest of the process (Friedman et al., 1977). Then, to calculate the local point density (LPD) for each point in the LiDAR data, a neighborhood definition follows. How the neighborhood of a point is defined affects not only the estimation of point density, but also its attributes. Therefore, an appropriate neighborhood definition that considers the proximity of points and the physical shape of objects is crucial. In this research, the adaptive cylinder neighborhood introduced by Filin and Pfeifer (2005, 2006) is adopted to define neighboring points. The adaptive cylinder is defined according to the orientation of the local planar surface, which can be determined through an iterative plane fitting of the neighboring points. More specifically, an initial fitted plane is derived through the least-squares adjustment (LSA) using the n-nearest points within a spherical neighborhood to the point of interest. The inverse of the normal distance between each point in the spherical neighborhood and the initial plane is used as a weight for the next iteration of the plane fitting, as illustrated in Figure 3.2. This procedure is repeated until the plane parameters do not change or a predefined number of iterations is reached. If the iterative plane fitting procedure does not converge within a pre-specified number of iterations, the point of interest is classified as part of a rough neighborhood. Once the final plane is derived through the iterative plane fitting, a cylinder can be defined by setting a buffer around the plane depending on the noise level in the LiDAR data. The points within the adaptive cylinder are considered as neighboring points of the point in question. If the majority of the points including the point in question/point of interest (i.e., more points than a pre-defined threshold) within the spherical neighborhood are inside the adaptive cylinder, the point of interest is deemed to be a part of a planar neighborhood.

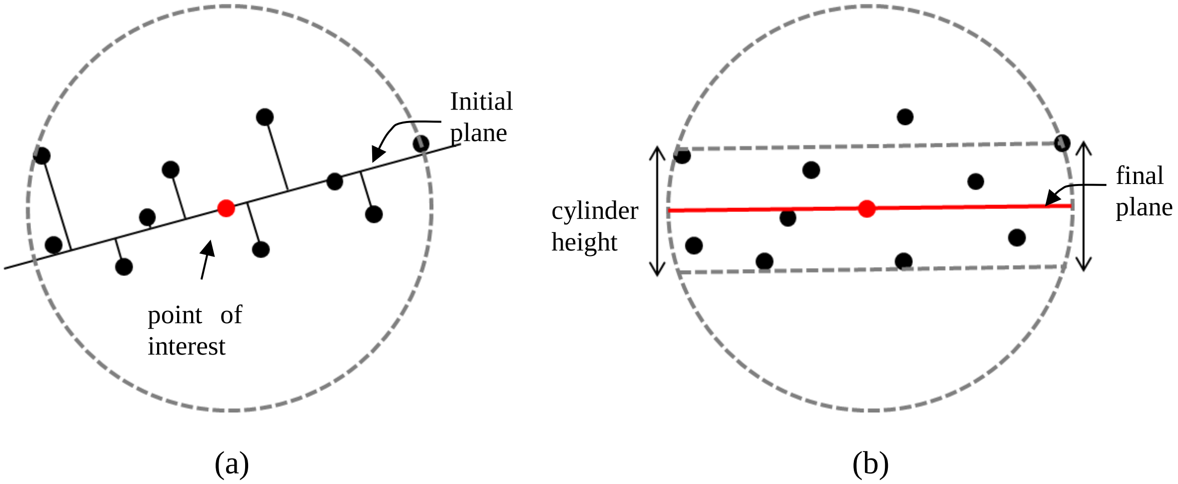


Figure 3.2: The initial fitted plane using neighboring points within the spherical neighborhood centered at the point of interest (a) and the final fitted plane derived through an iterative plane fitting procedure for the adaptive cylinder neighborhood establishment

Once the planarity of the established neighborhood is checked using the adaptive cylinder, the local point density can be calculated according to Equation (3.1). Only planar points are considered for the LPD estimation.

$$\mathrm{LPD}=\frac{\mathrm{k}}{\pi \mathrm{r}_{\mathrm{n}}^{2}} \qquad \textbf{(3.1)}$$

where, k is the number of neighboring points within the established adaptive cylinder and r_n is the distance from the point of interest to the n-nearest neighboring point (i.e., the radius of the spherical neighborhood used for the iterative plane fitting)

The radius r_n will be different depending on the average point spacing of the LiDAR dataset to include n-number of points, which will affect the calculation of LPD even if k is the same. Figure 3.3 shows that the radius of the spherical neighborhood containing n-nearest neighboring points is smaller when the average point spacing is smaller, which leads to higher local point density when compared to Figure 3.2(b). Figure 3.4 shows an example of the LiDAR point cloud colored according to height (a) with its local point density map for the points belonging to planar surfaces (b). 12 neighboring points are used to define the planar surface in this example.

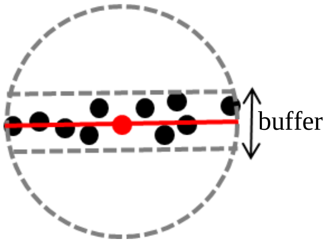


Figure 3.3: Adaptive cylinder neighborhood of dense point cloud

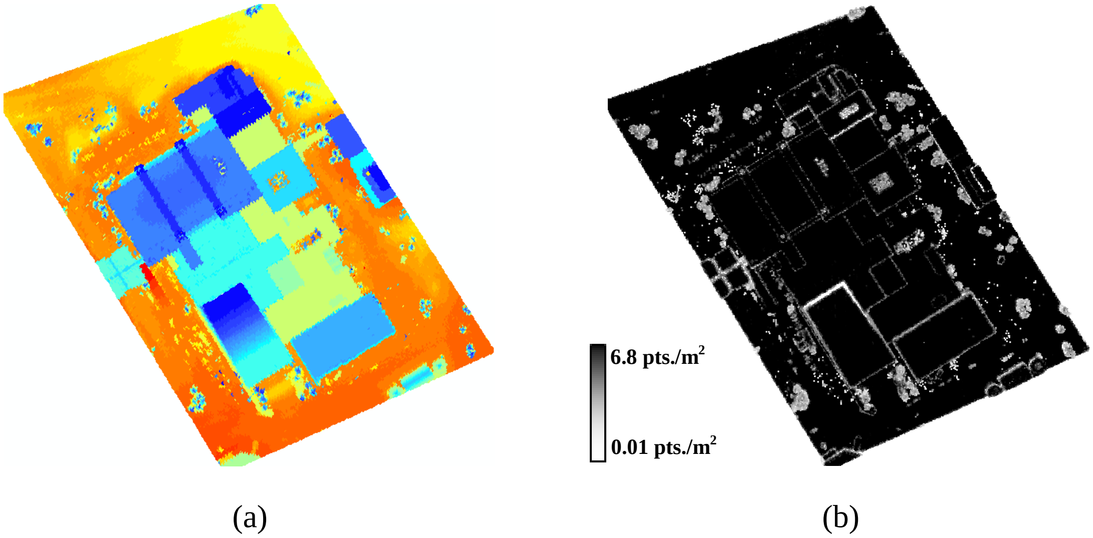
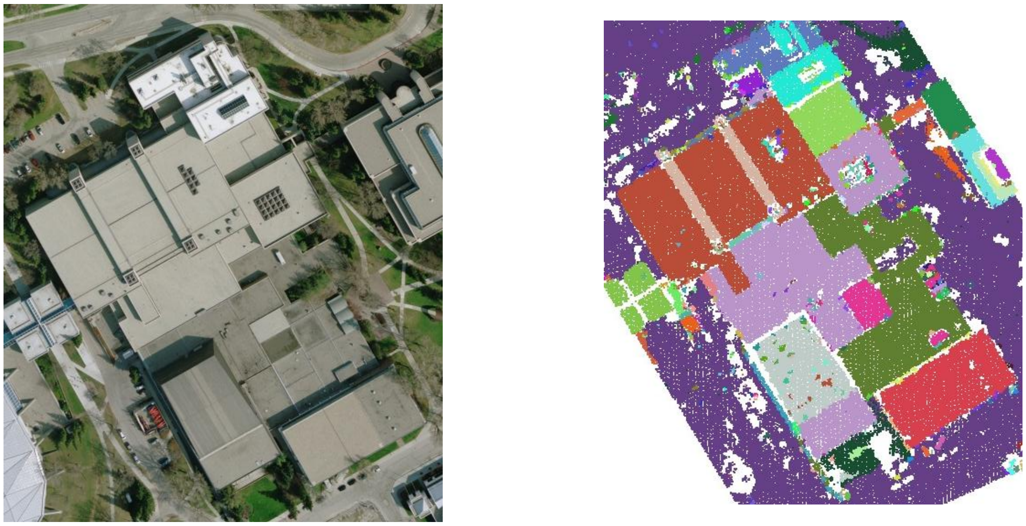


Figure 3.4: Original LiDAR point cloud colored according to height (a) and estimated local point density map (b)

3.2.2 Grouping and clustering

Once points are characterized if they belong to planar or non-planar objects and the local point density for each point is calculated, the planar points are grouped according to their 3D distance considering the local point density through a region growing process. Then, the clustering procedure will be carried out on the individual groups that have been classified as being part of the same planar surfaces. This will speed up the segmentation process compared to carrying out the segmentation process on the entire data all at once. The segmentation attributes are derived using the parameters of the best fitting plane for each point's neighborhood. The coordinates of the origin's projection onto the best fitting plane for each point's neighborhood (X_0, Y_0, Z_0) are used as attributes. The accumulated peaks in the attribute space based on these attributes represent the points with similar attributes (i.e., same origin's projection onto the best fitting plane) in the spatial domain. Therefore, these attributes are organized in a kd-tree structure to find a peak during the clustering approach without a need for a tessellation of an accumulator array, whose cell size affects the quality of the segmentation outcome. In this approach, the approximate extent of the clusters in the parameter domain is estimated considering acceptable spatial and angular deviations among points that would be deemed as part of a single planar region. Then, peak detection will be performed in two steps, coarse and fine detection, using the established kd-tree structure. First, an octree space partitioning (Samet, 1984) is utilized for the coarse peak detection. The space partitioning is repeated in the octant with the highest attribute count until the size of the octant reaches the approximate extent of the clusters. Then, using the points within the last octant, the precise peak location is detected through the second step. This step is incorporated to speed up the peak detection process. The precise peak location is determined as the one which includes the largest number of neighboring attributes within the established cluster extent. All the points in the detected peak are then removed from the parameter domain, and the same procedure is repeated until the number of remaining points is less than a predefined number which corresponds to the smallest region to detect. One should note that a reliable estimate of the plane parameters representing the plane surface through the points within the extracted peak can be derived through a least-squares adjustment. These plane parameters (i.e., a, b, c in $\mathrm{Z} = a\mathrm{X} + b\mathrm{Y} + c$), which are derived in this procedure, will be used in a model-based image fitting algorithm in Chapter 5. Figure 3.5(a) shows an example of airborne imagery over a building and Figure 3.5(b) shows the planar region segmentation results. Different colors represent different planar segments.

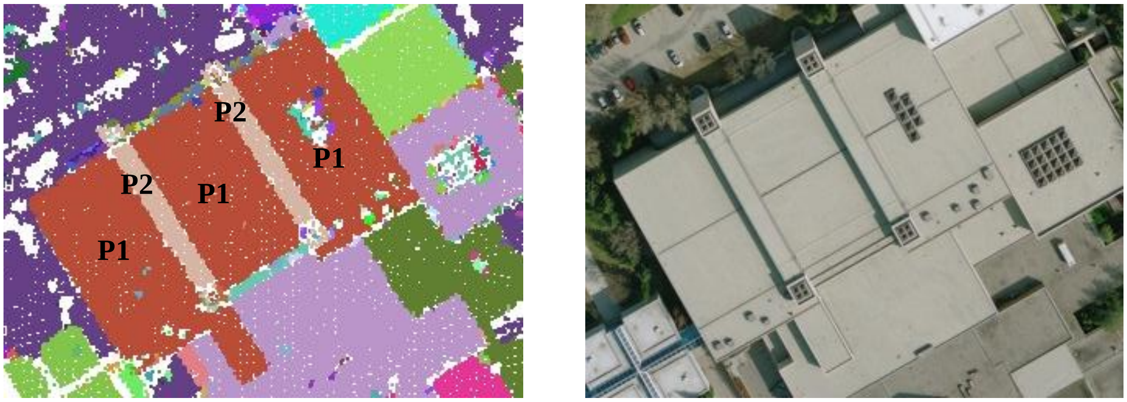


(a) (b)

Figure 3.5: Imagery over an area of interest (a) and planar region segmentation result (b)

3.2.3 Region boundary generation

As a result of the previous procedures, planar points that belong to the same plane are segmented. The main problem with parameter-domain segmentation procedures is that coplanar points that are spatially disconnected are segmented as one cluster. Figure 3.6 shows an example of this ambiguity problem. Figure 3.6(a) is a closer look at the planar region segmentation result presented in Figure 3.5(b), and one can observe that the spatially-disconnected coplanar surfaces are segmented as one cluster (three P1s are segmented as one and two P2s are segmented as one). Figure 3.6(b) shows the aerial image of the same region for reference.



(a)

(b)

Figure 3.6: The ambiguity problem of parameter-domain segmentation: planar region segmentation (a) and aerial image over the same planar surface (b)

In order to resolve such ambiguity, a neighborhood analysis is conducted through the boundary detection of the clustered points. To obtain the boundary of the segmented points, the modified convex hull method is applied (Sampath and Shan, 2007). This boundary tracing method starts by finding the most left point as a first point of the boundary. Then, it establishes a planar search space as a function of the local point density and selects all the points within the search space. The second point is chosen as the point that has the smallest clockwise angle between the positive y-axis and the vector from the first point to the target point. The remaining points along the boundary are determined by the ones which have the smallest clockwise angle with respect to the vector from the previous point to the current point. This process is repeated until the last point meets the first point which indicates the boundary tracing has been completed successfully. Figure 3.7 illustrates the aforementioned boundary tracing steps of the modified convex hull algorithm. As one can see in the last row, spatially-disconnected coplanar points can be separated during the boundary tracing process. When the distances between the points are larger than the threshold, which is predefined as a function of LPD, they are not considered as neighboring points. The neighborhood analysis through the boundary tracing resolves the ambiguity issues of parameter domain segmentation. Figure 3.8 presents an example of the generated building boundaries and segmentation results after the neighborhood analysis. Compared to the results presented in Figure 3.5(b), the spatially-disconnected coplanar regions are separated.

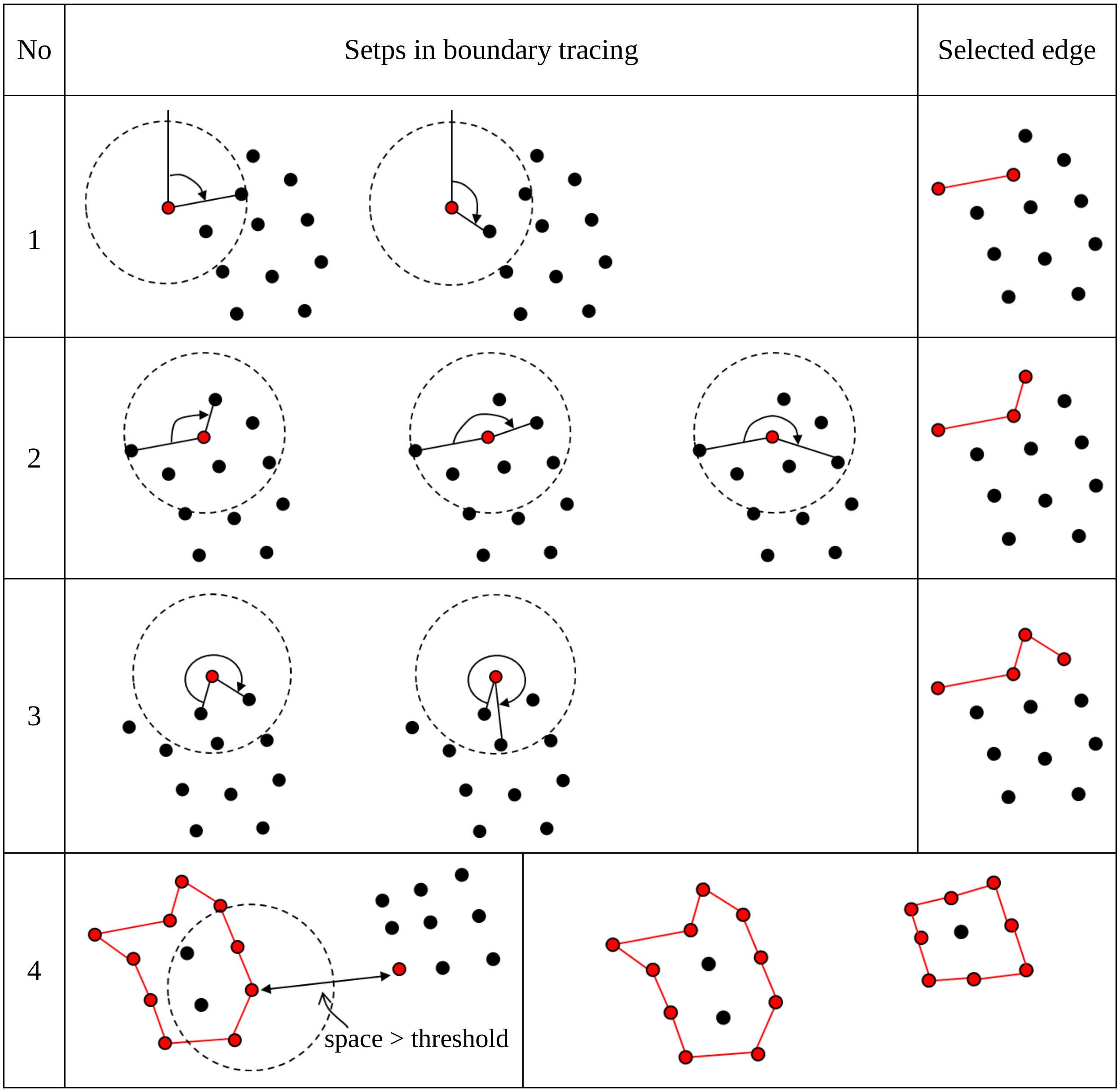
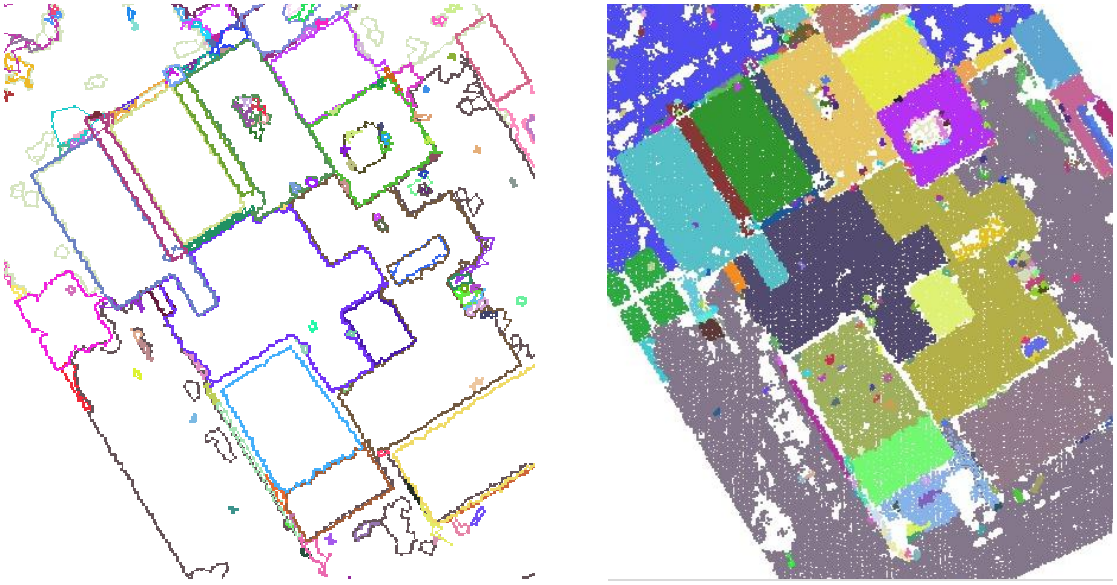


Figure 3.7: Steps of the modified convex hull algorithm (adapted from Sampath and Shan, 2007)



(a) (b)

Figure 3.8: An example of the traced boundaries (a) and accordingly segmented regions (b)

3.2.4 Quality Control

To evaluate the quality of the outcome from the proposed segmentation process, Lari and Habib (2013) hypothesized different possible segmentation problems and proposed procedures to identify and solve these problems. These problems can be categorized as 1) non-segmented planar points, 2) misclassified rough points, 3) over-segmentation, 4) under-segmentation, and 5) invading/invaded segments. As a criterion to evaluate the quality of the derived surfaces from the proposed segmentation procedure, the surface roughness factor is used. The surface roughness factor is computed as the root mean square of the normal distances between the points within a segmented region and the best-fitted plane. A brief explanation of addressed problems by each QC procedure follows, and the schematic diagram of each category is shown in Figure 3.9.

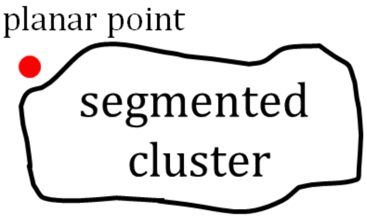
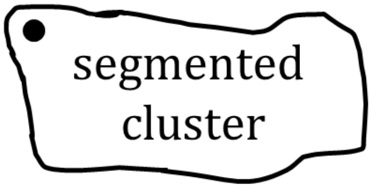
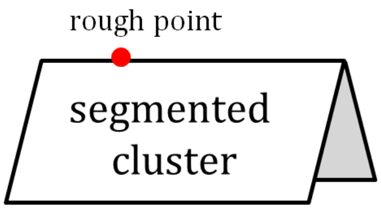
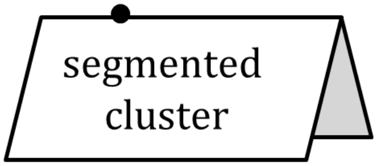
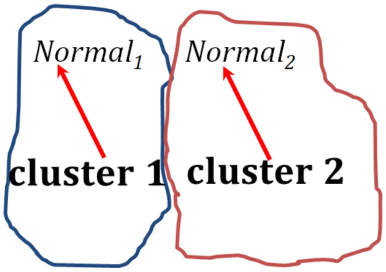
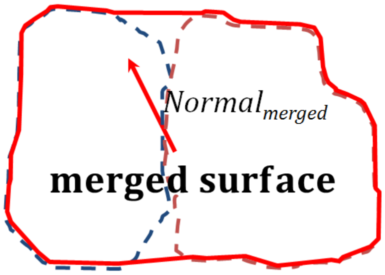
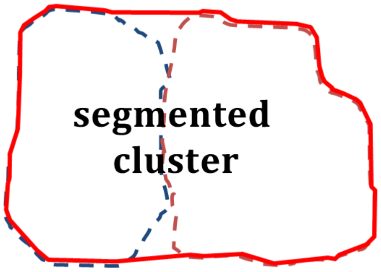
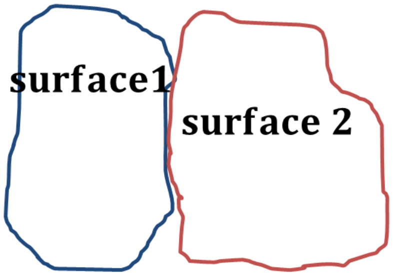
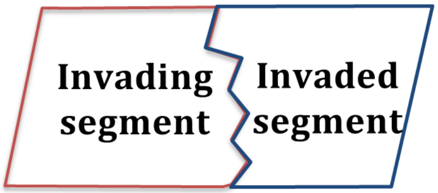
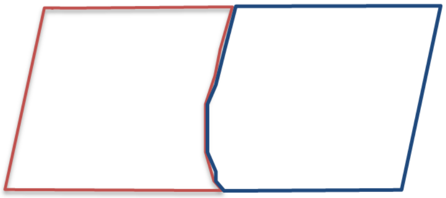
Category	Before QC	After QC
non-segmented planar points		
misclassified rough points		
over-segmentation		
under-segmentation		
invading/invaded segments		

Figure 3.9: The schematic diagram of addressed problems by the QC procedure

In the first case, a planar point might not have been segmented due to the minimum detectable cluster size or problems in the attributes calculation and/or cluster detection procedure. Its neighboring segment is identified according to the estimated local point density at its location, and its normal distance to the fitted plane through the points of the neighboring segment is calculated. If the normal distance is less than a predefined threshold, the point is added to the plane. The predefined threshold is chosen to be 2-3 times the neighboring segment's roughness factor. Next, a point classified as a part of a rough neighborhood may actually belong to a neighboring segment. This might happen due to improper classification of the points, especially at the edges of adjacent planar surfaces. In the same way as the first case, the normal distance to its neighboring plane is checked and added to the plane when it is less than a predefined threshold.

Due to problems either in the attribute computation or cluster/peak detection process, one planar surface might be segmented into more than one segment (i.e., over-segmentation), several planar surfaces may be segmented as one region (i.e., under-segmentation), or one segment could be invading or is being invaded by another segment (i.e., invading/invaded segments). For the over-segmentation case, when the surface normals of neighboring segments are similar and the surface roughness of the merged segments is similar to the individual segments, the segments are merged. In the case of under-segmentation, the average surface roughness factor using all the derived segments is assessed, and this criterion is used to detect under-segmentation. When the surface roughness factor of any segment is significantly higher than the average roughness factor, the segmentation process is repeated by defining a new origin location and new results are checked for any differences. After detecting the under-segmented clusters, an over-segmentation quality control procedure is performed to check whether or not newly-generated clusters can be merged with other clusters.

Lastly, to detect invading/invaded segments, each boundary point for a given cluster will be checked if there is any neighboring boundary point of another cluster whose normal distance to the cluster's best fitted plane is less than a threshold. When this happens, the boundary point from the neighboring segment will be moved to the selected cluster and the boundaries will be re-estimated for the invading and invaded clusters. Figure 3.10 shows the quality control of the segmentation results, where one can see that the quality of the segmentation is improved compared to the one before the QC procedure (Figure 3.8(b)).

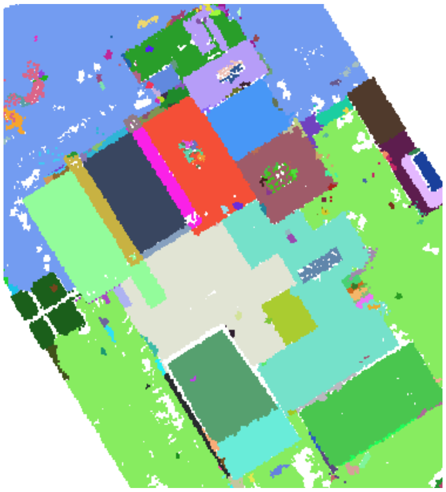
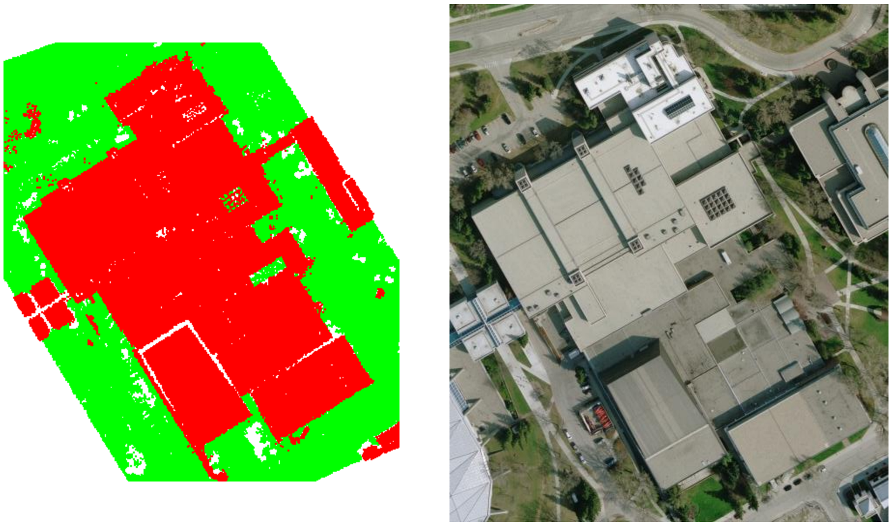


Figure 3.10: Quality control of segmentation results

3.3 Ground/non-ground classification

As a result of the proposed planar segmentation procedure, planar objects are detected from LiDAR data. Since buildings are assumed to be planar, non-ground objects, the LiDAR points should be classified into terrain and off-terrain points to find the non-ground planar objects. The classification approach employed in this research considers the 3D discontinuity between adjacent clusters in 2D and the physical properties of the derived clusters (i.e., area and slope) (Lari and Habib, 2012). First, off-terrain segments are found, the surfaces of which are steep with small areas or significantly higher areas than their neighboring segments. Therefore, the segments whose slopes are larger than the threshold and areas less than the threshold are classified as off-terrain surfaces. Then, the segments, which are higher than their 2D adjacent segments, are identified as off-terrain surfaces. When the segments are similar in height to adjacent ones that already have been classified as off-terrain, they are also classified as off-terrain. Once the off-terrain segment classification process is finished, finding terrain regions, which are usually low in height and large in area, is conducted. For this, the remaining unclassified segments are sorted based on the average height of their boundary points. When the area of the segment with the minimum average height is larger than a predefined threshold, the segment is classified as terrain. Then, the remaining non-classified segments are compared to their nearest terrain-classified segments. If the average height of the segment is similar to the nearest terrain region, it is deemed to be part of the terrain surface. Otherwise, it is classified as an off-terrain object. This process is performed in an ascending order based on the average height of their boundary points. Figure 3.11 presents the ground/non-ground classification results for the planar regions shown in Figure 3.10. Regions that are red in color have been classified as off-terrain objects and those green in color are terrain regions. A corresponding image is presented as a reference.

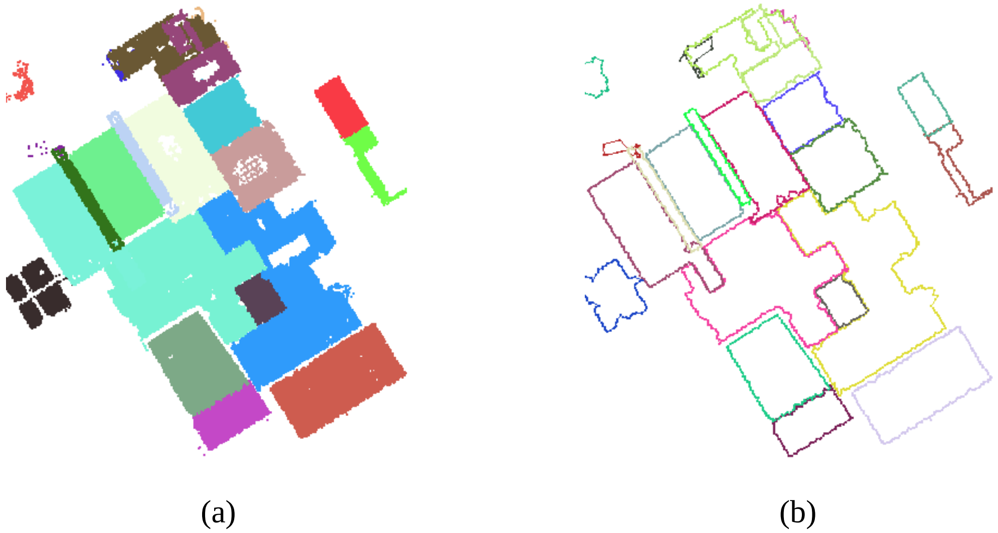


(a) (b)

Figure 3.11: An example of ground/non-ground classification (ground displayed in green and non-ground in red color) (a) and aerial image over the same area (b)

3.4 Boundary detection and quality control

Among the points that were identified as planar and non-ground through the aforementioned procedures, the points whose sizes and heights above neighboring terrain regions are larger than the predefined thresholds are derived as building hypotheses. Figure 3.12 shows the derived building hypotheses (a) and their traced boundaries (b). Please refer to Figure 3.11(b) for the corresponding image.



(a) (b)

Figure 3.12: Example of building hypotheses generation (different hypothesized buildings in different colors) (a) and their boundaries (b)

Figure 3.13 shows the projection of one of the building boundaries onto the corresponding imagery. It can be seen that the boundary is close to the real shape of the building, but it does not coincide with the precise building boundary. This is caused by the relatively low sampling rate of airborne LiDAR systems. Therefore, the detected building boundaries need to be refined. The refinement process will be discussed in Chapters 4 and 5.

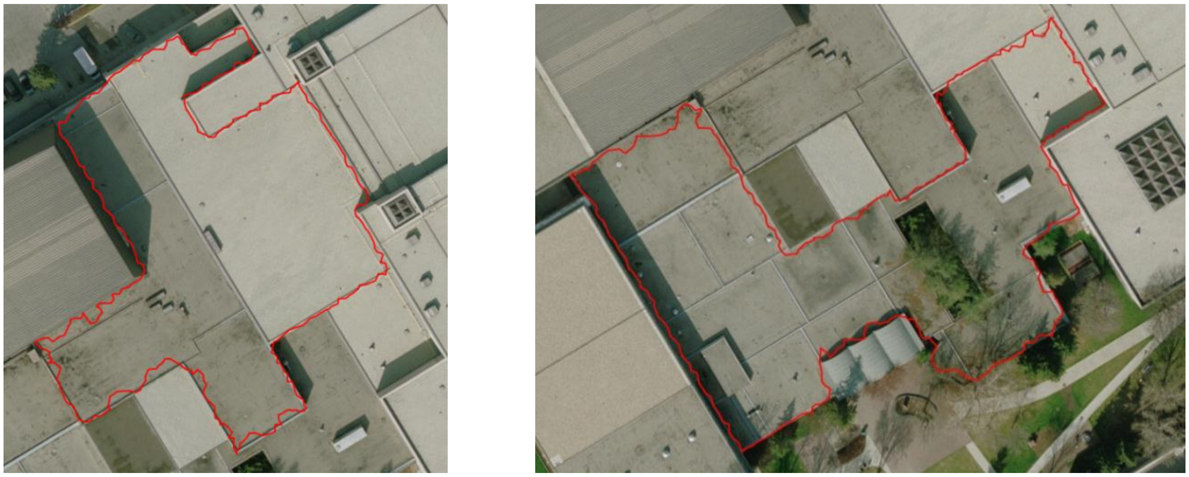


Figure 3.13: Example of projected LiDAR boundaries onto an image

Considering the fact that the goal of building detection is to distinguish buildings from other objects, the performance of the method can be evaluated as to whether or not the buildings are correctly identified. The question to be answered is: when can one conclude that the building is identified correctly? Performance metrics have been discussed in numerous papers (e.g., (Shufelt, 1999; Rottensteiner et al., 2007; Rutzinger et al., 2009; Rottensteiner et al., 2011). However, depending on which entities (e.g., pixels, voxels, areas, or objects) are evaluated and what criteria define a correct detection, different evaluation results can be derived for the same area.

The most commonly used metrics for thematic accuracy are based on the confusion or error matrix, which compares the detection results with reference data. Since buildings are the only focus of detection, two classes can be considered (i.e., buildings and other objects). Regions that are not part of the building hypotheses are all considered as other objects. In this case, four categories for each region are possible, which are summarized in Table 3.1. A True Positive (TP) is an object that is detected as a building which is a building in the reference data. A False

Positive (FP), i.e., an error of commission in the building class, is a non-building object in the reference data which is detected as a building. A False Negative (FN), i.e., an error of omission in the building class, is an object that corresponds to a building in the reference data which is detected as other objects, and a True Negative (TN) is an object that corresponds to other objects that are detected as non-building.

Table 3.1: The confusion matrix

	Reference data	
Sample data	Building	Other object
Building	True Positive (TP)	False Positive (FP)
Other objects	False Negative (FN)	True Negative (TN)

Heipke et al. (1997) introduced the *completeness* and the *correctness* of the building detection results based on the confusion matrix according to Equations (3.2) and (3.3) below. *Completeness* represents the rate of detected buildings among the buildings in the reference data, and *correctness* shows the rate of buildings among the ones detected as buildings in the test data. A good classification should satisfy both conditions, namely, high completeness and high correctness. Therefore, in this research, both are used as a quality indicator.

$$Completeness = \frac{TP}{TP + FN} \tag{3.2}$$

$$Correctness = \frac{TP}{TP + FP} \tag{3.3}$$

The remaining question is what constitutes a TP. Rutzinger et al. (2009) provided an extensive review on different definitions of what establishes a TP; and having 50-70% overlap between the entity and the reference data is typically considered as acceptable criteria for a TP. The ISPRS benchmark analyzed and compared the segmentation quality based on completeness and correctness of the extracted roof planes (Rottensteiner et al., 2012); and roof planes that have more than 50% overlap with the planes in the reference data are considered as a correct detection. Since an assessment result that is based on mutual overlap depends on the overlap threshold, the centroid of a detected building is checked as to whether it falls inside the building in the reference data set. The comparison can be carried out at two levels. First, it can be performed in the object level (i.e., building by building), then region level (i.e., detected plane by plane). One should note that only planes whose areas are larger than the minimum detectable area are considered for the comparison.

3.5 Summary

This chapter discusses the methodology to detect buildings automatically from LiDAR data and generate building hypotheses accordingly. Given the fact that buildings are above ground and are mainly comprised of planar rooftops, LiDAR data segmentation and ground/non-ground classification procedures are performed. Segmentation procedure proposed by Lari et al. (2011) is employed to detect the planar objects from LiDAR data. This segmentation approach considers the local point density thus is able to handle datasets with non-uniform point densities. One should note that the current research is not limited to the employed segmentation approach and any other available LiDAR segmentation approaches can be incorporated. To find non-ground planar objects, ground/non ground classification is performed based on the segmentation results. At the end, the non-ground, planar regions whose area and heights above neighboring terrain regions are larger than the predefined thresholds are derived as building hypotheses. The performance of the method is evaluated as to whether or not the buildings are correctly identified. The completeness and correctness are calculated based on the building level and also the plane level.

Chapter Four: **Automatic decomposition/evaluation of LiDAR building boundary**

4.1 Introduction

In Chapter 3, the procedures utilized to detect buildings from LiDAR data were explained, and building boundaries were successfully obtained. LiDAR data have been proven to be useful for deriving planar features in an automatic way and are suitable for detecting buildings at a high success rate. However, as presented in the previous chapter, the derived boundaries of the buildings from the LiDAR data need to be improved since they do not represent the actual boundaries of the objects due to the relatively low sampling rate of the airborne LiDAR system. Therefore, in order to improve the quality of the LiDAR derived building boundaries, photogrammetric data will be integrated in this research and a model-based approach is adopted. Model-based reconstruction, also referred to as a top-down approach, introduces prior knowledge about the object of interest during the reconstruction process. The object of interest, in this case, is the derived LiDAR building boundaries in Chapter 3. In this research, any right-angled-corner buildings, which include box, shed-roof, and gable-roof buildings or a mixture thereof, can be modeled using rectangular primitives. The justification for choosing a rectangular primitive will be explained in Section 4.2. In order to associate the LiDAR-derived boundaries from the data-driven approach with the model-based approach without human intervention, the boundaries need to be decomposed into rectangular primitives. This chapter introduces an automatic building boundary decomposition algorithm into the rectangular primitives and proposes criteria to evaluate whether or not the primitives accurately describe the building in question.

Since buildings that are comprised of rectangles are the focus of this research and a rectangle is chosen as the basic primitive, the Minimum Bounding Rectangle (MBR) algorithm is introduced (Chaudhuri and Samal, 2007; Freeman and Shapira, 1975). The MBR algorithm is modified here to achieve the purpose of this research and is applied in a recursive way (i.e., Recursive MBR or RMBR). This algorithm regularizes the initial LiDAR boundaries and generates sets of rectangular primitives. The RMBR algorithm can also determine/control the level of detail for the final building models. Depending on the application, building models with a high level of detail might not be necessary. In this case, the model can be generalized through the RMBR process. Section 4.3 discusses the detailed procedure of the RMBR algorithm. Even though the most common building shapes in urban areas are right-angled-corner buildings, irregular buildings still exist that cannot be reconstructed using rectangular primitives. To automatically generate building models over large urban areas, it is important to decide whether or not the buildings can be reconstructed using the proposed methodology. Therefore, Section 4.4 introduces the methods this research will use to examine the quality of the MBRs to verify whether or not the derived primitives can be used to accurately model the building in question.

4.2 Choice of the rectangular primitive

Conventionally, in the model-based approach, each primitive is associated with *pose* and *shape parameters* to describe its geometric properties. The *pose parameters* define the position (X_0, Y_0, Z_0) and orientation (ω_m, φ_m, κ_m) of the model primitives in object space as seen in Figure 4.1. The shape and size of the models are defined using *shape parameters*. For example, Figure 4.1 shows that for a solid box primitive, the *shape parameters* are defined by length (ℓ), width ($\mathscr{w}$) and height ($\mathscr{h}$). Different models are represented using different *shape parameters* (Tseng and Wang, 2003).

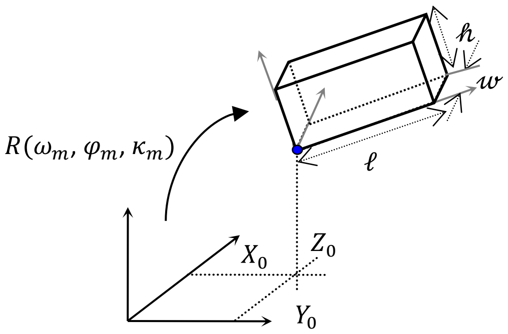
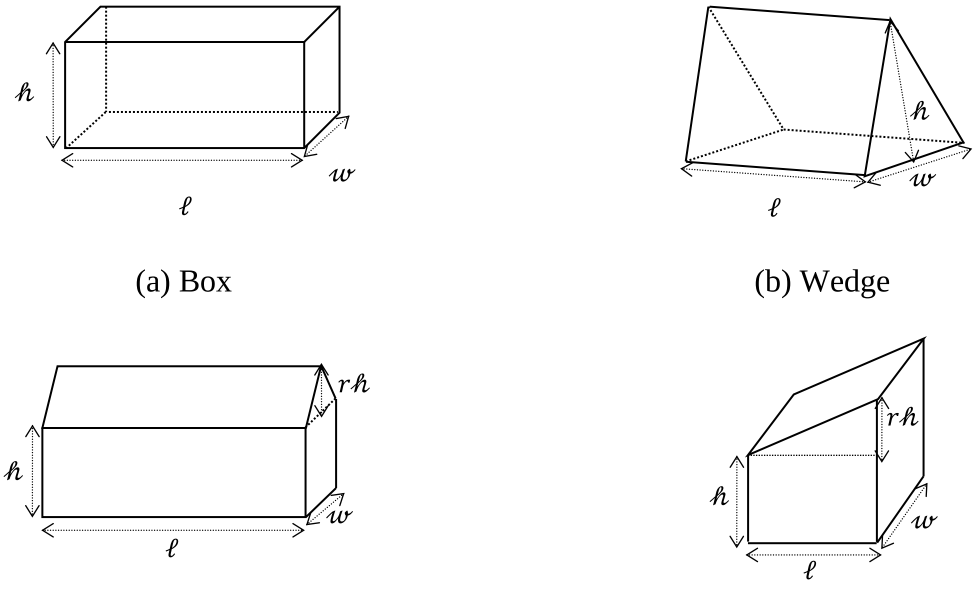


Figure 4.1: Definition of pose and shape parameters for a solid box primitive

Figure 4.2 shows the representation of several popular models using different shape parameters. Figure 4.2(a) and (b) show the most popular basic model primitives, which are box and wedge models. As mentioned before, the model-based approach can construct complex building models by combining a set of basic model primitives such as box, wedge, and cylinder models. As seen in Figure 4.2(c) and (d), gable-roof buildings and shed-roof buildings can be modeled by combining box models and wedge models. As a case in point, some of the past model-based literature predefined gable-roof building model primitives (Suveg and Vosselman, 2000; Tseng and Wang, 2003). In this case, the lack of predefined models or a poor choice of the proper model can lead to failure in reconstructing accurate building models.



(a) Box

(b) Wedge

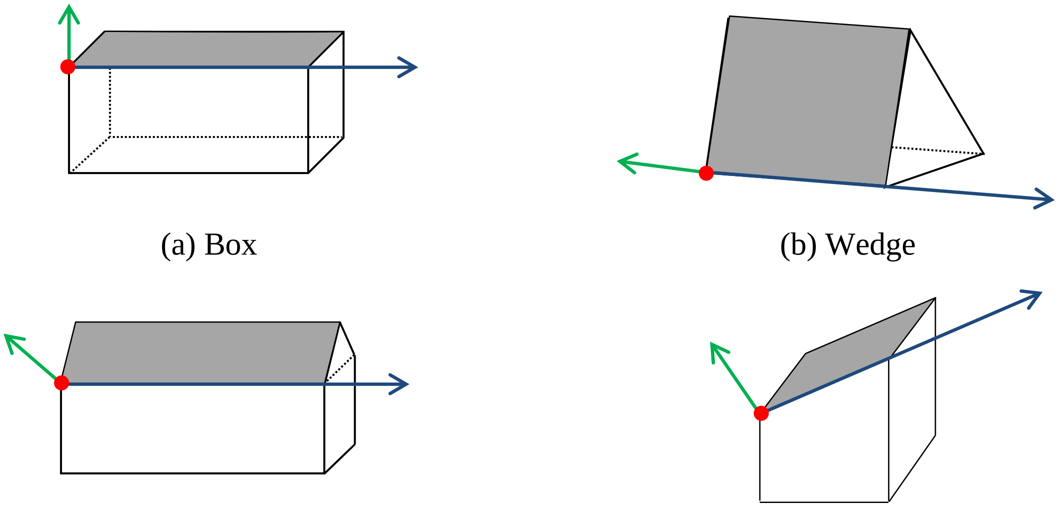
(c) Gable-roof building

(d) Shed-roof building

Figure 4.2: Shape parameters for different model primitives

Therefore, in this research, it is aimed to reduce the number of model primitives while still being able to deal with the aforementioned basic models. For this, the box primitive, which is the most basic model, is selected as the initial step. In the case of such a model, three *shape parameters*, the length, width, and height of the box, are required. However, in imagery, there is no guarantee that both the rooftops and footprints of buildings are visible at the same time so the heights of the buildings are hard to determine from imagery. In addition, LiDAR data provides both the rooftops of the buildings and the terrain, which suggests that the elevation of the building roof patch generated from LiDAR data is more reliable than the derived boundaries themselves.

Therefore, in this research, h is determined from the LiDAR data, which simplifies the box model down to a rectangular model primitive. By utilizing a rectangular primitive together with their plane parameters from LiDAR data, instead of a box primitive, complex buildings that are comprised of rectangles with various slopes can be reconstructed. Figure 4.3 illustrates how rectangular primitive can be used to represent different building shapes mentioned in Figure 4.2.



(a) Box

(b) Wedge

(c) Gable-roof building

(d) Shed-roof building

Figure 4.3: Rectangular primitive representation for different buildings

For this purpose, the building boundaries will be decomposed into sets of rectangular primitives in Section 4.3. All the decomposed rectangles forming the rooftops will be adjusted sequentially (Chapter 5).

4.3 Recursive Minimum Bounding Rectangle

The MBR algorithm finds the rectangle with the minimum area among the rectangles with arbitrary orientations that contains all the given points; in this case, the LiDAR derived boundary points (Freeman and Shapira, 1975; Chaudhuri and Samal, 2007). The derivation of the MBR begins with defining arbitrary coordinate systems using the boundary points. The x-axis is defined along the vector between a point and its next neighboring point and y axis follows a right-handed coordinate system. As Figure 4.4 illustrates, the minimum x and y and the maximum x and y coordinates are found according to the defined coordinate system. The four coordinates ($\min x$,$\min y$), ($\max x$,$\min y$), ($\max x$,$\max y$), and ($\min x$,$\max y$) define a rectangle. This process is repeated based on every coordinate system defined using all the boundary points with their next neighboring point. The areas of the rectangles are calculated and the rectangle with the minimum area gives the MBR.

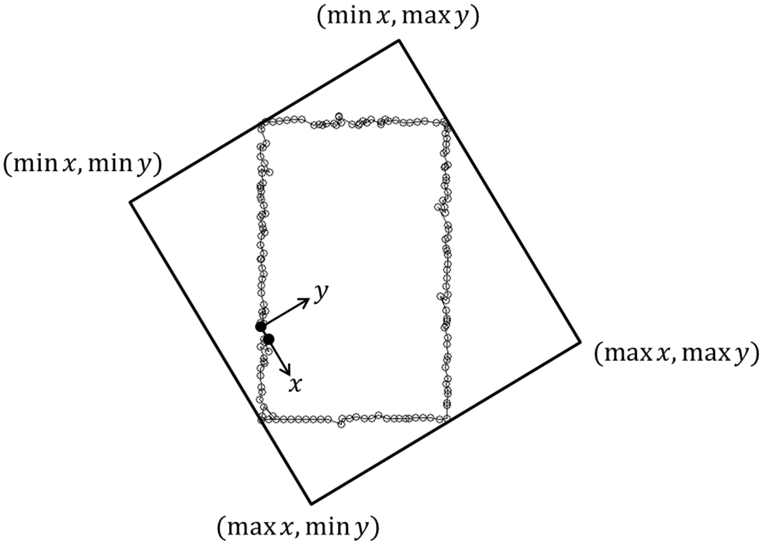
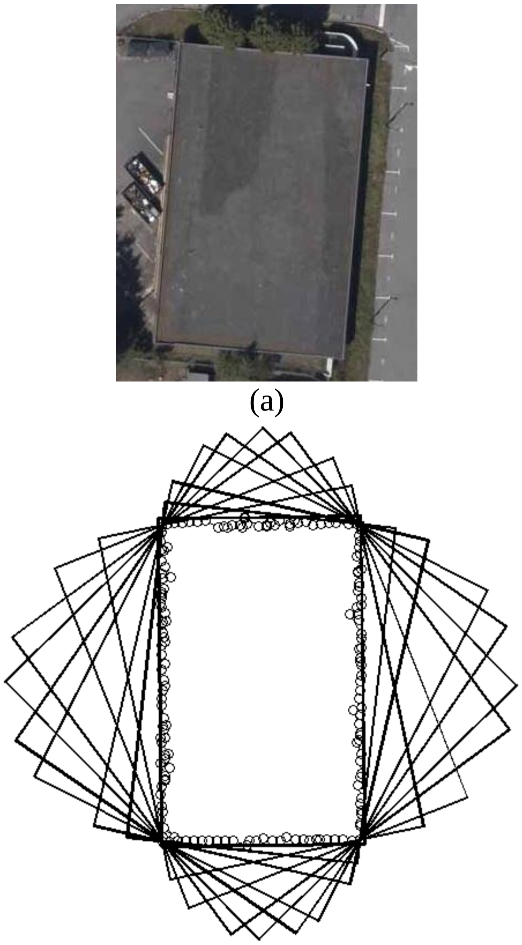


Figure 4.4: A bounding rectangle with a coordinate system definition using two consecutive points along the boundary

Figure 4.5(a) and (b) show an aerial image over a simple rectangular building and its boundary derived from LiDAR data projected onto the image. Figure 4.5(c) shows some of the possible rectangles with arbitrary orientations that contain the boundary points. Among all the possible bounding rectangles, the rectangle with the minimum area is chosen and it is shown in Figure 4.5(d).

As seen in Figure 4.5, the original MBR algorithm chooses the rectangle with the minimum area among the possible bounding rectangles. This condition produces satisfactory results in most cases; however, in some cases this condition is not enough to determine the best representative rectangle of the boundary in question.



(a) (b)

(c) (d)

Figure 4.5: An example of a simple rectangular building: aerial image (a), LiDAR derived boundary (b), bounding rectangles with arbitrary orientations (c), and the minimum bounding rectangle (d)

As seen in Figure 4.6(a), depending on the shape of the buildings, the bounding rectangle with the minimum area might derive a rectangle that does not coincide with the main orientation of the building. The representative rectangle should be redefined as a rectangle with the minimum area that has considerable overlap with the investigated boundary (i.e., LiDAR-based boundary).

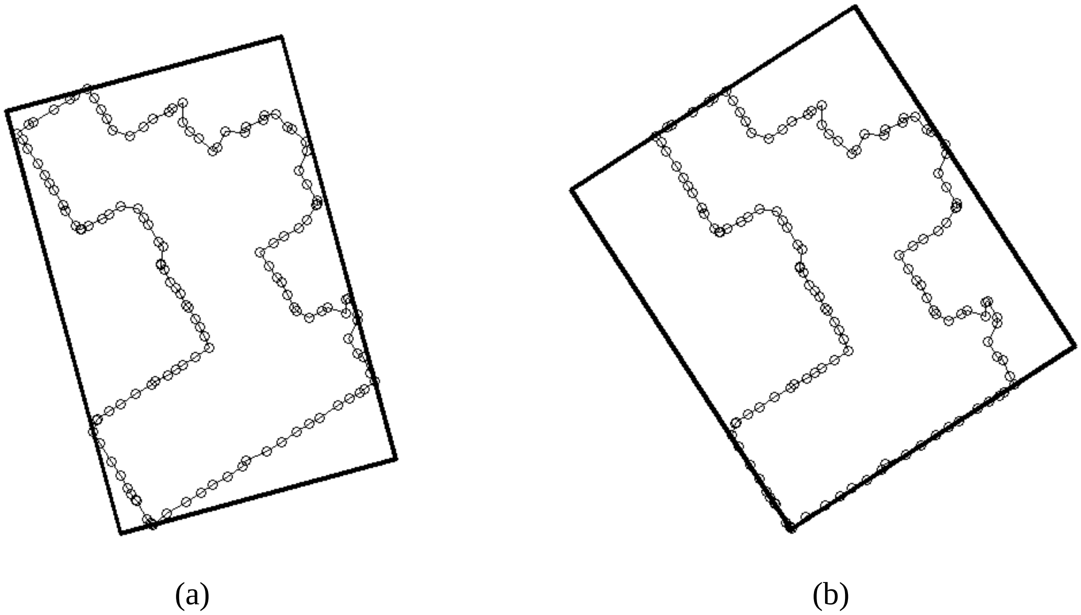


Figure 4.6: The MBR with the minimum area condition (a) and the MBR with the maximum overlapping ratio condition (b)

Therefore, in this research, the number of boundary points aligned with the MBR (i.e., overlapping ratio between the given boundary points and the MBR) is considered as a new criterion. The distance between the boundary points and the closest side of the generated bounding rectangle is calculated, and the number of points whose distance is less than a threshold, which is related to the average point spacing of the dataset, are counted. The bounding rectangle with the maximum number of boundary points within the normal-distance threshold is chosen as the minimum bounding rectangle instead of the one with the minimum area. Figure

4.6(b) shows the results from the modified MBR algorithm; namely, the rectangle with the largest number of boundary points aligned with the resulting MBR. Note that the resulting MBR is aligned with the main orientation of the building boundary.

To obtain regularized boundaries with multiple rectangles, this algorithm can be applied recursively to complex structures. Figure 4.7 illustrates the flowchart of the RMBR procedure proposed in this research. The RMBR algorithm starts by applying the introduced MBR algorithm to the LiDAR-derived boundary and the generated MBR is called the first level MBR. Then, tracking the non-overlapping initial boundary segments with the first level MBR is performed. When the length of the detected non-overlapping segments is larger than a threshold, the MBR algorithm is applied again on the non-overlapping segments, and their projection onto the MBR sides to derive the next level MBR. It should be pointed out that each level, except the first level, can have more than one MBR as shown in Figure 4.8. The same procedure is repeated until there are no LiDAR boundary points remaining. As a result of the recursive MBR, sets of rectangles are derived depending on the required level of detail (i.e., either one level of rectangle or several levels of rectangles). Examples to demonstrate the step–by–step procedure of the RMBR algorithm are introduced later. The building shapes differ in terms of complexity, which means they are represented using different MBR levels.

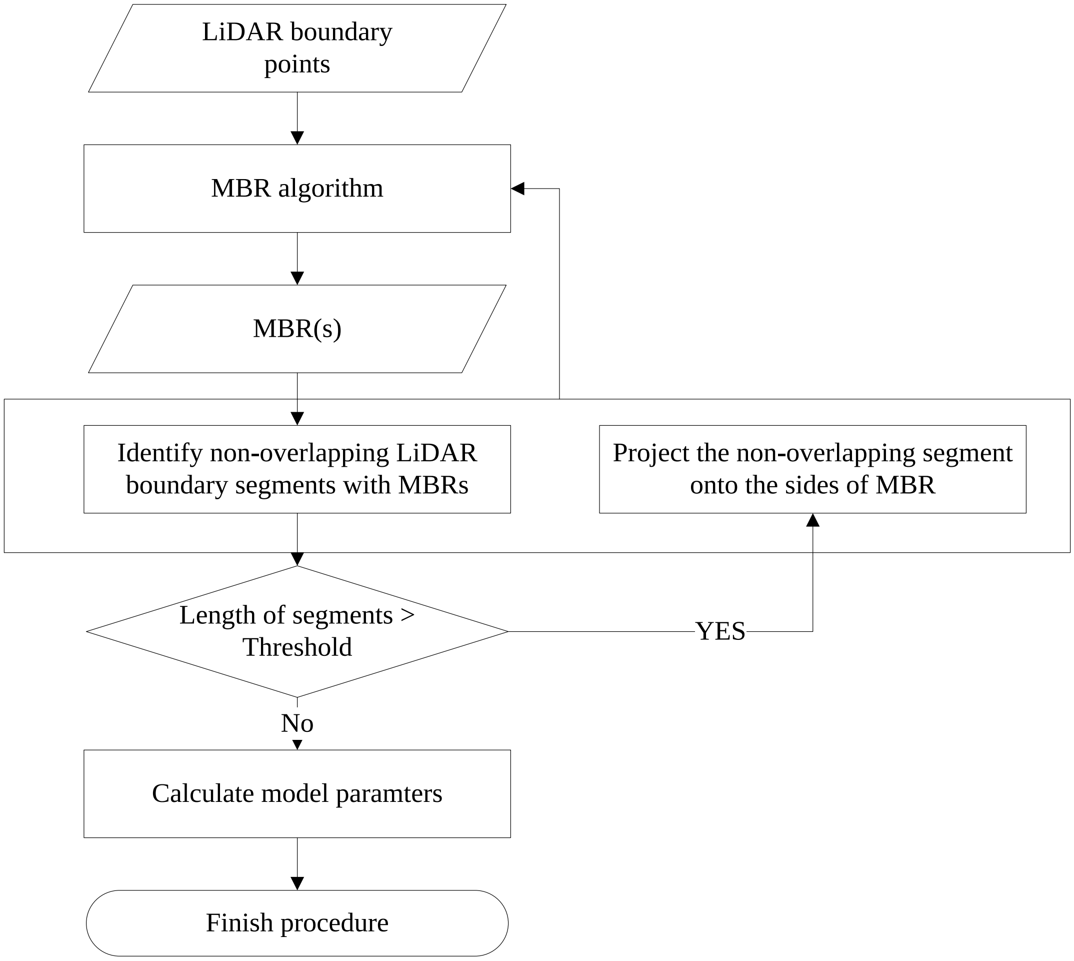
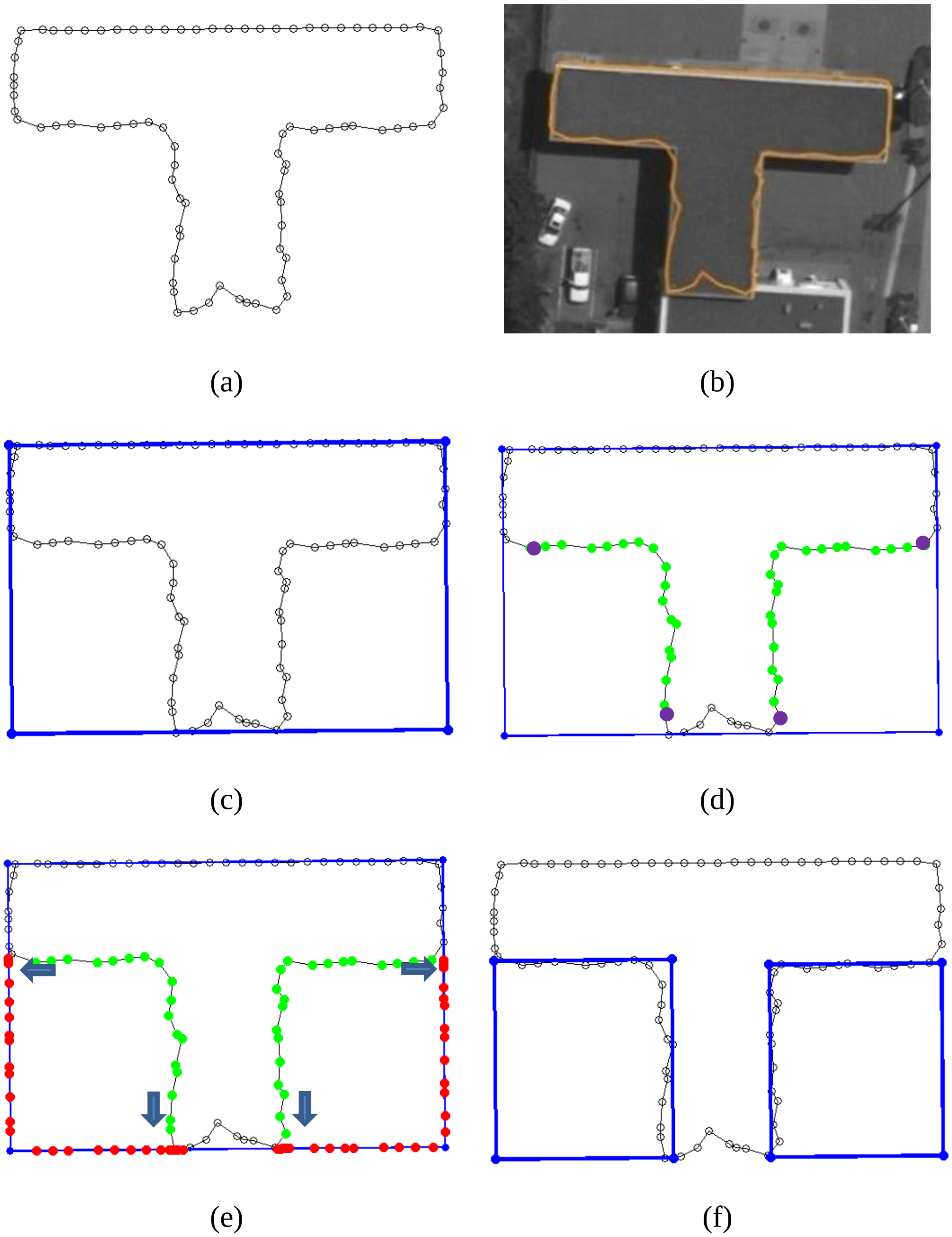

LiDAR boundary points
MBR algorithm
MBR(s)
Identify non-overlapping LiDAR boundary segments with MBRs
Project the non-overlapping segment onto the sides of MBR
Length of segments > Threshold
YES
No
Calculate model paramters
Finish procedure

Figure 4.7: Flowchart of the RMBR procedure

Figure 4.8(a) and (b) show an initial LiDAR boundary of the building and its projection onto an image. As proposed, the MBR algorithm is applied to the boundary points and the first level MBR (i.e., the blue rectangle in Figure 4.8(c)) is generated. Then, the entire boundary points are traced to find non-overlapping boundary segments with the first level MBR. The distances between the boundary points to the closest side of the MBR are calculated. If the distance is larger than a threshold which is twice the average point spacing of the LiDAR dataset, then the first point whose distance is larger than a threshold is considered as the start of the non-overlapping segment with the first level MBR. All the connected neighboring points whose distances are larger than the threshold are recorded as one non-overlapping segment until a point coincides with the MBR. The last non-overlapping point is identified as the end point of the non-overlapping segment. The closest MBR side(s) where the non-overlapping segments start and end (i.e., purple circles in Figure 4.8(d)) is recorded for the next step. Non-overlapping segments that contain less than the predefined number of points are ignored. The predefined number of points is determined considering the LiDAR point density and the model's target level of detail. As a result of this process, the lists of boundary segments that do not overlap with the detected MBR are found as seen in Figure 4.8(d) (green circles). The non-overlapping boundary segments are projected onto the MBR sides found in the previous step as displayed by the arrows in Figure 4.8(e), and the projected points are displayed in red circles. Using the non-overlapping boundary points and their projected counterparts, the MBR algorithm is applied to derive the second level MBRs. In this case, two second level MBRs are derived as seen in Figure 4.8(f). Since all the boundary points overlap with the derived MBRs, the procedure stops. It is worth mentioning that the orientation of the second level MBR always follows the first level MBR since the projected points, which are 50% of the points used to detect the second level MBRs, fall on the first level MBR. In addition, the MBRs share one of the vertices of the rectangles with the previous level MBR, which only leaves the dimensions of the rectangle in question for the next level MBR.



(a) (b)

(c) (d)

(e) (f)

Figure 4.8: The step-by-step illustration of the RMBR procedure for two MBR levels

When finding the closest MBR side(s) of the start and end points of the non-overlapping segments, one should check for the condition that the projection rays do not go through the LiDAR boundary points. The condition is implemented in a way that ensures the two projected sides are in sequential order. Figure 4.9 shows an example of when this condition is not considered. After the first MBR is derived, one non-overlapping segment (i.e., green and purple circles) is found. The LiDAR-derived boundary points are ordered counter-clock-wise and the closest sides of the start and last of the non-overlapping segment are not sequential (i.e., side 1 and side 4). When projecting the non-overlapping points onto the closest sides of the start and end points (i.e., red circles), the projections rays go through the given LiDAR boundary points, as seen in Figure 4.9(a). The final shape can be derived by subtracting the second level MBR from the first level MBR. As a result of this, an undesirable final shape is derived as seen in Figure 4.9(b). On the other hand, Figure 4.10(a) shows that the non-overlapping segments are projected onto the other side of the rectangle by checking the proposed condition. One can observe that Figure 4.10(b) shows a result that represents the LiDAR boundary better.

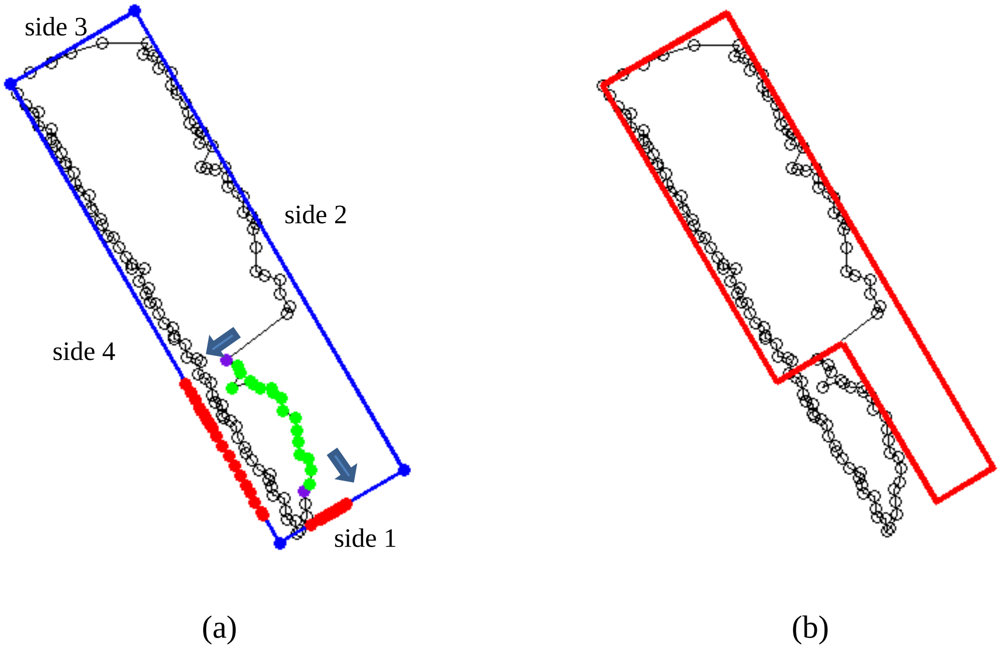
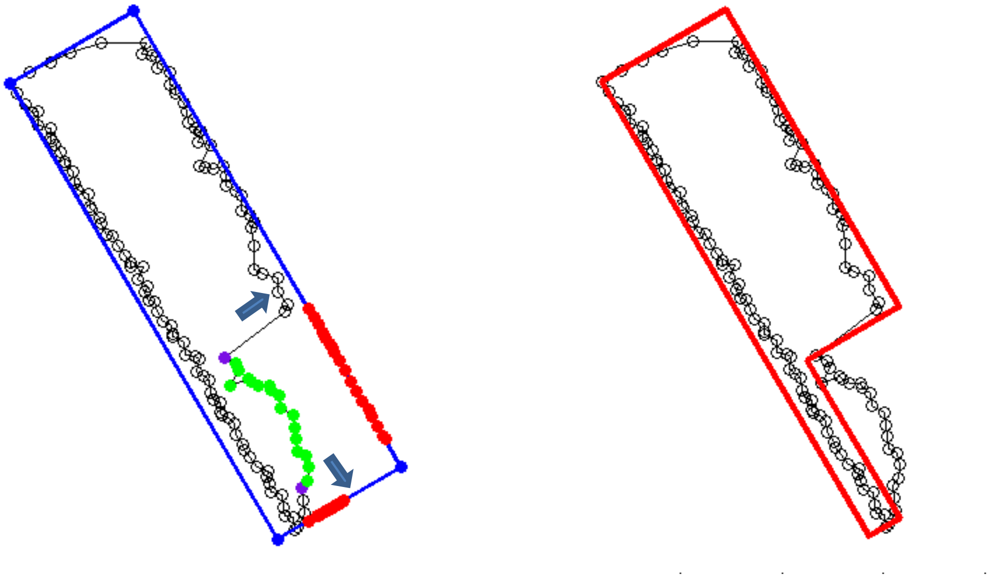


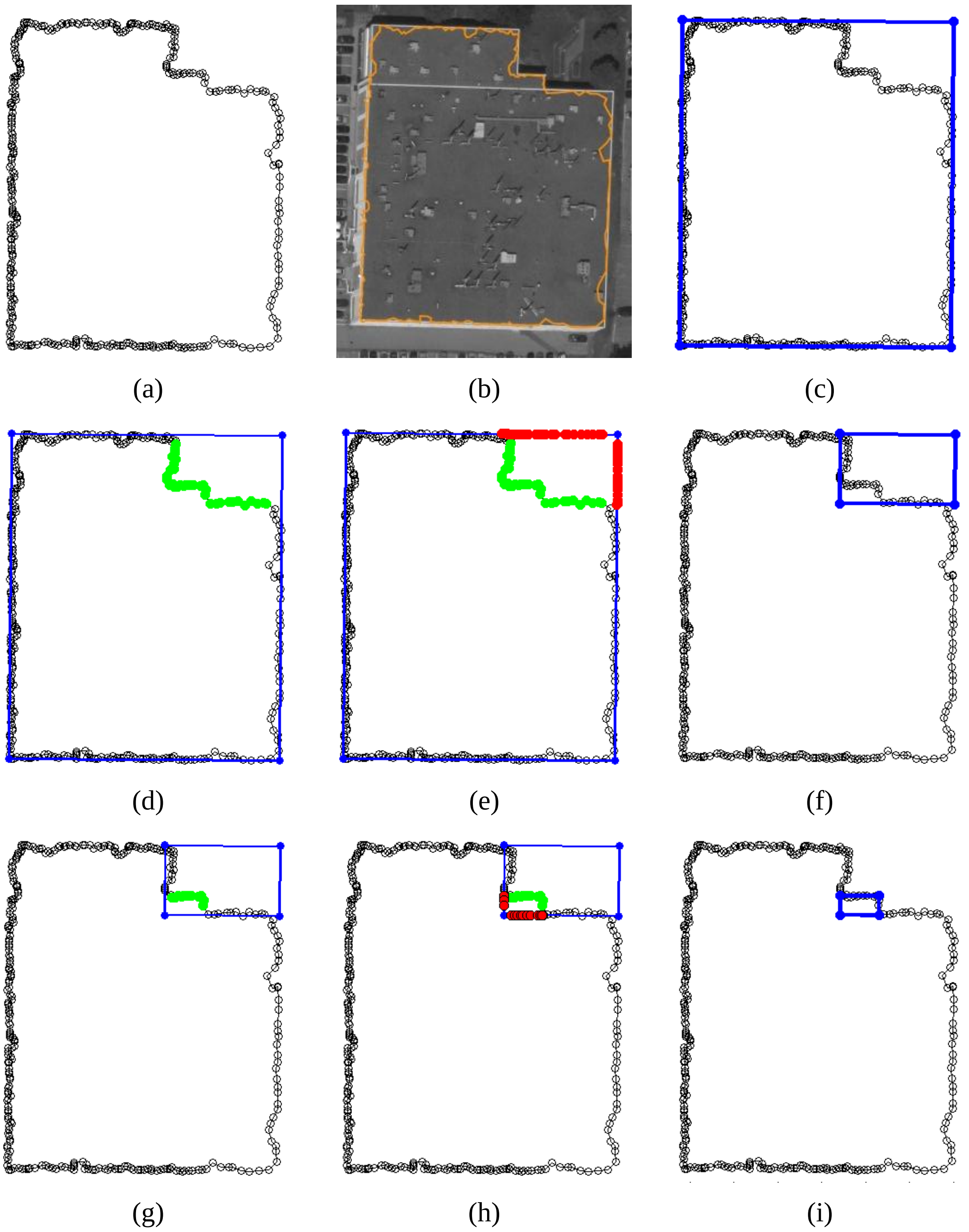
Figure 4.9: Example of projecting non-overlapping segments with the projection rays going through the LiDAR boundary points (a) and the RMBR result (b)



(a) (b)

Figure 4.10: Example of projecting non-overlapping segments without the projection rays going through the LiDAR boundary points (a) and the RMBR result (b)

Figure 4.11 illustrates the RMBR procedure for another building which includes three MBR levels. An initial LiDAR boundary of the building and its projection onto an image is presented in Figure 4.11(a) and (b). The same procedure demonstrated in the previous example is applied, and the first level MBR is shown in Figure 4.11(c). Figure 4.11(d) displays the non-overlapping points with the first level MBR in green circles; and Figure 4.11(e) shows their projection onto the first level MBR in red circles. The second level MBR (i.e., Figure 4.11(f)) is derived using the points in the colors green and red in Figure 4.11(e). The same procedure is repeated to find the non-overlapping LiDAR points with the second level MBR (i.e., green circles in Figure 4.11(g)) and their projection onto the second level MBR (i.e., red circles in Figure 4.11(h)). Figure 4.11(i) shows the third level MBR in blue.



(a) (b) (c)

(d) (e) (f)

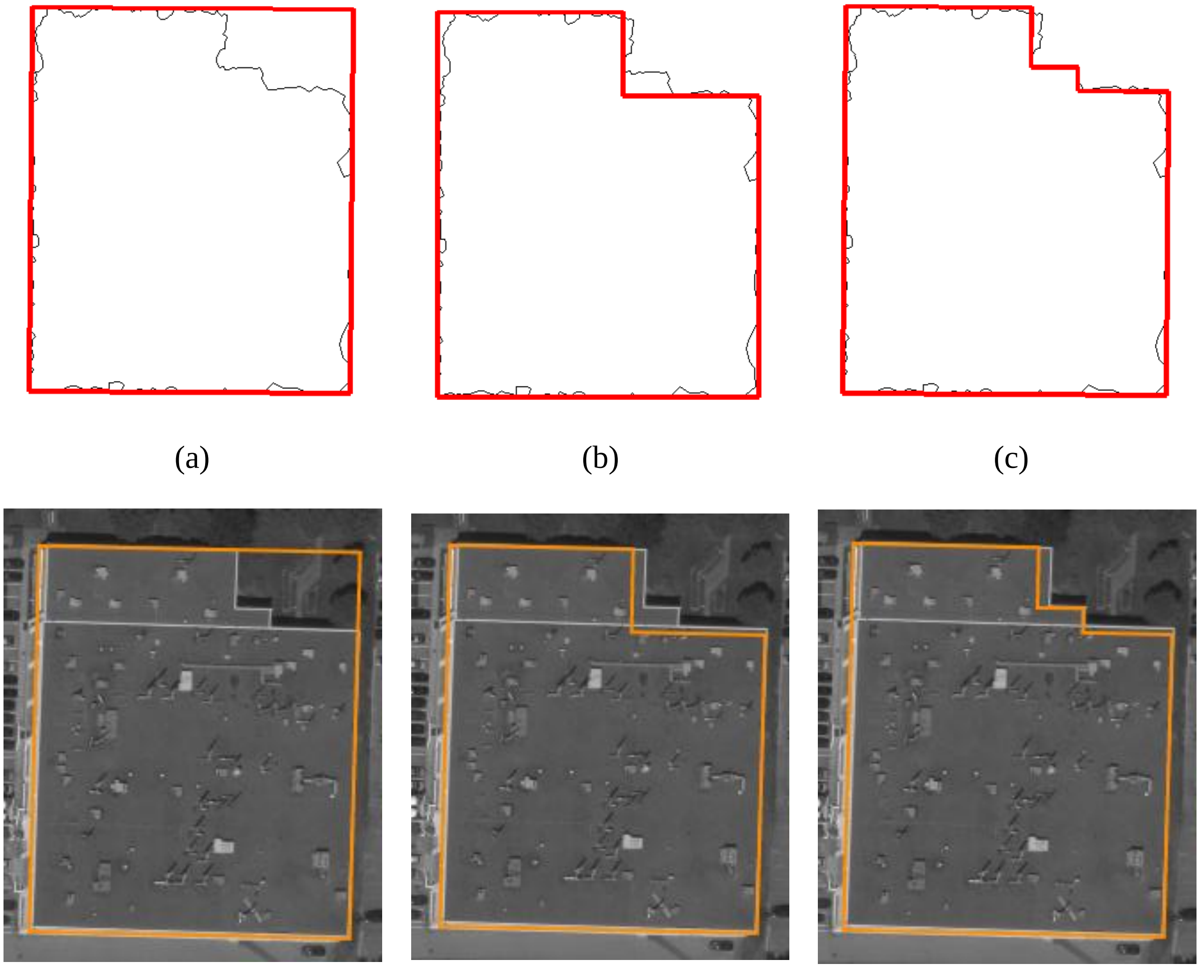
(g) (h) (i)

Figure 4.11: The step-by-step illustration of the recursive MBR procedure for three MBR levels

As a result of the recursive MBR, different levels of MBRs are derived and by alternating subtraction and addition of each level, the final shape can be generated. This operation is represented by the alternating series in Equation (4.1), where k represents the RMBR levels and n represents the maximum number of RMBR levels. In practice, the maximum level of RMBRs (n) does not exceed five levels even in the most complex building cases. The maximum level of RMBRs determines the level of detail for the final building models and the maximum level is determined by the LiDAR point density. More specifically, details less than twice average point spacing of the dataset will not be detected.

$$Final\ Shape = \sum_{k=1}^{n} (-1)^{k} \mathrm{RMBR}_{k} \qquad \textbf{(4.1)}$$

Figure 4.12 presents the results of the alternating series operation according to Equation (4.1) on the MBRs from Figure 4.11 using different levels of detail. Figure 4.12(a) shows the single level MBR (n =1), and it is identical to the first level MBR. Figure 4.12(b) uses two levels of MBR (n =2), and the results are derived by subtracting the second level MBR from the first level MBR. Three levels of MBR (n =3) are constructed by adding the third level MBR to the previous result and is shown in Figure 4.12(c). The projection of the final shape using different levels of MBRs onto the corresponding images are shown in Figure 4.12(d), (e), and (f).



(a) (b) (c)

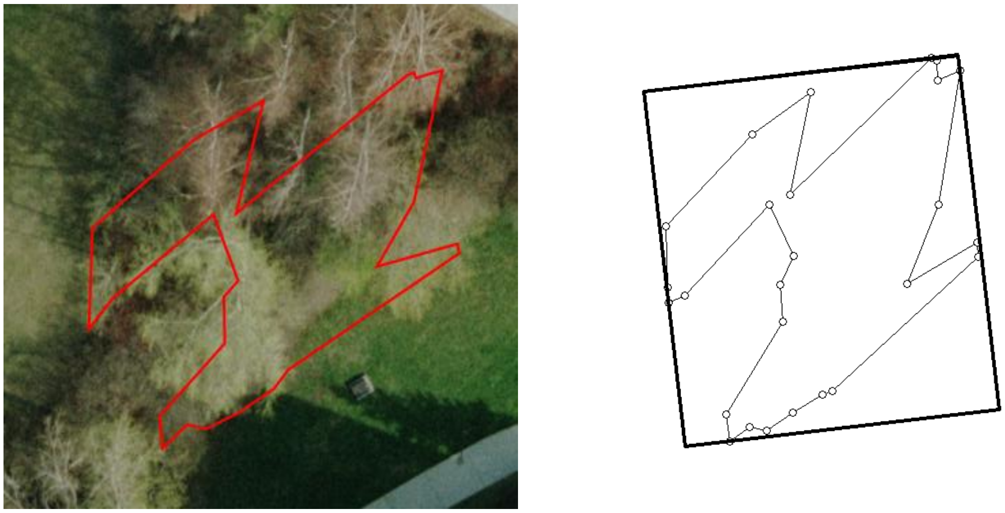
(d) (e) (f)

Figure 4.12: Final results using different RMBR levels: single level (a), two levels (b), three levels (c), projection of single level RMBR operation onto an image (d), projection of two levels of RMBR operation onto an image (e), and projection of three levels of RMBR operation onto an image (f)

It can be seen that, even though the final shape derived from the LiDAR data show quite promising results in terms of the building location and general shape representation, the horizontal locations of the boundaries need improvement. Therefore, before performing the operations of different MBR levels, the determined rectangular primitives from the LiDAR data will undergo a refinement process using the image data based on the model-based image fitting which will be discussed in Chapter 5.

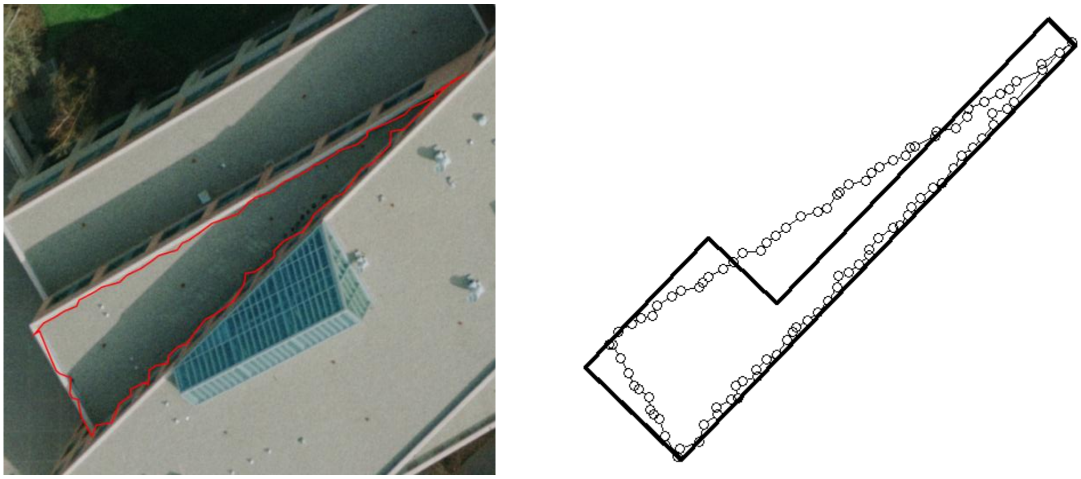
4.4 The RMBR evaluation

The proposed RMBR algorithm decomposes the LiDAR-derived boundaries into sets of rectangles automatically. As previously mentioned, they are to be adjusted sequentially using a model-based photogrammetric technique to improve the horizontal accuracy of the LiDAR boundary. As mentioned earlier, the model-based image fitting in this research is based on rectangular primitives. However, not all the detected buildings are regular in shape and comprised of sets of rectangles. Moreover, depending on the noise level in the LiDAR data, the thresholds used for plane segmentation and the area/height thresholds for building hypotheses, there might be instances where non-building objects might be falsely identified as buildings. Figure 4.13(a) shows the LiDAR boundary of a tree object projected onto an image. The tree object is an example of false positive buildings (i.e., non-building objects that are classified as buildings) that should not have been included as a building hypothesis. Figure 4.14(a) shows the boundary of an irregular building, which cannot be decomposed into rectangles, projected onto an image. Figure 4.13(b) and Figure 4.14(b) illustrate that the RMBR algorithm generates a bounding rectangle which has the maximum overlapping ratio with the boundary points regardless of the building shape. This will give false model primitives that do not represent the physical boundaries of the building.



(a) (b)

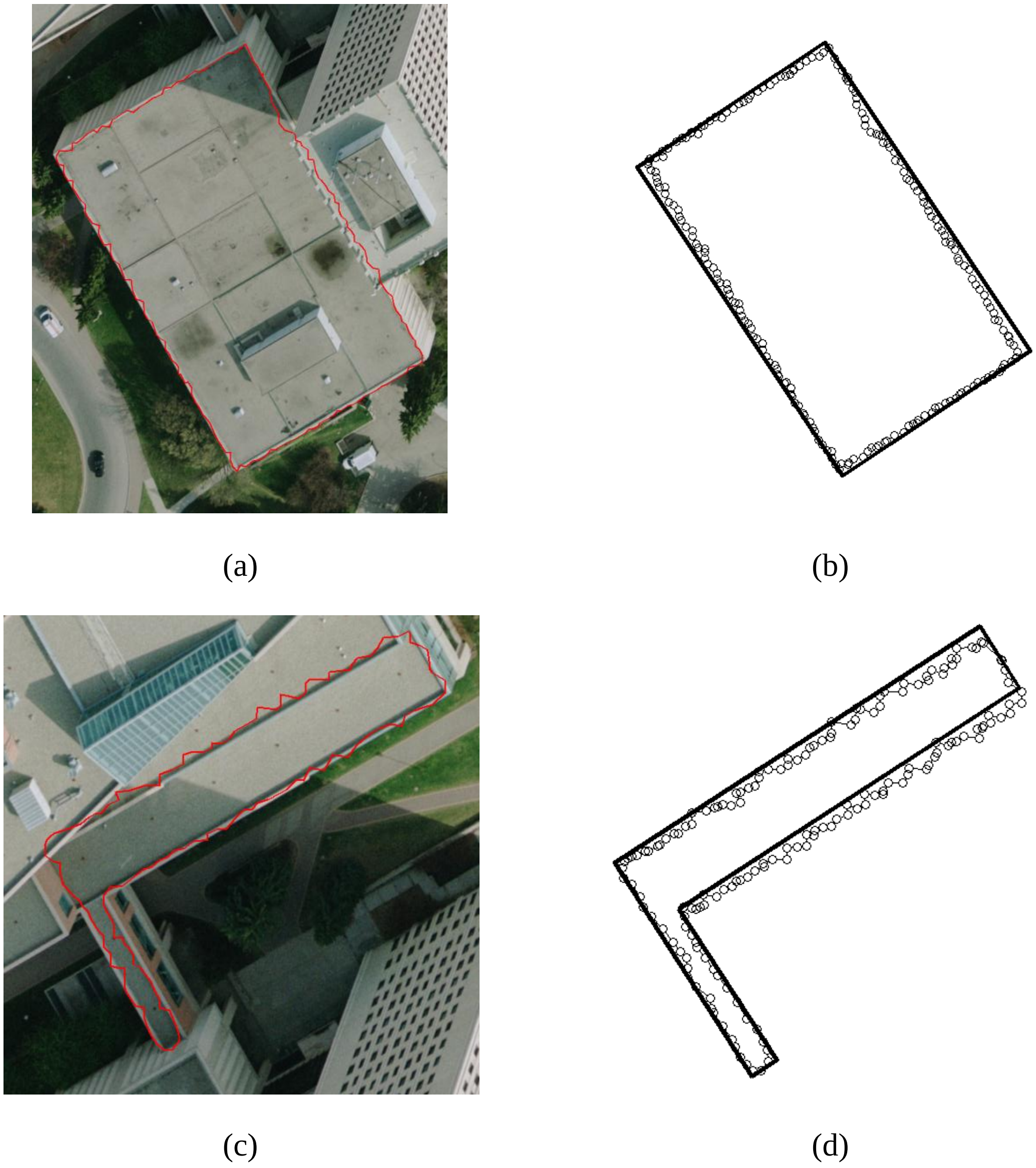
Figure 4.13: The final MBR of an irregular shape – non-building object: LiDAR boundary projected onto an image (a) and its MBR (b)



(a) (b)

Figure 4.14: The final MBR of an irregular shape – buildings that are not comprised of rectangles: LiDAR boundary projected onto an image (a) and its MBR (b)

When a manual interaction is involved, the MBRs derived from the false positive buildings and non-right-angled-corner buildings would be eliminated for the next step. Only rectangular primitives derived from the regular shape buildings shown in Figure 4.15 would have been incorporated with photogrammetric data.


(a) (b)

(c) (d)

Figure 4.15: The final MBRs of regular shapes that are comprised of rectangles: LiDAR boundaries projected onto an image (a), (c) and their MBRs (b), (d)

It is important to automatically discern if the buildings can be reconstructed using the proposed methodologies. However, a prompt manual evaluation process might not be possible, especially over large urban areas; and the lack of this confirmation will slow down the whole process. Therefore, in this research, an automatic process to determine the quality of the derived MBRs (i.e., if they are from regular shape buildings) is proposed.

Hence, the quality of the MBRs needs to be examined to verify that the buildings have regular shapes which include sets of rectangles. For this process, two criteria are proposed to evaluate the adequacy of the MBRs derived from LiDAR data. The assumption is that if the building is regular in shape, the similarity between the final MBR and the LiDAR derived boundary should be higher than the irregular-shaped ones. First, the areas of the LiDAR boundary and the final MBR are compared using Equation (4.2) to check the similarity of the two polygons.

$$Area\,ratio = \frac{Area\ of\ LiDAR\ boundary}{Area\ of\ final\ MBRs} \qquad \textbf{(4.2)}$$

The area of the polygons enclosed by n vertices (x_i, y_i) is given by Equation (4.3). Using this equation, the area of the LiDAR-derived boundary and the MBRs can be calculated. The MBRs with the area ratio close to 1 represents high similarity with the LiDAR boundary.

$$A = \frac{1}{2}\sum_{i=0}^{n-1}(x_i y_{i+1} - x_{i+1} y_i) \qquad \textbf{(4.3)}$$

Most of the MBRs with an area ratio close to 1 show similarity with the LiDAR boundaries; however, in some cases, the MBRs still may have areas similar to the LiDAR boundaries when their actual similarity is low. Figure 4.14(b) shows similar areas between the final MBR and the LiDAR boundary, but the MBRs do not represent the actual building boundary since the building is irregular in shape. Therefore, additionally, the Root Mean Squared Error (RMSE) of the distances between the LiDAR boundary points and the closest side of the MBR are used to check the similarity of the original and regularized boundaries. The MBRs with an RMSE less than twice the average point spacing of the dataset confirm that the final MBRs are close to the LiDAR boundaries.

Table 4.1 presents the area ratio and RMSE of the aforementioned examples in Figure 4.13, Figure 4.14, and Figure 4.15. The regular buildings have an area ratio close to 1 and RMSE values less than twice the average point spacing of the LiDAR data, (i.e., 0.88m) while the irregular buildings in Figure 4.13 show a low area ratio (i.e., less than 0.5) and high RMSE values (i.e., more than four times the average point spacing of the LiDAR data). As mentioned before, Figure 4.14 shows high area ratio, but the RMSE value is larger than the threshold. The values confirm that the proposed criteria can determine the adequacy of the MBRs to be used for the next step. Therefore, only MBRs with area ratios higher than 0.8 and RMSEs less than twice the average point spacing are declared as right-angled-corner buildings that are properly detected from LiDAR data.

Table 4.1: The criteria for the model adequacy evaluation

	Area ratio	RMSE (m)
Figure 4.13-irregular shape	0.49	4.18
Figure 4.14-irregular shape	0.90	1.80
Figure 4.15(a)-Regular shape	0.92	1.22
Figure 4.15(c)-Regular shape	0.98	1.14

4.5 Summary

This chapter provided the justification why a rectangular primitive is employed in this research. To reduce the number of model primitives while still being able to deal with any kinds of right-angled-corner buildings, simple box primitive is chosen as an initial model. Then, it is demonstrated that by utilizing a rectangular primitive together with their plane parameters from LiDAR data, instead of a box primitive, complex buildings that are comprised of rectangles with various slopes can be reconstructed.

In order to associate the LiDAR-derived boundaries from the data-driven approach in Chapter 3 with the model-based approach without human intervention, the MBR algorithm is introduced and modified to achieve the purpose of this research. The modified MBR algorithm generates a rectangle that best represent the LiDAR-derived boundary; a rectangle with the minimum area that has the considerable overlap with the investigated boundary. The modified MBR algorithm is applied in a recursive way to decompose the boundary into rectangular primitives automatically. As a result, different levels of MBRs are generated and they are used to derive the initial model parameters which will be refined through model-based image fitting in Chapter 5.

To automatically discern if the buildings can be reconstructed using the proposed methodologies (i.e., whether the investigated boundaries are derived from buildings with right-angled-corners), the quality of the MBRs need to be examined. Two criteria that can represent the similarity between the LiDAR-derived boundary and generated MBRs are proposed. Examples of the boundaries from the false positive buildings and non-right-angled-corner buildings were presented and compared to the ones from right-angled-corner buildings.

Chapter Five: **Model-based building reconstruction using photogrammetric data**

5.1 Introduction

In Chapter 3, initial building boundaries were detected from LiDAR data based on the data-driven approach. These boundaries were automatically decomposed into sets of rectangular primitives through the RMBR process in Chapter 4. Also, the quality of the rectangular primitives was evaluated to confirm that they are derived from the buildings that are regular in shape (i.e., buildings with right-angled-corners). As noted before, the horizontal location of the resulting boundaries needs to be improved. Therefore, in this chapter, the derived model primitives from the LiDAR data will be refined through a model-based approach to improve the horizontal accuracy of the boundary while keeping the LiDAR vertical accuracy at a high level by integrating image information. One of the traditional 3D building reconstruction techniques, which is based on photogrammetric data, matches the features derived either manually or automatically. Although matching-based reconstruction techniques produce high quality building boundaries, the matching ambiguities in multiple images remains the main obstacle to overcome. In this regard, the derived rectangular model primitives from the LiDAR data are incorporated to restrict the search space and resolve the matching ambiguities in the multiple images.

The main idea behind the model-based reconstruction is to describe the whole object using a predefined parametric model. Starting from the approximate values of the model parameters, those parameters are adjusted by minimizing the difference between the projected model, represented by the initial parameters, and its realization in the imagery. Realization of the model in the imagery is described by its bounding edges. This approach involves a recognition step to determine which model must be used and where in the image it is located (i.e., selection of the model primitives and their initial parameter determination). In this research, rectangular model primitives are used and their initial parameters are determined from the LiDAR data. Then, through a LSA fitting, the parameters will be refined to minimize the discrepancy between the projected model and the edges extracted from the images.

The main advantage of the proposed model-based reconstruction methodology, compared to the existing matching-based reconstruction method, is that it does not require a special procedure to identify any conjugate features. Therefore, even when partial occlusions exist in the images, which can be problematic in the case of the image-matching-based reconstruction, the model-based image fitting can still produce reliable results.

Since the LiDAR data information is incorporated with aerial images, which are from different sources, it must be preregistered to a common reference frame. Section 5.2 will discuss the registration of the two datasets. The details of the rectangular model parameters used in this research and the relationships between the involved coordinate systems will be presented in Section 5.3. The process to extract relevant edge information is discussed in Section 5.4. Section 5.5 explains the details of the model-based image fitting algorithm using LSA, and Section 5.6 proposes how to apply the algorithm sequentially to adjust multiple level MBRs.

5.2 Prerequisites

To make use of the synergetic properties of combined photogrammetric and LiDAR data, the following prerequisite processes should be performed: Quality Assurance (QA), Quality Control (QC) of photogrammetric and LiDAR data, and co-registration of both datasets.

QA activities are usually performed before the aerial mission to ensure that the quality of the service, item, and process meet the user's requirements, which is comprised of the management activities related to data collection and includes determination of the flying height and GPS base station distances. The key activity in QA is the calibration procedure. To achieve the desired quality of the final products, all the individual system components should be accurately calibrated and the relationship between the various system components also should be accurately estimated. For the photogrammetric system, the Interior Orientation Parameters (IOPs) are determined through a camera calibration process. The spatial and rotational offsets between various system components are derived through total system calibration. A typical LiDAR acquisition system is composed of a Global Positioning System (GPS), an Inertial Navigation System (INS), and a laser scanning system. The calibration of a LiDAR system aims at the estimation of systematic errors in the system parameters (e.g., lever arm offset and boresight angles). In this research, the camera and LiDAR systems were calibrated by adopting the approaches in Habib and Morgan (2005) and Habib et al. (2010), respectively.

QC is a post-mission procedure to verify whether the desired quality (i.e., data integrity, correctness, and completeness) was achieved. The photogrammetric QC process can be carried out by internal QC and external QC. Precision evaluation, internal QC, of the derived data can be attained based on the redundant measurements. External QC assesses the accuracy of the final product through check point analysis. Internal LiDAR QC can be implemented by evaluating the coincidence of the common features in overlapping strips. External LiDAR QC is a similar step to the photogrammetric check point analysis, but it requires specially-designed LiDAR targets (Habib et al., 2009).

Once the QA and QC of the photogrammetric and LiDAR data verify the quality of the datasets, the data can be used with confidence. Since two datasets from different sources are utilized, they must be registered to a common reference frame. Photogrammetric geo-referencing is the process of defining the camera's position and attitude relative to the object space coordinate system at the moment of exposure. LiDAR data can be utilized as a source of control for photogrammetric geo-referencing. Due to the irregular nature of LiDAR data, point-to-point correspondence between photogrammetric and LiDAR data is virtually impossible. Previous work by Habib et al. (2008) confirmed the feasibility of using the linear and areal features derived from LiDAR data as a source of control for photogrammetric geo-referencing. Therefore, in this research, the photogrammetric georeferencing method in Habib et al. (2008) is used. Straight lines and planar patches are extracted from LiDAR data in a semi-automatic way and used as the registration primitives for photogrammetric and LiDAR dataset integration. Point-based bundle adjustment processes with modified weight matrices are used to incorporate linear and planar features (Habib et al., 2008).

5.3 Model parameters and coordinate systems

The model-based approach involves three coordinate systems (model, object, and image coordinate systems). Transformation between the coordinate systems can be performed based on the associated parameters as seen in Figure 5.1. More specifically, the *shape* and *pose parameters* are used when transforming from the model to the object coordinate systems, and the exterior orientation parameters (EOP) and interior orientation parameters (IOP) are utilized when transforming from the object to the image coordinate systems. The following subsections will discuss the parameters of the rectangular primitives and the relationships between the coordinate systems.

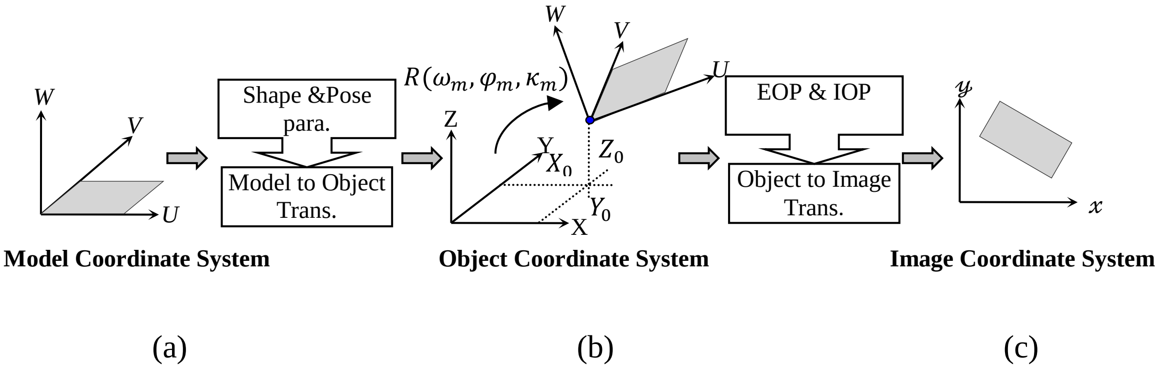


Figure 5.1: The involved coordinate systems in the model-based approach and their relationships: model coordinate system (a), object coordinate system (b), and image coordinate system (c)

5.3.1 Model parameters and model to object coordinate system transformation

A model has its own coordinate system (U, V, W) and the origin is defined at a reference point (e.g., the lower left corner point on a building rooftop) shown as a blue circle in Figure 5.2. The U and V axes are aligned along the sides of the rectangle, and the W axis is normal to the surface of the rectangle. d_l and d_w are the shape parameters of the rectangular model primitive as introduced in Chapter 4.

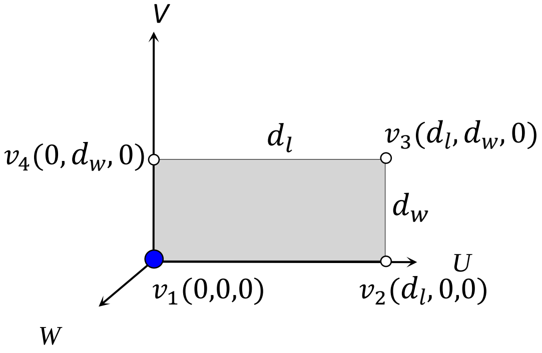


Figure 5.2: Definition of the rectangular model coordinate system

The object coordinate system is the reference coordinate system used for the LiDAR building detection in Chapter 3. X_0, $Y_{0,}$ Z_0 refer to the position of the reference point on the rectangle (i.e., the origin of the model coordinate system, vertex 1 (v_1)) in object space. The ω_m, φ_m, κ_m parameters are the angles that need to be applied to the object coordinate system in order for the object coordinate system to become parallel to the model coordinate system. The relationship between the coordinates of the rectangle vertices in the model and object coordinates can be expressed according to Equation (5.1). It is assumed that the rotation angles (ω_m, φ_m, κ_m) are applied in the same sequence as in traditional photogrammetric procedures (Kraus, 1993). Table 5.1 lists the four vertices of a rectangle in the model and object coordinate systems.

$$\begin{bmatrix} X \\ Y \\ Z \end{bmatrix}_{object} = \begin{bmatrix} X_0 \\ Y_0 \\ Z_0 \end{bmatrix}_{object} + \mathrm{R}_{model}^{object}(\omega_m, \varphi_m, \kappa_m) \begin{bmatrix} U \\ V \\ W \end{bmatrix}_{model} \quad \textbf{(5.1)}$$

Table 5.1: Vertices of a rectangle in the model and object coordinate systems

Vertex	Model coordinates	Transformed coordinates in object coordinate system
v_1	(0, 0, 0)	(X_0, Y_0, Z_0)
v_2	$(d_l, 0, 0)$	$(X_0 + d_l \mathrm{r}_{11}, Y_0 + d_l \mathrm{r}_{21}, Z_0 + d_l \mathrm{r}_{31})$
v_3	$(d_l, d_w, 0)$	$(X_0 + d_l \mathrm{r}_{11} + d_w \mathrm{r}_{12}, Y_0 + d_l \mathrm{r}_{21} + d_w \mathrm{r}_{22}, Z_0 + d_l \mathrm{r}_{31} + d_w \mathrm{r}_{32})$
v_4	$(0, d_w, 0)$	$(X_0 + d_w \mathrm{r}_{12}, Y_0 + d_w \mathrm{r}_{22}, Z_0 + d_w \mathrm{r}_{32})$

As mentioned in Chapter 3, precise plane information (i.e., the plane parameters a, b, c) for each building's rooftop is available from the LiDAR plane segmentation. Therefore, Z_0 can be expressed as a function of the plane parameters and the X_0, Y_0 coordinates of the reference point according to Equation (5.2).

$$Z_0 = \mathrm{f}(X_0, Y_0, \text{plane paramters}) = aX_0 + bY_0 + c \qquad \textbf{(5.2)}$$

Using rectangular primitives, gable-roof buildings and shed-roof buildings can be modeled without the necessity of defining them in advance as different models.

Previous literature (Suveg and Vosselman, 2000; Tseng and Wang, 2003; Vosselman and Veldhuis, 1999) only use a rotation angle around the Z-axis (κ_m), and ω_m and φ_m are not considered, assuming that that buildings stand vertical with horizontal rooftops. For more general building models (i.e., not only horizontal rooftops, but also tilted rooftops), the ω_m and φ_m angles can be defined with high accuracy using the orientation of the normal to the planar surface. The relationship between the normal to the plane in the model coordinate system (0, 0, 1) and the object space (n_x, n_y, n_z) can be expressed using Equation (5.3).

$$\begin{bmatrix} n_x \\ n_y \\ n_z \end{bmatrix}_{object} = R_{model}^{object}(\omega_m, \varphi_m, \kappa_m)\begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}_{model} = \begin{bmatrix} \mathrm{r}_{11} & \mathrm{r}_{12} & \mathrm{r}_{13} \\ \mathrm{r}_{21} & \mathrm{r}_{22} & \mathrm{r}_{23} \\ \mathrm{r}_{31} & \mathrm{r}_{32} & \mathrm{r}_{33} \end{bmatrix}\begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}_{model} \qquad \textbf{(5.3)}$$

where,

$\mathrm{r}_{11} = \cos\varphi_m \cos\kappa_m$

$\mathrm{r}_{12} = -\cos\varphi_m \sin\kappa_m$

$\mathrm{r}_{13} = \sin\varphi_m$

$\mathrm{r}_{21} = \cos\omega_m \sin\kappa_m + \sin\omega_m \sin\varphi_m \cos\kappa_m$

$\mathrm{r}_{22} = \cos\omega_m \cos\kappa_m - \sin\omega_m \sin\varphi_m \sin\kappa_m$

$\mathrm{r}_{23} = -\sin\omega_m \cos\varphi_m$

$\mathrm{r}_{31} = \sin\omega_m \sin\kappa_m - \cos\omega_m \sin\varphi_m \cos\kappa_m$

$\mathrm{r}_{32} = \sin\omega_m \cos\kappa_m + \cos\omega_m \sin\varphi_m \sin\kappa_m$

$\mathrm{r}_{33} = \cos\omega_m \cos\varphi_m$

The relationship between the normal to the plane in the object space (n_x, n_y, n_z) and the plane parameter (a, b, c from Equation (5.2)) is expressed in Equation (5.4). From Equation (5.4), φ_m can be derived as $\arcsin(n_x)$ and ω_m can be derived as $\arctan(-\mathrm{n_y}/\ \mathrm{n_z})$.

$$\begin{bmatrix} n_x \\ n_y \\ n_z \end{bmatrix}_{object} = \begin{bmatrix} \frac{-a}{\sqrt{a^2+b^2+1}} \\ \frac{-b}{\sqrt{a^2+b^2+1}} \\ \frac{1}{\sqrt{a^2+b^2+1}} \end{bmatrix} = \begin{bmatrix} \mathrm{r}_{13} \\ \mathrm{r}_{23} \\ \mathrm{r}_{33} \end{bmatrix} = \begin{bmatrix} \sin\varphi_m \\ -\sin\omega_m\ \cos\varphi_m \\ \cos\omega_m\ \cos\varphi_m \end{bmatrix} \qquad \textbf{(5.4)}$$

In summary, among the aforementioned *pose* (X_0, Y_0, Z_0, ω_m, φ_m, κ_m) and *shape* (d_w, d_l) parameters, ω_m and φ_m are accurately defined by the normal to the surface of the building rooftop plane from the LiDAR segmentation process, and Z_0 is defined as a function of X_0, Y_0, and the plane parameters. Only approximate values are available from the previous step for the remaining parameters. Thus, the model parameters in need of refinement are: X_0, Y_0, κ_m, d_w, and

d_l as illustrated in Figure 5.3. These five parameters are estimated by integrating information from images, where the building in question appears through the model-based image fitting. One should note that a good estimation of the initial values of the five parameters is already available from LiDAR data segmentation and the RMBR procedure.

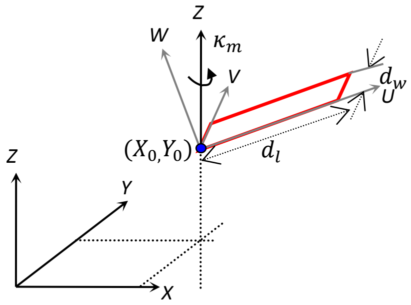


Figure 5.3: Rectangular primitive and related parameters used in this research

5.3.2 Object to image coordinate system transformation

The rectangular primitive in the object space is transformed into an image coordinate system to integrate image information as discussed in the previous section. The EOPs and IOPs of the images are derived from the registration process and camera calibration, respectively. The vertices of the rectangle can be projected onto the image using the collinearity equations with the derived EOP and IOP according to Equation (5.5).

$$x_i = x_p - c\frac{r_{11}^e(X_i - X_e) + r_{21}^e(Y_i - Y_e) + r_{31}^e(Z_i - Z_e)}{r_{13}^e(X_i - X_e) + r_{23}^e(Y_i - Y_e) + r_{33}^e(Z_i - Z_e)} + \Delta x \; + e_x$$
$$= f_x(X_i, Y_i, Z_i, EOP, IOP) + e_x$$

(5.5)

$$y_i = y_p - c\frac{r_{12}^e(X_i - X_e) + r_{22}^e(Y_i - Y_e) + r_{32}^e(Z_i - Z_e)}{r_{13}^e(X_i - X_e) + r_{23}^e(Y_i - Y_e) + r_{33}^e(Z_i - Z_e)} + \Delta y + e_y$$
$$= f_y(X_i, Y_i, Z_i, EOP, IOP) + e_y$$

where,

x_i, y_i: projected image coordinates of vertex i

X_i, Y_i, Z_i : coordinates of vertex i in object coordinate system as shown in Table 5.1

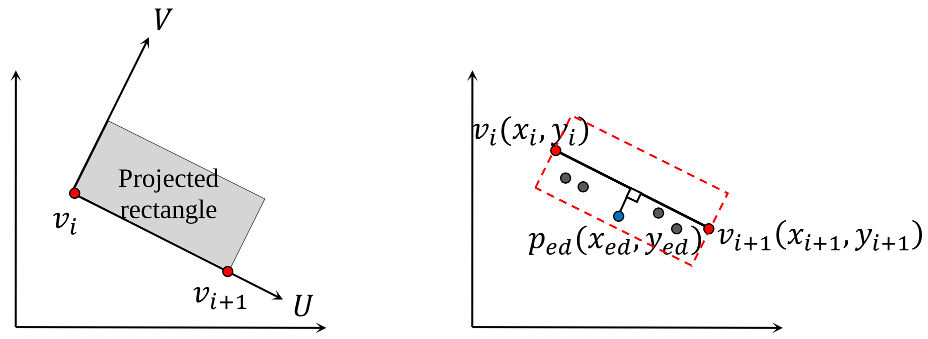
x_p, y_p, c: image coordinates of principal point and principal distance

$\Delta x, \Delta y$: contribution of the distortions

X_e, Y_e, Z_e: the object coordinates of the perspective center

$r_{11}^e \cdots r_{33}^e$: the elements of the rotation matrix that depend on the rotation angles (ω_e, φ_e, κ_e)

Figure 5.4(a) depicts the projected rectangular primitive under consideration into the imagery using the provided EOP, IOP, and initial model parameters. The initial model parameters are acquired from the LiDAR data segmentation and RMBR process so the initial projection should be close to the actual outlines of the building roof on the aerial photo.



(a) (b)

Figure 5.4: Projected rectangular model primitive onto the image space (a) and the effective buffer zone for one side of the rectangle ($v_i v_{i+1}$) (b)

Based on the initial projection of the model onto the images, a buffer is set around each side of the rectangle and only the edge pixels within the buffer are utilized for the next step. Figure 5.4(b) shows the selected edges within the buffer. The model parameters will be updated through a LSA, which will minimize the distance between the projected model and the extracted edges from the imagery. More details of the edge detection process are discussed in the next section.

5.4 Edge extraction

All the extracted edges from the images do not necessarily represent the rooftop boundaries due to shadows, noise, or irrelevant features near or on the roofs. Since using incorrect edges will lead to inaccurate model parameters, filtering irrelevant edge information is important. Figure 5.5 shows the procedure to extract only the relevant edge pixels from the images for the model-based image fitting process. The discrete edge points are first detected by applying the Canny edge detector (Canny, 1986) to all the images in which the building appears. The Canny edge detector is one of the most common edge detection methods used in the image processing research, which theoretically produces thin edges close to the true edges. After a buffer is defined around the initial boundary of each rectangular primitive in the images, only the edge points that are inside the defined buffer zone are considered as candidate edge pixels. Figure 5.6(a) shows an example of the initial rectangular model projected onto one of the images it appears in.

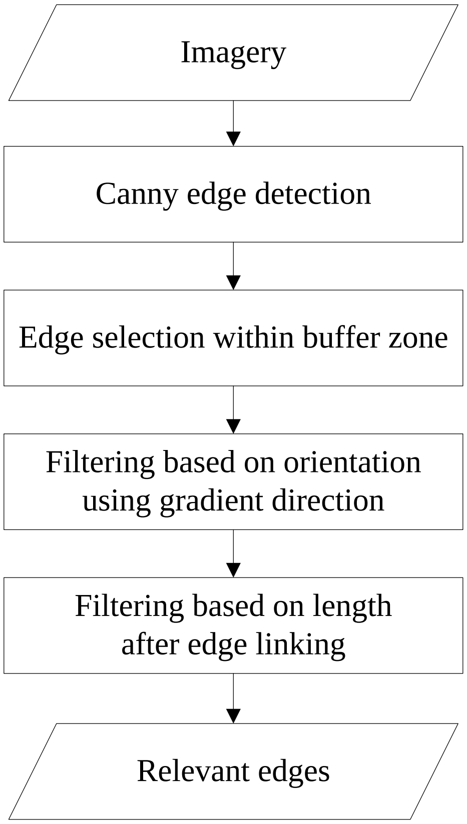

Imagery
Canny edge detection
Edge selection within buffer zone
Filtering based on orientation using gradient direction
Filtering based on length after edge linking
Relevant edges

Figure 5.5: Edge extraction and filtering process

Ideally, the Canny edge detector should yield sets of pixels that are one pixel thick, lying on building boundaries; however, as seen in Figure 5.6(b), it produces unnecessary edges near the building boundary because of noise, non-uniform illuminations, shadows, and other objects. To exclude irrelevant edges, the position and orientation information of the initial model are incorporated. Since a decent approximation of the derived rectangular primitive from the LiDAR data is available, only the edges within the buffer around the projected model and whose orientation is similar to the orientation of the sides of the projected model are selected. The gradient angle of the extracted edges is compared with the angle of the normal vector of the initial model line (i.e., each side of the rectangle) using Equation (5.6). Note that the direction of the edge is perpendicular to the gradient vector.

$$|\alpha(x, y) - \alpha(\mathrm{line}_n)| \leq A \tag{5.6}$$

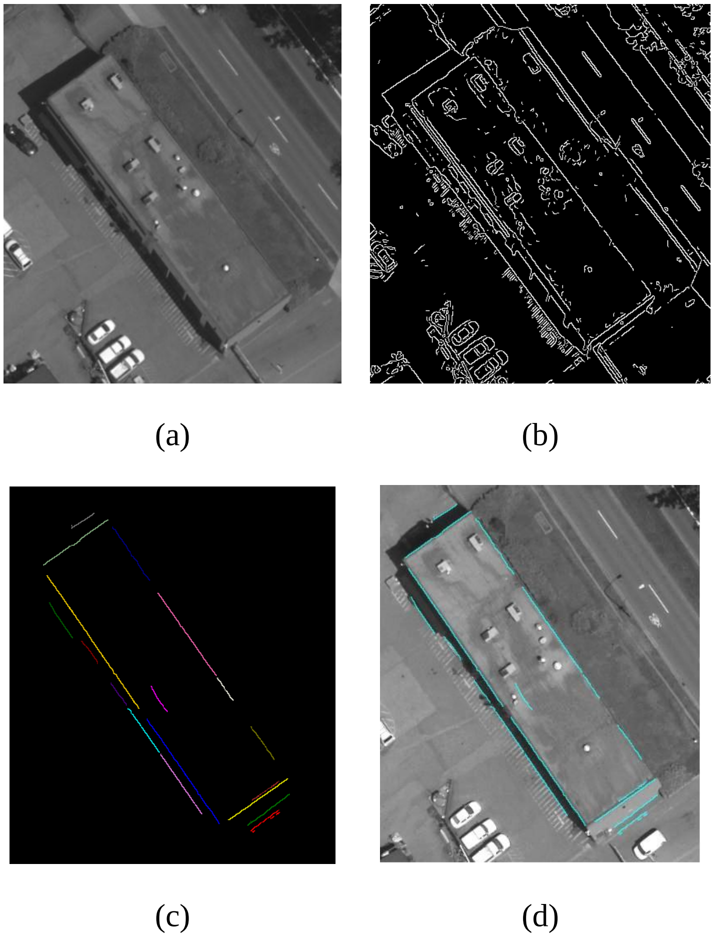
where,

$\alpha(x, y)$ is the direction angle of the gradient vector of an edge pixel at (x, y)

$\alpha(\mathrm{line}_n)$ is the angle of the normal vector of the initial rectangular model line (i.e., side of rectangle)

A is a positive angle threshold

After that, to remove unnecessary small segments, edge linking is performed on the edge pixels, and segments whose lengths are smaller than a given threshold are removed. This process is applied to all the images where the building primitive appears. Figure 5.6(c) shows the edge linking results where different colors represent different linked edge groups. In summary, only the edge pixels within the set buffer, which have similar orientation to the projected model and whose lengths are longer than a predefined threshold, are used for the adjustment as seen in Figure 5.6(d). When compared with Figure 5.6(b) (detected edges using the Canny operator), unnecessary edge information is removed.



(a) (b)

(c) (d)

Figure 5.6: Aerial photo over the area of interest (a), Canny edge detection (b), filtered/linked edges where different grouped lines are shown in different colors (c), and the filtered / linked edges displayed on the image (d)

5.5 Model-based image fitting

This section addresses how a model-based image fitting algorithm adjusts the model parameters using the extracted edge information from the corresponding images. The approximate model parameters are simultaneously refined through a LSA procedure using all the images in which the building rooftop in question appears. This is achieved through observation equations that minimize the normal distance between the sides of the projected model and the detected edge pixels within the buffer around the sides. This constraint is applied to all sides of the rectangle in all the images while refining the model parameters. While the concept of the LSA with normal distance minimization is straightforward, its implementation is complex and will be discussed in Subsection 5.5.1. Therefore, for simplicity in the implementation, modification to the traditional Gauss Markov stochastic model is incorporated by assuming point-based observation. The details of this implementation will be described in Subsection 5.5.2.

5.5.1 LSA based on the normal distance constraint

Figure 5.7 depicts one side of the initial rectangle derived from the LiDAR data and the extracted edges around it. The normal distance (d_n) from one edge pixel (x_{ed}, y_{ed}) to a side of the rectangle, which is the projected line with end points ($x_{\mathrm{i}}, y_{\mathrm{i}}$) and ($x_{\mathrm{i+1}}, y_{\mathrm{i+1}}$), is formulated in Equation (5.7) as a function of the provided EOPs, IOPs, and initial model parameters. In this case, observations are EOPs, IOPs, ω_m, φ_m, and plane parameters (a, b, c). Table 5.1 presented the object coordinates of the four corner points (X_i, Y_i, Z_i) expressed as a function of the model parameters.

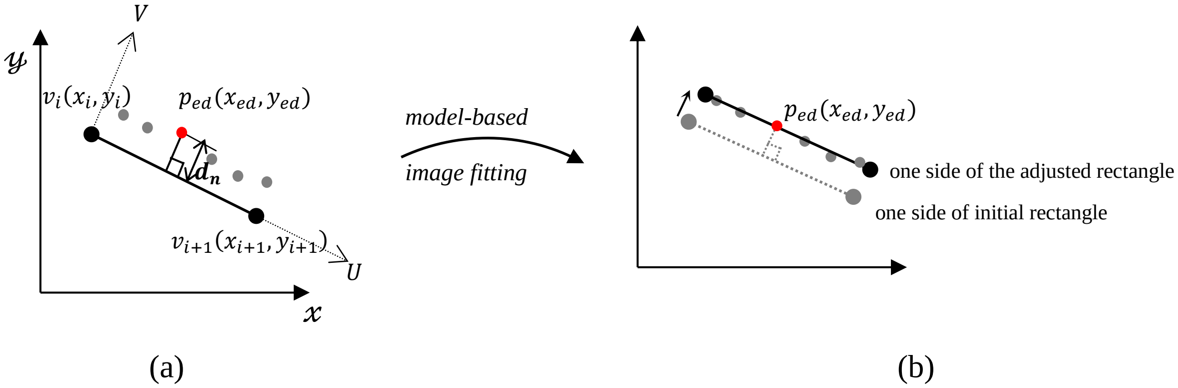


Figure 5.7: Normal distance between the edge pixel and one side of the initial rectangle (a) and the adjusted side of the rectangle after the model-based image fitting (b)

$$d_n = \frac{|(y_{\mathrm{i}} - y_{\mathrm{i+1}})x_{ed} + (x_{\mathrm{i+1}} - x_{\mathrm{i}})y_{ed} + (x_{\mathrm{i}}y_{\mathrm{i+1}} - x_{\mathrm{i+1}}y_{\mathrm{i}})|}{\sqrt{(x_{\mathrm{i+1}} - x_{\mathrm{i}})^2 + (y_{\mathrm{i+1}} - y_{\mathrm{i}})^2}} = 0$$

where, $(x_{\mathrm{i}}, y_{\mathrm{i}}) = f(X_{\mathrm{i}}, Y_{\mathrm{i}}, Z_{\mathrm{i}}, EOPs, IOPs)$, **(5.7)**
$(X_{\mathrm{i}}, Y_{\mathrm{i}}, Z_{\mathrm{i}}) = f(X_0, Y_0, d_w, d_l, \kappa_m)$,
$(x_{\mathrm{i+1}}, y_{\mathrm{i+1}}) = f(X_{\mathrm{i+1}}, Y_{\mathrm{i+1}}, Z_{\mathrm{i+1}}, EOPs, IOPs)$,
$(X_{\mathrm{i+1}}, Y_{\mathrm{i+1}}, Z_{\mathrm{i+1}}) = f(X_0, Y_0, d_w, d_l, \kappa_m)$,

Since the normal distance is a non-linear function between the unknown parameters and observations, it should be linearized using Taylor series expansion which can be expressed using the Gauss Helmert model, also known as the general case of least squares estimation as seen in Equation (5.8).

$$y = Ax + Be \quad where\ \ e \sim (0, \Sigma)\ and\ \ \Sigma = \sigma_o^2 P^{-1} \tag{5.8}$$

The Gauss Helmert model can be transformed into the Gauss Markov model with the appropriate variable substitution (Koch, 1999), which can be written as Equation (5.9).

$$y = Ax + e \qquad where\ \ e \sim (0, \Sigma)\ and\ \ \Sigma = \sigma_o^2 (BP^{-1}B^T) \tag{5.9}$$

In this equation, y represents the calculated normal distance using the approximate values of the model parameters, x is the corrections to the initial values, and the design matrix (A) is derived using the partial derivatives of d_n with respect to the unknown model parameters. e is the random error which is normally distributed with a zero mean and variance-covariance matrix, Σ. The statistical properties of the e are expressed using the variance-covariance matrix (Σ) derived through the law of error propagation using the partial parameters with respect to the observations (B) in Equation (5.7).

The LSA procedure aims at estimating the unknown parameters, which minimize the sum of squares of the weighted residuals ($e^T Pe$), and it iterates until the difference in the corrections is smaller than a set threshold. It is worth mentioning that the buffer size around the sides of the projected rectangle for the edge detection and filtering decreases as the solution is converging. The traditional least-squares solution is shown in Equation (5.10) (Mikhail, 1976). In addition, the stochastic properties of the estimated parameters can be derived through the variance-covariance matrix of the adjusted parameters as seen in Equation (5.11). Also, the overall quality of the adjustment can be evaluated using the a-posteriori variance factor shown in Equation (5.12).

$$\hat{x} = (A^T PA)^{-1} A^T Py = N^{-1} C \qquad \text{(Solution Vector)} \qquad \mathbf{(5.10)}$$

$$\Sigma\{\hat{x}\} = \hat{\sigma}_o^2 (A^T PA)^{-1} = \hat{\sigma}_o^2 N^{-1} \qquad \text{(Variance-Covariance Matrix)} \qquad \mathbf{(5.11)}$$

$$\hat{\sigma}_o^2 = (\tilde{e}^T P \tilde{e})/(n - m) \qquad \text{(A-posteriori Variance Factor)} \qquad \mathbf{(5.12)}$$

where, n is the number of extracted and filtered edge pixels, and m is the number of unknown parameters

As mentioned, the design matrix A is derived using partial derivatives for d_n with respect to the unknown parameters as seen in Equation (5.13). This can be derived using the Chain rule and an example of the derivation is shown in Equation (5.14).

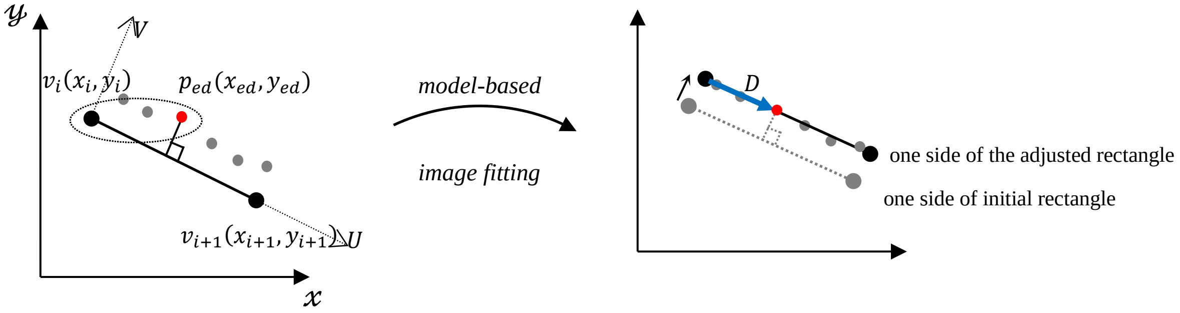
$$\mathrm{A}_{1\times 5} = \left[\frac{\partial d_n}{\partial X_0} \; \frac{\partial d_n}{\partial Y_0} \; \frac{\partial d_n}{\partial d_w} \; \frac{\partial d_n}{\partial d_l} \; \frac{\partial d_n}{\partial \kappa}\right] \qquad \mathbf{(5.13)}$$

$$
\begin{aligned}
\frac{\partial d_n}{\partial X_0} = \frac{\partial d_n}{\partial x_i}\cdot\frac{\partial f_{x_i}}{\partial X_i}\cdot\frac{\partial X_i}{\partial X_0} + \frac{\partial d_n}{\partial x_i}\cdot\frac{\partial f_{x_i}}{\partial Y_i}\cdot\frac{\partial Y_i}{\partial X_0} + \frac{\partial d_n}{\partial x_i}\cdot\frac{\partial f_{x_i}}{\partial Z_i}\cdot\frac{\partial Z_i}{\partial X_0} \\
+ \frac{\partial d_n}{\partial y_i}\cdot\frac{\partial f_{y_i}}{\partial X_i}\cdot\frac{\partial X_i}{\partial X_0} + \frac{\partial d_n}{\partial y_i}\cdot\frac{\partial f_{y_i}}{\partial Y_i}\cdot\frac{\partial Y_1}{\partial X_0} + \frac{\partial d_n}{\partial y_i}\cdot\frac{\partial f_{y_i}}{\partial Z_i}\cdot\frac{\partial Z_i}{\partial X_0} \\
+ \frac{\partial d_n}{\partial x_{i+1}}\cdot\frac{\partial f_{x_{i+1}}}{\partial X_{i+1}}\cdot\frac{\partial X_{i+1}}{\partial X_0} + \frac{\partial d_n}{\partial x_{i+1}}\cdot\frac{\partial f_{x_{i+1}}}{\partial Y_{i+1}}\cdot\frac{\partial Y_{i+1}}{\partial X_0} + \frac{\partial d_n}{\partial x_{i+1}}\cdot\frac{\partial f_{x_{i+1}}}{\partial Z_{i+1}}\cdot\frac{\partial Z_{i+1}}{\partial X_0} \\
+ \frac{\partial d_n}{\partial y_{i+1}}\cdot\frac{\partial f_{y_{i+1}}}{\partial X_{i+1}}\cdot\frac{\partial X_{i+1}}{\partial X_0} + \frac{\partial d_n}{\partial y_{i+1}}\cdot\frac{\partial f_{y_{i+1}}}{\partial Y_{i+1}}\cdot\frac{\partial Y_{i+1}}{\partial X_0} + \frac{\partial d_n}{\partial y_{i+1}}\cdot\frac{\partial f_{y_{i+1}}}{\partial Z_{i+1}}\cdot\frac{\partial Z_{i+1}}{\partial X_0}
\end{aligned}
\qquad \textbf{(5.14)}
$$

Due to the complexity of the partial derivatives using the normal distance minimization, alternative implementation utilizing modified LSA is employed.

5.5.2 Modified LSA

As mentioned above, the normal distance minimization constraint makes LSA implementation complex. Therefore, in this research, an alternative implementation, modified LSA, is employed. To minimize the normal distance between the projected rectangular model and the extracted and filtered edges, the corner points of the adjusted rectangle should be aligned along the line connecting the edges as seen in Figure 5.8(b). When one assumes that the detected and filtered edge pixels and the vertex point of the projected rectangle as conjugate-points, the vertex point (v_i) and the edge points(p_{ed}) are not corresponding to each other and have a discrepancy vector D between them. Therefore, these points will be referred to as pseudo-conjugate points. Each edge pixel after the linking and filtering process, p_{ed}, contributes one set of observation equations assuming that one of the vertex points of rectangle (v_i) and edge pixel (p_{ed}) are conjugate.


y
V
$v_i(x_i, y_i)$
$p_{ed}(x_{ed}, y_{ed})$
$v_{i+1}(x_{i+1}, y_{i+1})$
U
x
model-based
image fitting
D
one side of the adjusted rectangle
one side of initial rectangle

(a) (b)

Figure 5.8: Pseudo-conjugate points (i.e., the vertex of the rectangle and an edge pixel) (a) and the additional unknown vector D (b)

The mathematical model describing the discrepancy between the pseudo conjugate points (x_{ed}, y_{ed}) and (x_i, y_i) will take the form in Equation (5.15).

$$x_{ed} = x_i + e_x + D_x = f_x(X_i, Y_i, Z_i, EOP, IOP) + e_x + D_x$$
$$y_{ed} = y_i + e_y + D_y = f_y(X_i, Y_i, Z_i, EOP, IOP) + e_y + D_y \qquad \textbf{(5.15)}$$

where,

D_x and D_y are the components of the discrepancy vector (D) between the adjusted vertex after model-image fitting ($v_{i(adjusted)}(x_i, x_i)$) and the edge point ($p_{ed}(x_{ed}, y_{ed})$) in the x and y directions, respectively.

The stochastic model describing the mathematical relationship among the pseudo-conjugate points can be represented by the Gauss Markov stochastic model in Equation (5.16).

$$y = Ax + D + e \quad e \sim (0, \Sigma) \quad where\ \Sigma = \sigma_o^2 P^{-1} \qquad \textbf{(5.16)}$$

The design matrix A is derived through the partial derivatives of f_x, $f_y$with respect to the unknown parameters as seen in Equation (5.17), and an example for the chain rule can be seen in Equation (5.18). It can be seen that the populated A matrix gets simpler than the A matrix from the previous implementation.

$$\mathrm{A}_{2\times5} = \begin{bmatrix} \frac{\partial f_x}{\partial X_0} & \frac{\partial f_x}{\partial Y_0} & \frac{\partial f_x}{\partial d_w} & \frac{\partial f_x}{\partial d_l} & \frac{\partial f_x}{\partial \kappa} \\ \frac{\partial f_y}{\partial X_0} & \frac{\partial f_y}{\partial Y_0} & \frac{\partial f_y}{\partial d_w} & \frac{\partial f_y}{\partial d_l} & \frac{\partial f_y}{\partial \kappa} \end{bmatrix} \quad \textbf{(5.17)}$$

$$\frac{\partial f_x}{\partial X_0} = \frac{\partial f_x}{\partial X_i}\cdot\frac{\partial X_i}{\partial X_0} + \frac{\partial f_x}{\partial Y_i}\cdot\frac{\partial Y_i}{\partial X_0} + \frac{\partial f_x}{\partial Z_i}\cdot\frac{\partial Z_i}{\partial X_0} \quad \textbf{(5.18)}$$

Appendix A shows the partial derivatives of collinearity equations and the partial derivatives of each rectangle vertex with respect to each parameter.

The modified LSA is introduced in Habib et al. (2010) to consider the model in Equation (5.16) without estimating the additional unknown vector (D) during the LSA. This can be achieved by modifying stochastic properties of the random errors to satisfy the condition ($\mathrm{P}'\mathrm{D} = 0$) as presented in Equation (5.19). This condition implies that the modified weight matrix P' is not positive-definite; thus, the inverse matrix P'^{-1} does not exist. Therefore, the modified variance-covariance matrix will be represented as Equation (5.19).

$$\Sigma'\{e\} = \sigma_o^2\, P'^{+} \text{ where } P'D = 0 \quad \textbf{(5.19)}$$

where, P'^{+} is the Moore-Penrose pseudo inverse of P'

The LSA target function can be rewritten as Equation (5.20) using the modified weight matrix (P'). By incorporating the fact that $P'D = 0$, the LSA target function reduces to the form in Equation (5.21), which can be expanded to Equation (5.22).

$$e^T P' e = (y - Ax - D)^T P' (y - Ax - D) = min|_{x,D} \qquad \text{(LSA Target function)} \qquad \textbf{(5.20)}$$

$$e^T P' e = (y - Ax)^T P' (y - Ax) = min|_x \qquad \text{(LSA Target Function)} \qquad \textbf{(5.21)}$$

$$\phi(x) = y^T P' y + x^T A^T P' Ax - 2x^T A^T P' y = min|_x \qquad \text{(LSA Target Function)} \qquad \textbf{(5.22)}$$

The solution vector ($\hat{x}$) that minimizes $\phi(x)$ can be obtained by differentiating $\phi(x)$ with respect to x and equaling it to zero (Equation (5.23)). Thus, the solution ($\hat{x}$) to the LSA target function is defined by Equation (5.24). One should note that although the modified weight matrix (P') is not a full rank matrix, the solution is available due to the large number of observations. Using the law of error propagation, the variance-covariance matrix of the solution vector, $\Sigma\{\hat{x}\}$, can be obtained as in Equation(5.25).

$$\frac{\partial\phi}{\partial x} = 2A^T P' Ax - 2A^T P' y = 0 \qquad \textbf{(5.23)}$$

$$\hat{x} = (A^T P' A)^{-1} A^T P' y = N^{-1} A^T P' y \text{ (Estimated Unknowns)}$$
$$\text{where } N = A^T P' A \qquad \textbf{(5.24)}$$

$$\Sigma\{\hat{x}\} = \sigma_o^2 N^{-1} \text{ (Variance-Covariance Matrix)} \qquad \textbf{(5.25)}$$

The a-posteriori variance factor ($\hat{\sigma}_o^2$) for the modified LSA can be derived as follows. The expected value of the sum of squares of the weighted predicted residuals in Equation (5.26) can be reorganized to produce the form in Equation (5.27) by substituting the derived solution for $\hat{x}$ from Equation (5.24). Equation (5.27) can be manipulated as seen in Equation (5.28) based on the basic properties of the trace; the trace of a scalar equals to the scalar (i.e., $tr(S) = S)$ and the trace operation is commutative (i.e., $tr(AB) = tr(BA)$) (Koch, 1988). Equation (5.28) yields Equation (5.29) using the properties that $tr(A) + tr(B) = tr(A+B)$ and $E\{tr(A)\} = tr\{E(A)\}$ (Koch, 1988).

$$E(\tilde{e}^T P' \tilde{e}) = E\{(y - A\hat{x} - D)^T P' (y - A\hat{x} - D)\} = E\{(y - A\hat{x})^T P' (y - A\hat{x})\} \tag{5.26}$$

$$E(\tilde{e}^T P' \tilde{e}) = E\{y^T P' y - y^T P' A N^{-1} A^T P' y\} \tag{5.27}$$

$$E(\tilde{e}^T P' \tilde{e}) = E\{tr(P' y y^T) - tr(P' A N^{-1} A^T P' y y^T)\} \tag{5.28}$$

$$\mathrm{E}\left(\tilde{\mathrm{e}}^{\mathrm{T}} \mathrm{P}^{'} \tilde{\mathrm{e}}\right) = \mathrm{trP}^{'}\left[\mathrm{E}(\mathrm{yy}^{\mathrm{T}}) - \mathrm{AN}^{-1}\mathrm{A}^{\mathrm{T}}\mathrm{P}^{'}\mathrm{E}(\mathrm{yy}^{\mathrm{T}})\right] = \mathrm{trP}^{'}\left(\mathrm{I}_{\mathrm{n}} - \mathrm{AN}^{-1}\mathrm{A}^{\mathrm{T}}\mathrm{P}^{'}\right)\mathrm{E}(\mathrm{yy}^{\mathrm{T}}) \\ \text{where, } \mathrm{I}_{\mathrm{n}} \text{ is an } nxn \text{ identity matrix} \tag{5.29}$$

The variance-covariance matrix of the observations vector, $\Sigma\{y\}$, can be defined as seen in Equation (5.30) and therefore the term $E(yy^T)$ can be derived according to Equation (5.31).

$$\Sigma\{y\} = \sigma_o^2 P'^{+} = E\{(y - Ax - D)(y - Ax - D)^T\} \tag{5.30}$$

$$\begin{aligned} E(yy^T) &= \sigma_o^2 P'^{+} + (Ax + D)(Ax + D)^T \\ &= \sigma_o^2 P'^{+} + Axx^T A^T + AxD^T + Dx^T A^T + DD^T \end{aligned} \tag{5.31}$$

Equation (5.29) can be simplified to Equation (5.32) by substituting $E(yy^T)$ in Equation (5.29) with Equation (5.31) and using the fact that $P^{'}D = 0$. Equation (5.33) can be stated based on the properties that $tr(AB) = rank(AB)$ (given that AB is idempotent) and $rank(AB) \leq \min(rankA, rankB)$ (Kock, 1988). Since $tr(P^{'}P^{'+}) = q$ (rank of the modified matrix as shown in Equation (5.33)) and $P'P'^{+}P' = P'$ (where P'^{+} is Moore–Penrose pseudo inverse), Equation (5.32) can be simplified to Equation (5.34). As a result, the a-posteriori variance factor ($\hat{\sigma}_o^2$) can be derived according to Equation (5.35).

$$E(\tilde{e}^T P' \tilde{e}) = \sigma_o^2 trP'(I - AN^{-1}A^T P')P'^{+} = \sigma_o^2 trP'P'^{+} - \sigma_o^2 trN^{-1}A^T P'P'^{+}P'A \qquad \textbf{(5.32)}$$

$$tr(P'P'^{+}) = rank(P'P'^{+}) = min(rankP', rankP'^{+}) = rankP' = q \qquad \textbf{(5.33)}$$

$$E(\tilde{e}^T P' \tilde{e}) = \sigma_o^2 q - \sigma_o^2 trN^{-1}N = \sigma_o^2 q - \sigma_o^2 trI_m = \sigma_o^2 q - \sigma_o^2 m \qquad \textbf{(5.34)}$$

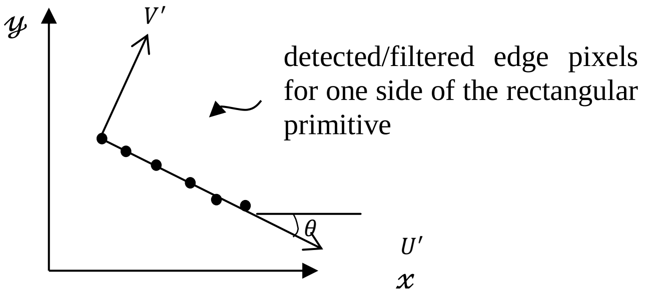
where, m is the number of unknown parameters

$$\hat{\sigma}_o^2 = \frac{E(\tilde{e}^T P' \tilde{e})}{(q - m)} = \frac{(y - A\hat{x})^T P'(y - A\hat{x})}{(q - m)} \qquad \textbf{(5.35)}$$

In summary, the LSA solution to the stochastic model in Equation (5.36) can be derived using Equations (5.24), (5.25), and (5.35). The modified LSA solution is similar to the solution of the traditional Gauss Markov model with the exception that the redundancy is evaluated as the difference between the rank of the modified weight matrix (q) and the number of unknowns (m).

$$y = Ax + D + e \quad e \sim (0, \Sigma') \quad where\ \Sigma' = \sigma_o^2 P'^{+}\ and\ P'D = 0 \qquad \textbf{(5.36)}$$

It is therefore proven that by modifying the weights of the residuals, the additional unknown vector, D, is eliminated from the estimation while still being able to use a similar solution to the traditional Gauss Markov model. Consequently, one can estimate the model parameters while dealing with the non-conjugate points (i.e., the edge pixels and end vertices of the sides of the rectangle) using the modified LSA in the same manner it would deal with conjugate points using traditional LSA. For this purpose, the weight matrix (P) needs to be modified as follows (P'). First, a local coordinate system ($U'V'$) is defined where U' axis is aligned along the extracted edge pixels, and the V' axis is perpendicular to the U' axis as seen in Figure 5.9. The relationship between the image coordinate system (xy) and a local coordinate system ($U'V'$) is described using the angle (θ) between the U' axis and the x axis (Equation (5.37)).



y
V'
detected/filtered edge pixels for one side of the rectangular primitive
θ
U'
x

Figure 5.9: The relationship between the image and the local coordinate systems for the weight modification

$$\begin{bmatrix} U' \\ V' \end{bmatrix} = R_\theta \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} cos\theta & sin\theta \\ -sin\theta & cos\theta \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} \quad \textbf{(5.37)}$$

The original weight matrix, P_{xy}, is defined as the inverse of the variance-covariance matrix $\sum_{xy}$, which depends on the accuracy of the edge extraction as seen in Equation (5.38). The accuracy of the extracted edge pixel is determined by its gradient intensity (G_{xx} and G_{yy}).

$$P_{xy} = \sum_{xy}{}^{-1} = \begin{bmatrix} G_{xx} & 0 \\ 0 & G_{yy} \end{bmatrix} \quad \textbf{(5.38)}$$

The weight of the points in the local coordinate system ($P_{U'V'}$) can be derived using the law of error propagation according to Equation (5.39).

$$P_{U'V'} = R_\theta P_{xy} R_\theta{}^T = \begin{bmatrix} P_{U'} & P_{U'V'} \\ P_{V'U'} & P_{V'} \end{bmatrix} \quad \textbf{(5.39)}$$

Then, the weight matrix can be modified according to Equation (5.40) by assigning a zero value for the weights along the line, to obtain a new weight matrix $P'_{U'V'}$ in the $U'V'$ coordinate system.

$$P'_{U'V'} = \begin{bmatrix} 0 & 0 \\ 0 & P_{V'} \end{bmatrix} \quad \textbf{(5.40)}$$

Finally, the modified weight matrix P'_{xy} in the image coordinate system (xy) can be derived as shown in Equation (5.41).

$$P'_{xy} = R_\theta{}^T P'_{U'V'} R_\theta \qquad \textbf{(5.41)}$$

Using the modified weight matrix in Equation (5.41), one can confirm that the modified weight matrix satisfies the given condition (i.e., $P'_{xy}\ D = 0$) (Equation (5.42)). One should note that after the model-based image fitting, the discrepancy vector in the $U'V'$ coordinate system ($D_{U'}, D_{V'}$) only has U' components and V' components is zero.

$$P'_{xy}D = R_\theta{}^T P'_{U'V'} R_\theta \begin{bmatrix} D_x \\ D_y \end{bmatrix} = R_\theta{}^T P'_{U'V'} \begin{bmatrix} D_{U'} \\ 0 \end{bmatrix} = R_\theta{}^T \begin{bmatrix} 0 & 0 \\ 0 & P_{V'} \end{bmatrix} \begin{bmatrix} D_{U'} \\ 0 \end{bmatrix} = 0 \qquad \textbf{(5.42)}$$

In summary, the model-based image fitting aims at estimating the model parameters by minimizing the difference between the projected model onto the image space and its realization in the imagery. The projected model is defined by the IOPs and EOPs of the utilized cameras, the plane parameters from the LiDAR building detection process, and the current estimates of the model parameters. The model realization in the imagery is defined by the detected and filtered edges. This fitting can be simultaneously conducted for all the images in which the rectangular primitive appears and for all the sides of the rectangle. It should be noted that there is no limitation in terms of the number of images used. The use of a single image would produce satisfactory results; however, the use of more images is expected to provide various views of the building, which decreases the impact of occlusion.

5.6 Sequential model-based image fitting

For complex buildings, different levels of rectangular primitives are derived from LiDAR data using the recursive MBR procedure described in Section 4.3. Since different MBRs have shared vertices between the levels and have the same orientation, the results from the previous level MBRs can be incorporated as apriori information when adjusting the next level MBRs. This section discusses how the modified LSA is sequentially applied to all the rectangular primitives while incorporating this apriori information as fixed constraints during the LSA.

Figure 5.10 illustrates the flowchart of the proposed sequential model-based image fitting. There are three different ways (i.e., two general cases and one gable-roof case) to constrain the prior information depending on the relationships between the different level MBRs. The sequential model-based image fitting starts with determining whether the building in question belongs to the gable-roof case or either of the general building cases. As it will be demonstrated later, a building can be identified from LiDAR data as a gable-roof through adjacency analysis of the neighborhood planes and the intersections of them. Then, the general cases can be distinguished into two cases (i.e., common corner case and corner on side case) by checking two conditions. First, does the second level MBR share any common corner points with the first level MBR? Secondly, when the different level MBRs do not have shared corner points, do any corner points of the second level MBR exist between two corner points from the first level MBR? The following subsections will present diagrams of popular types of buildings (i.e., L-shape, U-shape, gable-roof, T-shape, and multiple level building) to demonstrate the sequential adjustment process of multiple level MBRs.

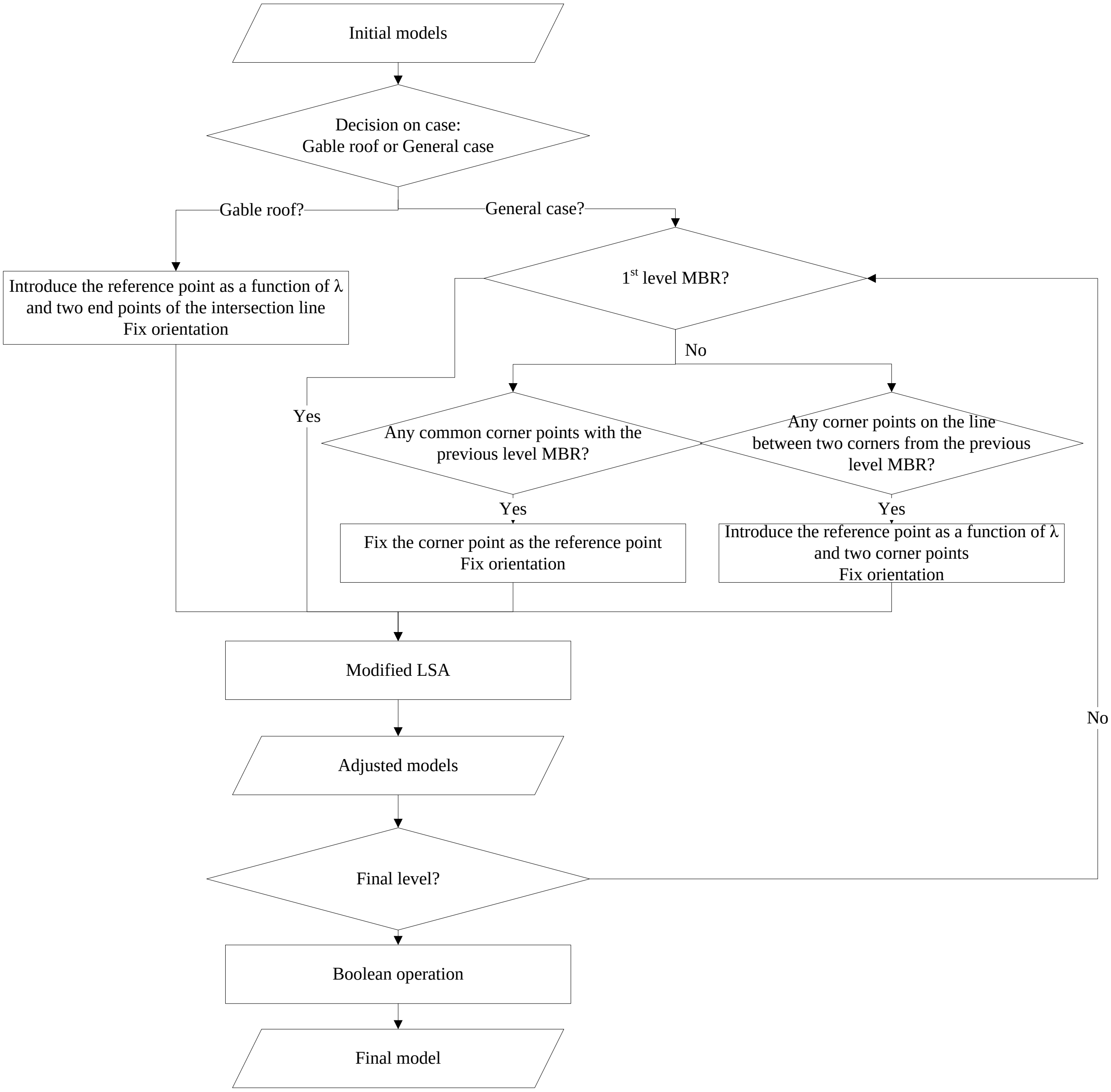
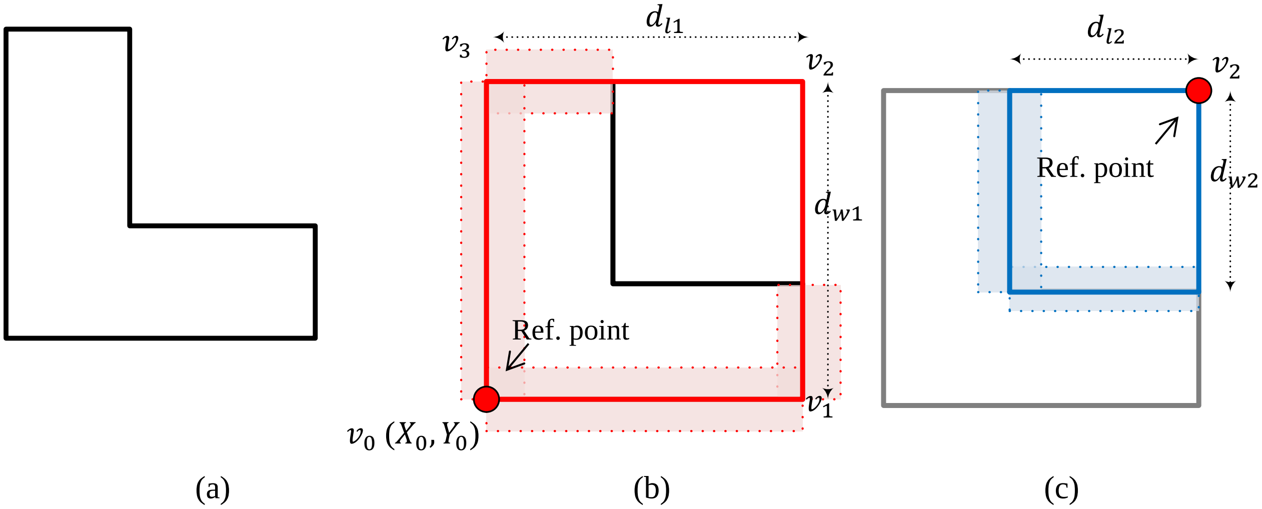

Initial models
Decision on case:
Gable roof or General case
Gable roof?
General case?
Introduce the reference point as a function of λ
and two end points of the intersection line
Fix orientation
1st level MBR?
No
Yes
Any common corner points with the
previous level MBR?
Any corner points on the line
between two corners from the previous
level MBR?
Yes
Yes
Fix the corner point as the reference point
Fix orientation
Introduce the reference point as a function of λ
and two corner points
Fix orientation
Modified LSA
No
Adjusted models
Final level?
Boolean operation
Final model

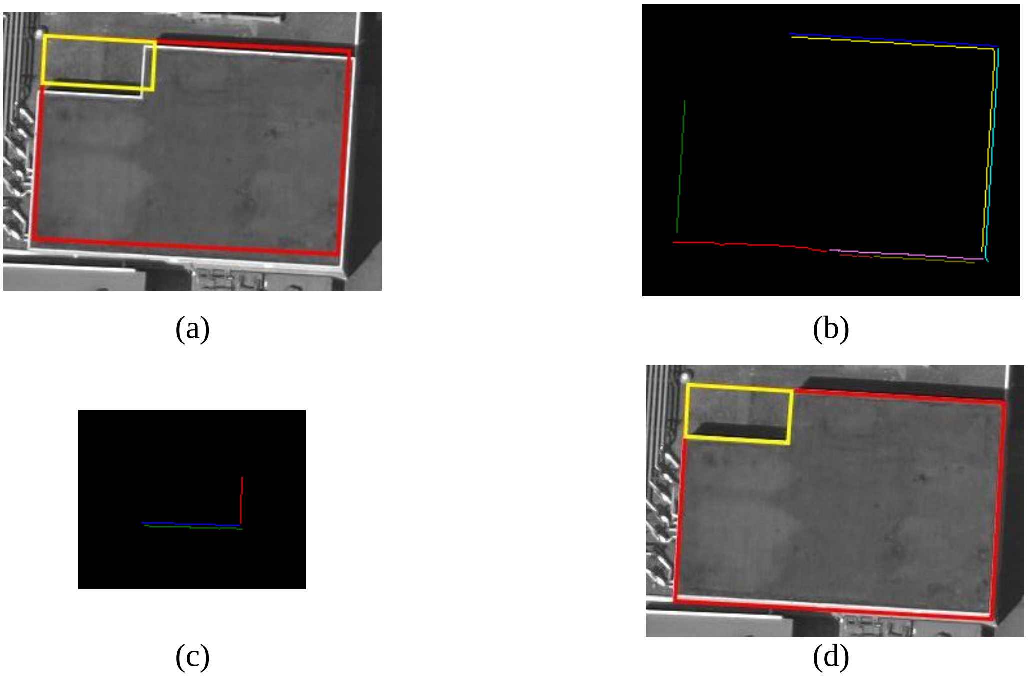
Figure 5.10: Flowchart of the sequential MBR adjustment

An L-shape building is a typical example of the common corner case in the general case. It includes two MBR levels as shown in Figure 5.11(a). The first level MBR is adjusted using the modified LSA proposed in the previous section; and as a result, five model parameters (X_0, Y_0, d_{w1}, d_{l1}, and κ_m) are refined. The second level MBR is adjusted in the same manner as the first level MBR except that the results from the first level MBR are incorporated as aprior information. This is implemented by zeroing the respective rows and columns in the N and C matrix except the diagonal element. In this case, one of the vertices of the second level MBR is shared with the first level MBR (v_2 in Figure 5.11(b)). Since the four corner points of the first level MBR are accurately defined from the first level MBR model-based image fitting, the adjusted coordinates of the shared corner point are used as a reference point of the second level MBR (red circle in Figure 5.11(c)). The reference point coordinate is fixed as a constraint and not solved during the adjustment. The orientation of the second level MBR follows the first level MBR, thus the κ_m angle is fixed during the adjustment. Therefore, the unknown parameters to be estimated reduce to the dimensions of the rectangle, which are d_{l2} and d_{w2}. This can be implemented by replacing the values corresponding to X_0, Y_0, and κ_m in each parameter location with zero in the N matrix and C matrix except the diagonal elements during the LSA. It should be noted that only the edges from the actual building boundaries are considered during the LSA. The implementation is realized by disregarding the edge pixels that do not belong to the parts which do not correspond to the rooftop boundary within the current MBR level. The zones in gray color define the buffer around the physical building boundary as seen in Figure 5.11(b) and Figure 5.11(c). Figure 5.12(b) and (d) show the actual edges used for the adjustment for each level. For the second level MBR, the edge pixels from two sides of the rectangle are used for the adjustment, rather than all four sides.



(a) (b) (c)

Figure 5.11: L-shape building (a), first level MBR (b), and second level MBR (c)



(a) (b)

(c) (d)

Figure 5.12: L-Shape building: Initial MBRs projected onto image (a), edges used for the first level MBR adjustment (b), edges used for the second level MBR adjustment (c), and adjusted MBRs projected onto image (d)

The second instance of the general case (i.e., corner on side case) can be explained best using a U-shape building. This U-shape building is comprised of two MBR levels which do not have any shared corner points between them (Figure 5.13(a)).

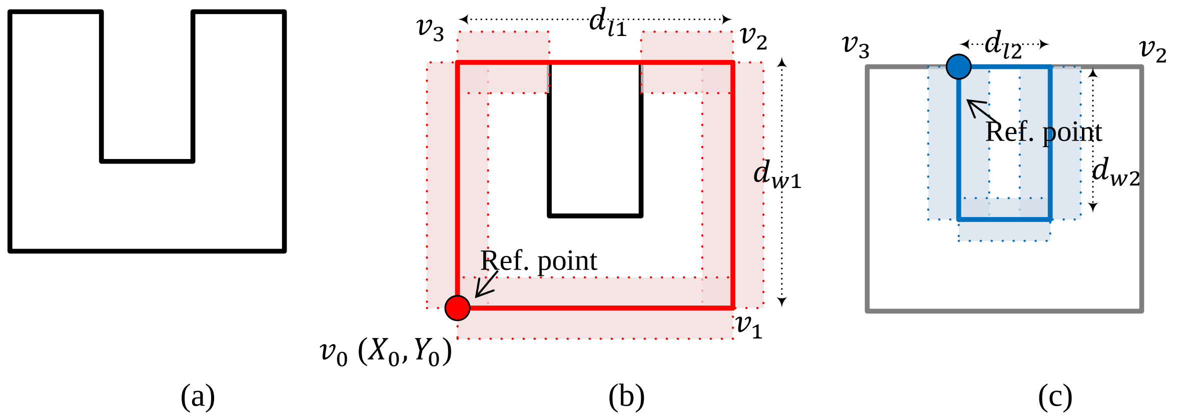


Figure 5.13: U-shape building (a), first level MBR (b), and second level MBR (c)

The first level MBR is adjusted through the LSA (Figure 5.13(b)). In this case, there is no common corner point between the first level MBR and the second level MBR. Instead, one of the corner points of the second level MBR exists along the line between the two corner points of the first level MBR (v_2 and v_3 in Figure 5.13(c)). Hence, the reference point (X_0, Y_0) for the second level MBR can be expressed as a function of the two corner points of the first level MBR (v_2 and v_3) by introducing the scale factor λ according to Equation (5.43). The orientation angle is the same as the first level MBR. Consequently, the number of unknowns for the second level MBR becomes three, which are d_{l2}, d_{w2}, and λ.

$$\begin{bmatrix} X_0 \\ Y_0 \end{bmatrix} = \begin{bmatrix} X_A \\ Y_A \end{bmatrix} + \lambda \begin{bmatrix} X_B & - & X_A \\ Y_B & - & Y_A \end{bmatrix} \tag{5.43}$$

where A and B are the first level MBR corner points

The last case is a gable-roof building. As mentioned, this building case can be automatically identified from the LiDAR data through adjacency analysis and the intersection of neighboring planes. After LiDAR data segmentation in Chapter 3, each plane is checked for any neighboring groups that are adjacent to each other. This is done by comparing the distance from the centroid of each plane to the farthest point of its boundary (d_i, and d_j) with the distance between the centroid of each plane (d_{ij}). As seen in Figure 5.14(a), adjacent planes satisfy the condition in Equation (5.44). Figure 5.14(b) shows that when two planes are not adjacent the condition is not satisfied.

$$(d_i + d_j) \geq d_{ij} \tag{5.44}$$

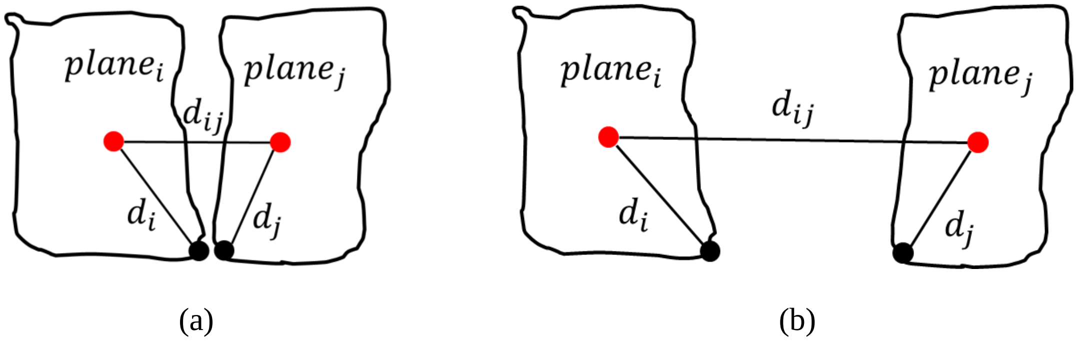


Figure 5.14: Adjacent neighboring planes (a) and non-adjacent neighboring planes (b)

Once adjacent neighboring planes are identified, the intersection line between the neighboring planes is generated. To avoid the intersection of the building primitives that are almost parallel and/or coplanar, the intersection angle between the two planes is checked. If the angle is less than a pre-defined threshold, the intersection line is not considered for further processing. After this analysis, the gable-roof buildings are identified with their intersection line. Gable-roof buildings can be treated as two separate rectangular primitives; however, since LiDAR data produces accurate plane parameters, precise boundary segments between the two sides of a gable-roof can be generated by intersecting the two planes as seen in Figure 5.15(b). Since the intersection of the planes results in an infinite line, two end points are defined by projecting the segmented points onto the intersection line and choosing the end points on the line. Thus, a restriction is imposed that the reference point should exist along the intersection line of the two planes. The reference point is defined using the two end points of the LiDAR intersection line based on Equation (5.43). Using these two end points, the orientation angle of the building can be derived. Hence, a gable-roof model has one unknown parameter λ for the reference point and two shape parameters for the dimensions of each side of the roof.

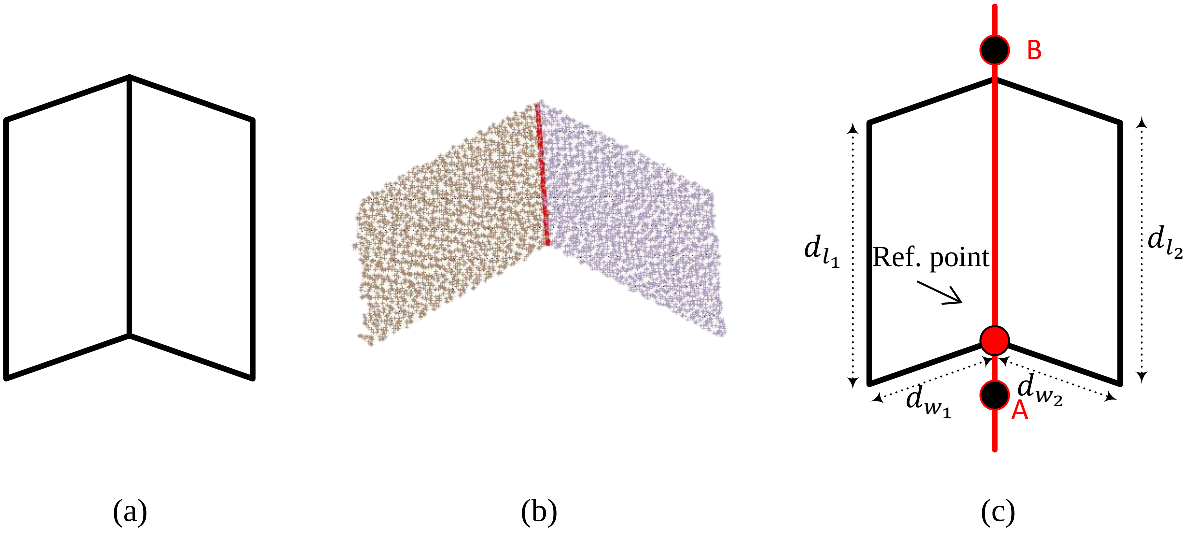


Figure 5.15: Gable-roof building (a), intersection of two LiDAR derived planes (b), and adjustment of gable-roof (c)

Most of the right-angle-corner buildings can be reconstructed based on these three cases and/or a mixture of them. Following are two more examples of buildings having more than one second level MBR and more than two MBR levels.

Figure 5.16(a) shows a T-shape building which has one first level MBR with two second level MBRs. After performing the first level MBR adjustment as seen in Figure 5.16(b), two second level MBRs are adjusted independently. For each second level MBR, shared corner points with the first level MBR are found, which are chosen as the reference points (circles in Figure 5.16(c)). The accurate coordinates of the corner points (v_0 and v_1) and the orientation of the building are known as a result of the first level MBR adjustment. Therefore, the length and width of the second level MBRs are adjusted using the edge information extracted within the buffer, which is the shaded zone shown in Figure 5.16(c). This can be implemented by manipulating the N and C matrix as mentioned in the previous example of corner overlapping case.

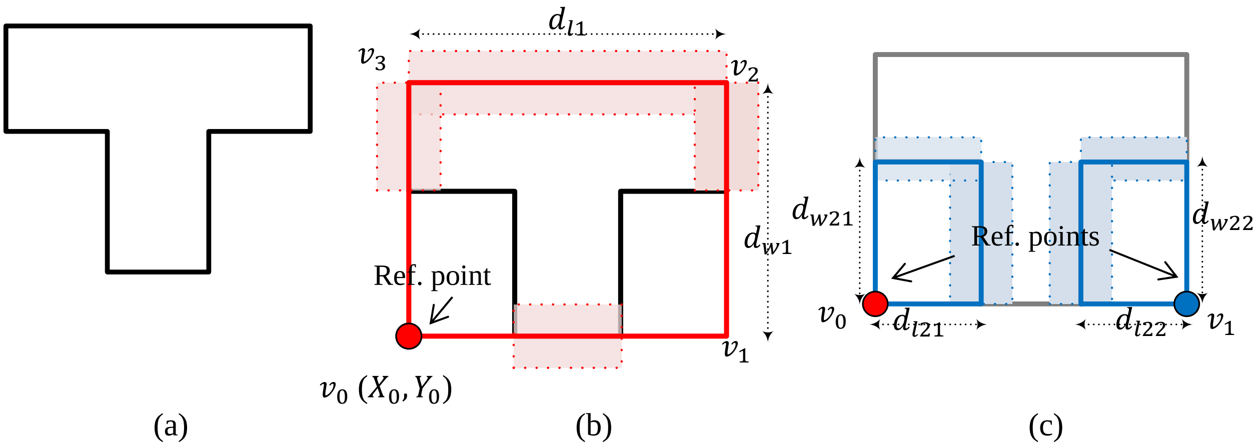


Figure 5.16: T-shape building (a), first level MBR (b), and second level MBR (c)

Another example is a building with three MBR levels, which were presented in Figure 4.11 and shown below in Figure 5.17. All the levels of MBRs are adjusted sequentially in the same manner. After the first level MBR adjustment, the second level MBR is adjusted using the result of the first level MBR. Similar to the L-shape case, the second level MBR has a shared corner point with the first level MBR; therefore, the adjusted corner point from the first level MBR adjustment will be used as a reference point for the second level MBR. Since they also have the same orientation, second level MBR LSA will solve for the length and width. Next, the third level MBR is adjusted by incorporating the result from the second level MBR. As seen in Figure 5.17(c), the third level MBR has a shared corner point with the second level MBR and it has the same orientation as the second level MBR, which is also the same as the first level MBR. The adjusted corner points from the second level MBR adjustment is incorporated as a reference point. Thus, the corresponding rows/columns in the N and C matrix is zeroed except the diagonal elements while only solving for two parameters (d_{l3}, and d_{w3}).

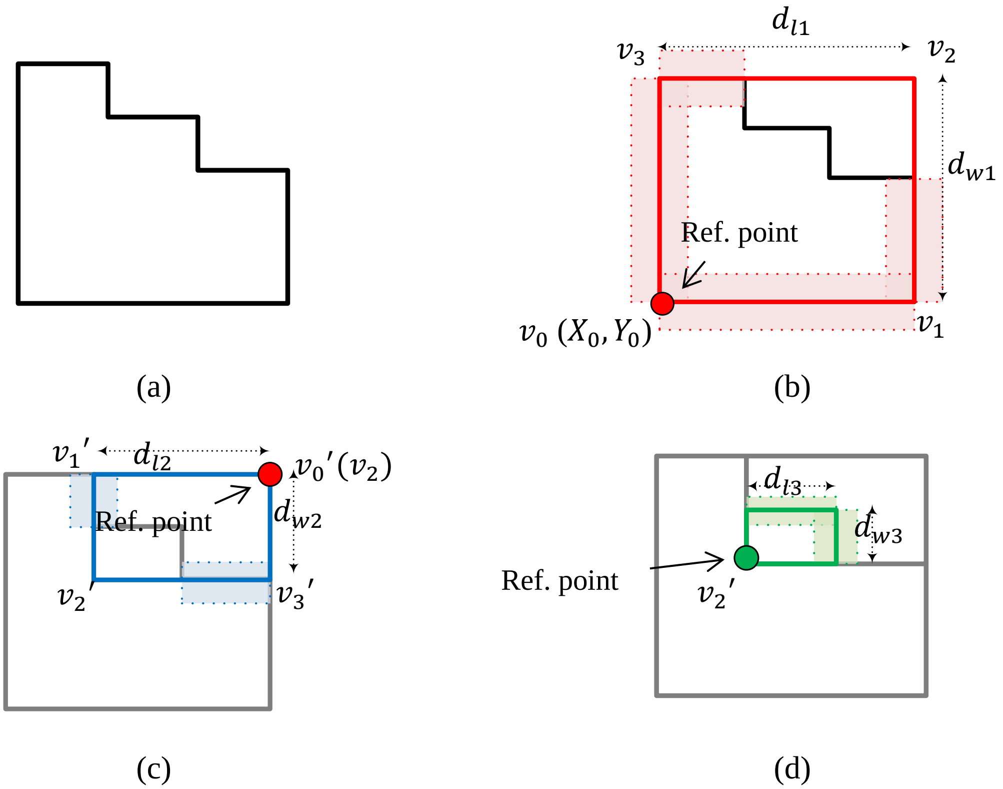
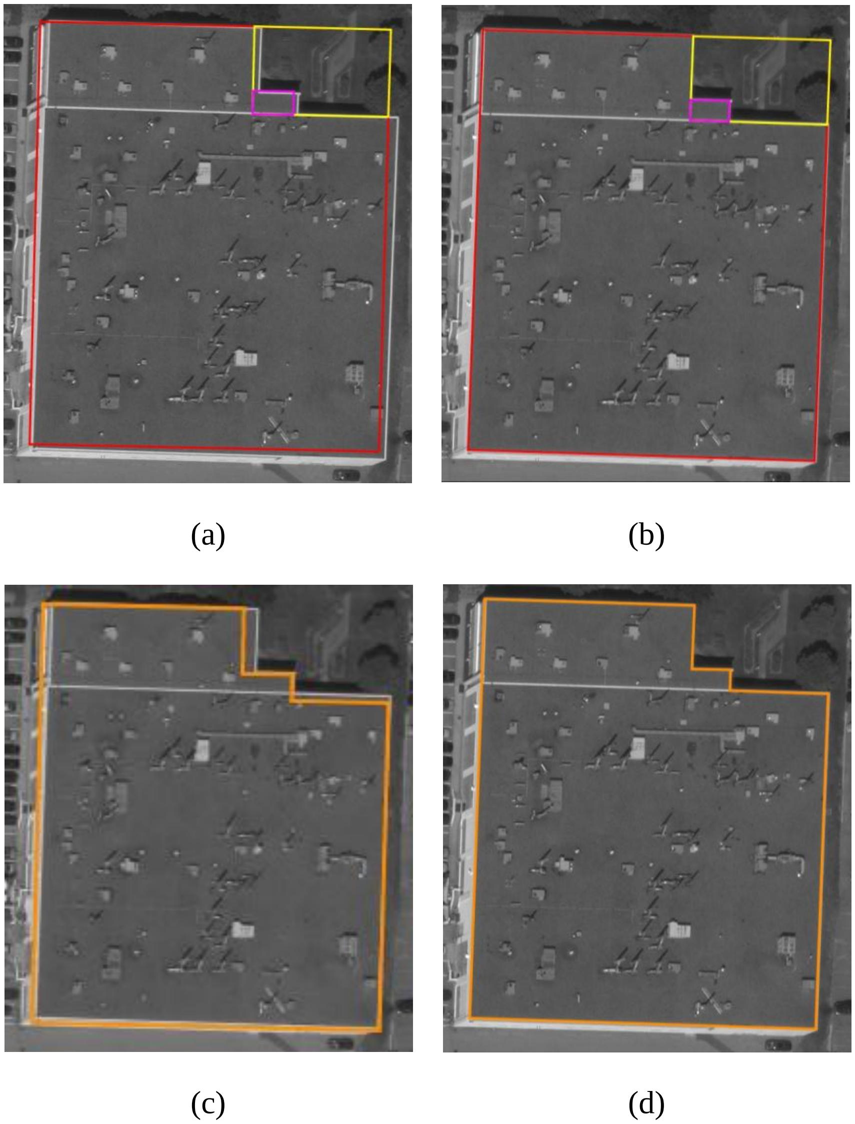


Figure 5.17: Three level building (a), first level MBR (b), second level MBR (c), and third level MBR (d)

Figure 5.18(a) presents three levels of initial models derived from the LiDAR data (Section 4.3) projected onto an image. Different colors represent different MBR levels. The projection of the results after the proposed sequential model-based image fitting is shown in Figure 5.18(b).



(a) (b)

(c) (d)

Figure 5.18: Initial MBRs projected onto an image (a), adjusted MBRs projected onto an image (b), final shape derived from only LiDAR data projected onto an image (c), and final shape adjusted using photogrammetric data projected onto an image (d)

After all the rectangular primitives are adjusted sequentially, the final model is obtained by alternating the subtraction and addition of each level of the adjusted rectangle. Figure 5.18(c) and (d) show the final shape projected onto an image. The final shape derived from LiDAR data as a result of the RMBR process (Figure 5.18(c), which was presented in Figure 4.17, can be compared with the shape after the proposed adjustment based on the image data (Figure 5.18(d)). One can confirm that the accuracy of the building boundary is improved when visually checked.

The determination of different cases is carried out automatically by checking if there are any shared corner points (i.e., common corner case), if any point falls on the side of the rectangle (i.e., corner on the side case), or if two neighboring planes intersect (i.e., gable-roof case). After the sequential model-based image fitting is finished, the quality of adjustment can be checked through the qualitative analysis as presented in Figure 5.18 by projecting the final shape onto an image. As a quantitative analysis, the stochastic properties of the LSA will be used. The overall quality of the adjustment will be evaluated using the a-posteriori variance factor. The stochastic properties of the estimated parameters will be derived through the standard deviation of the adjusted parameters. Also, to check the accuracy of the final product, the derived coordinates of the corner points of each building using the proposed methodology will be compared with the one derived from image-matching reconstruction based on manual image measurement.

5.7 Summary

This chapter employed the model-based approach integrating information from aerial images to improve the horizontal location of the derived model primitives from LiDAR data. To ensure that both the photogrammetric and LiDAR datasets could be combined with their fully synergetic properties, prerequisite procedures of QA, QC, and co-registration of both datasets were performed.

Model-based reconstruction describes the building using a predefined parametric model. In this research, rectangular model primitives were used and the initial model parameters were derived from LiDAR data. Among the initial model parameters, the parameters to be calculated from LiDAR data and the model parameters to be refined through model-based reconstruction were defined. The relationships between the involved coordinate systems were explained. Starting from the approximate values of the model parameters, the parameters to be refined were adjusted by minimizing the difference between the projected model represented by the initial parameters and its realization in the imagery. Realization of the model in the imagery is described by its bounding edges. The edge detection and filtering process was included to ensure that relevant edge pixels that can represent building boundaries were extracted. The discrepancy between the projected model and the edges extracted from images were minimized through a least-squares adjustment fitting while refining the model parameters. For the simplified implementation of the LSA, modified LSA was used. Complex buildings include different levels of rectangular primitives. The modified LSA was sequentially applied to all the rectangular primitives while incorporating information from the previous level MBRs as apriori constraints. The detailed procedure was discussed with examples of typical building types.

Chapter Six: **Experimental results**

6.1 Introduction

The previous chapters discussed the procedures to generate 3D building models of right-angled-corner buildings automatically using both LiDAR and photogrammetric data. In the experimental phase, building hypotheses were first generated from LiDAR data under the assumption that building rooftops are planar and off-ground (Chapter 3). To refine the LiDAR derived building boundaries, a rectangular model-based approach was employed in this research. Therefore, the building boundaries were decomposed into sets of rectangular primitives. The validity of the derived primitives was established using the similarity criteria between the LiDAR building boundaries and the derived primitives (Chapter 4). Finally, the sets of primitives were adjusted sequentially using the available overlapping images through model-based image fitting (Chapter 5). To evaluate the feasibility of the proposed methodology, our experimental results are presented in this chapter using real LiDAR and image data.

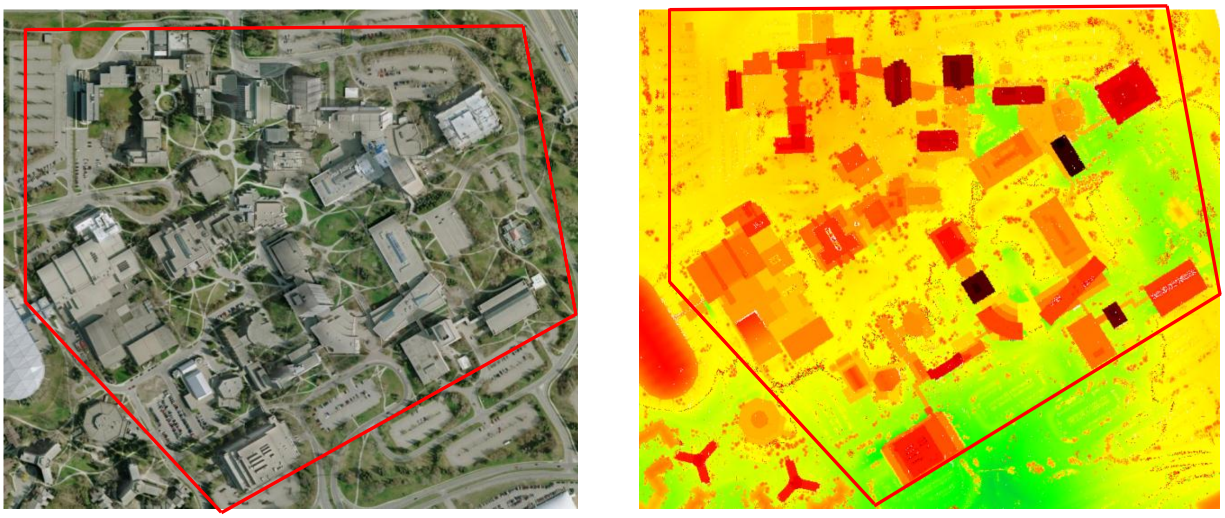
6.2 Dataset description

To verify the proposed methodology, a dataset over the University of Calgary, Canada was used. The dataset includes multiple aerial images and multiple strips of airborne LiDAR data collected over the same area. A brief description of the datasets is provided in Table 6.1.

Table 6.1: Summary of data description

	Photogrammetric data	LiDAR data
System	Analog camera RC30	Optech 3100
Flying height	770m	1,000m and 1,400m
Ground Sampling Distance for image / Average point density for LiDAR	6cm	1.3 points/m^2 for combined data from different flying heights
Planimetric accuracy	6cm	50cm and 70cm for each flying height
Vertical accuracy	14cm	15cm

Nine photos were captured by an analog camera (RC30) at an average flying height of 770m and then were digitally scanned at a resolution of 12µm. The ground sampling distance (GSD) for the scanned images was 6cm. The LiDAR dataset was captured by an Optech 3100 system from flying heights of 1,000m and 1,400m. Ten strips were combined after the LiDAR QA/QC procedures and the average point spacing of the combined strips was approximately 0.88m. The main campus area among the University of Calgary dataset was selected for further examination (i.e., red polygon in Figure 6.1). Figure 6.1 shows (a) one of the aerial images over the area of interest and (b) the LiDAR data displayed in different colors according to their heights.



(a) (b)

Figure 6.1: Aerial photo over the area of interest (a) and LiDAR points colored according to height over the same area (b)

To ensure that the information obtained from both datasets could be combined together, co-registration of both datasets was completed by geo-referencing the images with respect to the LiDAR reference frame. For this purpose, both the derived linear and planar features from the LiDAR data were used as a source of control during the image geo-referencing process as discussed in Chapter 5. As a result, the EOPs were determined with high precision and no significant correlations were observed among the estimated EOPs. The standard deviations of the attitude ranged from 5 to 20 sec and the positions were between 0.024 and 0.097m.

6.3 Building detection experimental results

To detect the possible buildings, the methodology discussed in Chapter 3 was employed on the LiDAR dataset. First, the planar regions were generated through the LiDAR segmentation and QC process. The values and the justification of the used thresholds during this process are described in Table 6.2. The values can be justified according to the average point density, noise level of the dataset, and details of the buildings that need to be detected. Then, the modified convex hull algorithm was applied on the derived segments to identify the boundaries of the regions. The traced boundaries of the segmented results after the QC process are presented in Figure 6.2.

Table 6.2: Values and justifications for the used thresholds for the plane segmentation

Procedure	Threshold	Value	Justification
Neighborhood definition/point density calculation	Number of neighboring points for the plane definition	12 points	Reliable number of points to define planes while considering the possibility for having some outliers
Roughness checking	Height of the cylinder (buffer defined above and below the final fitted plane)	0.8m (0.4m each for above and below)	About four times the vertical accuracy of the LiDAR data (i.e., 0.2m)
	Percentage for point classification as being part of a plane	95%	To be determined as a plane, 95% of the LiDAR points inside the spherical neighborhood should be within the cylinder
Grouping	Radius of the spherical neighborhood for region growing	3·LPD	Radius to define the neighboring points
Iterative plane fitting	Number of neighboring points for attribute computation	20	Reliable number of points to define meaningful plane parameters while considering the possibility of having some outliers

Procedure	Threshold	Value	Justification
Clustering	Minimum angular separation between detectable planes	10°	Acceptable angular deviations among points that would be deemed as part of a single planar region
	Minimum distance between parallel planes	0.8m	Acceptable spatial deviations among points that would be deemed as part of a single planar region (about four times the vertical accuracy of the LiDAR data)
	Size of the minimum detectable cluster	10 points	Smallest region to detect which is related to prior knowledge about a building's size within the area of interest and the average point density
Intersection checking between adjacent clusters	Minimum angle between two clusters before intersection	20°	Avoids the intersection of planes that are almost parallel
	Minimum distance between two clusters to be determined as adjacent clusters	1.5m	Less than twice the average point spacing of the LiDAR data

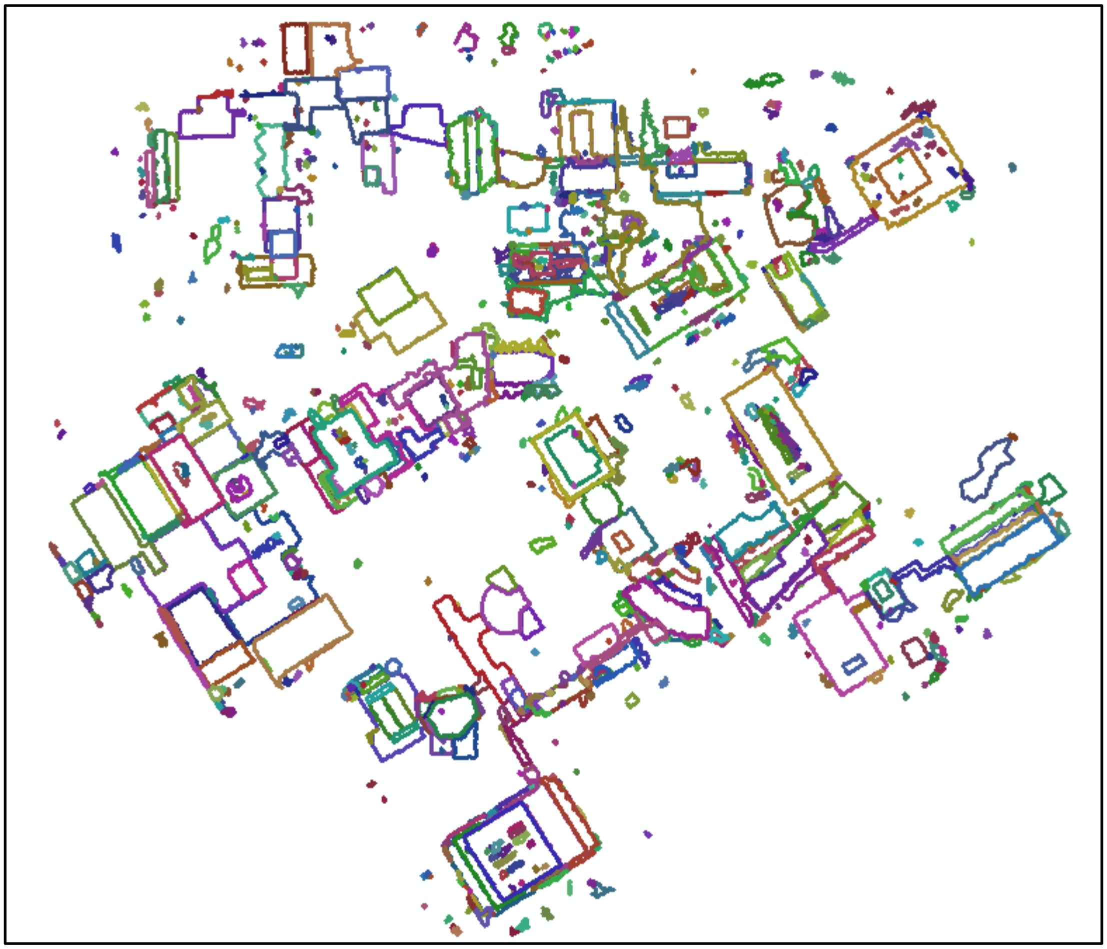


Figure 6.2: Region boundaries after the planar segmentation and QC process

To generate building hypotheses, in addition to detecting planar regions, non-ground regions must be identified. The segmentation-based LiDAR classification discussed in Chapter 3 was performed to distinguish terrain and off-terrain points. Figure 6.3 displays the non-ground regions of the study area in blue.

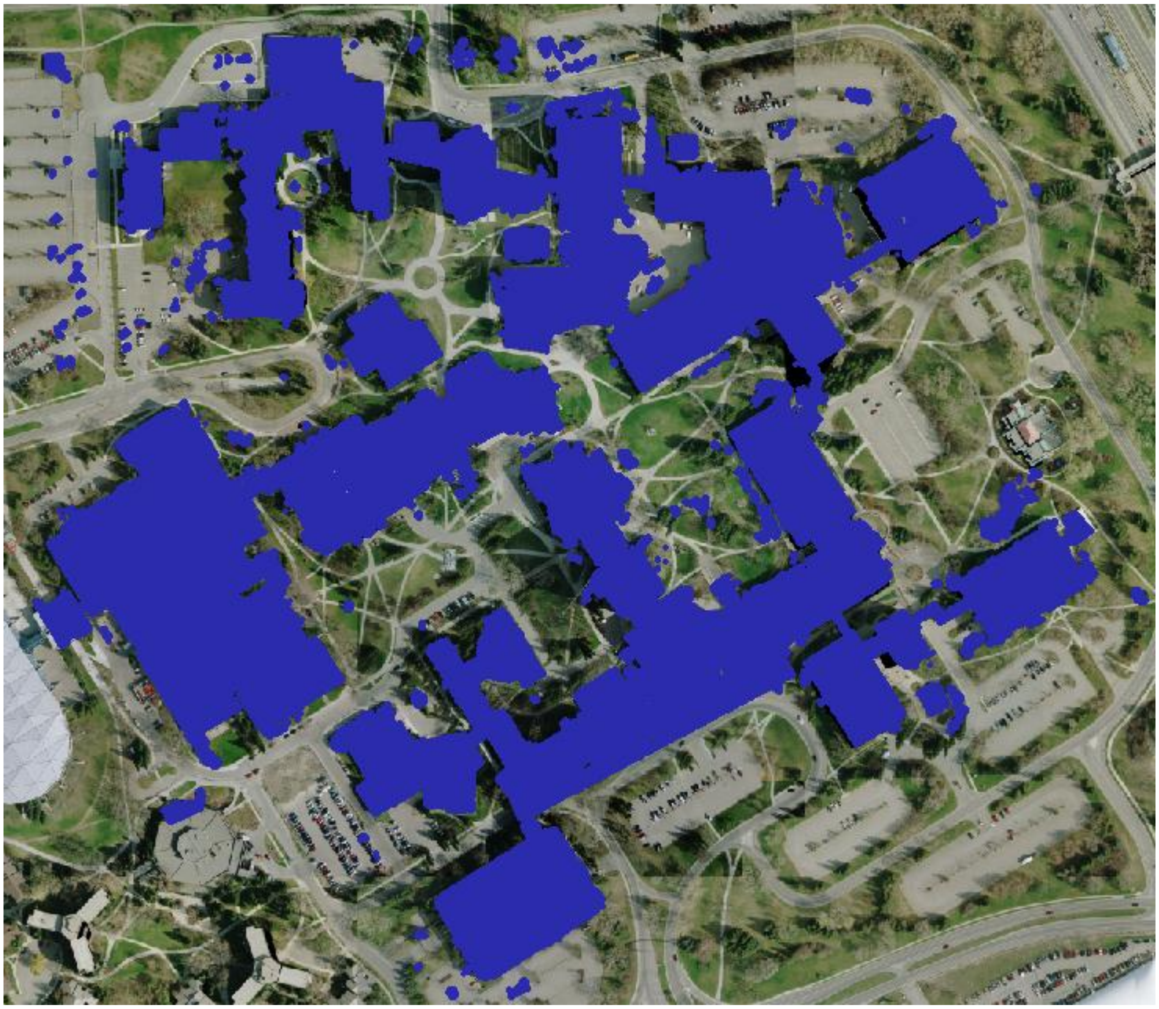


Figure 6.3: Classified off-terrain points

Among the points identified as planar and non-ground through the aforementioned procedures, regions with sizes and heights larger than our predefined thresholds (i.e., 200m^2 and 20m) were identified as building hypotheses. The generated building hypotheses and their boundaries are shown in Figure 6.4 and Figure 6.5, respectively. As seen in Figure 6.5, LiDAR-derived boundaries represent the location of the building roughly and its shape. However, the irregular boundaries need to be regularized and refined to represent actual building boundaries, which are achieved by incorporating image information.

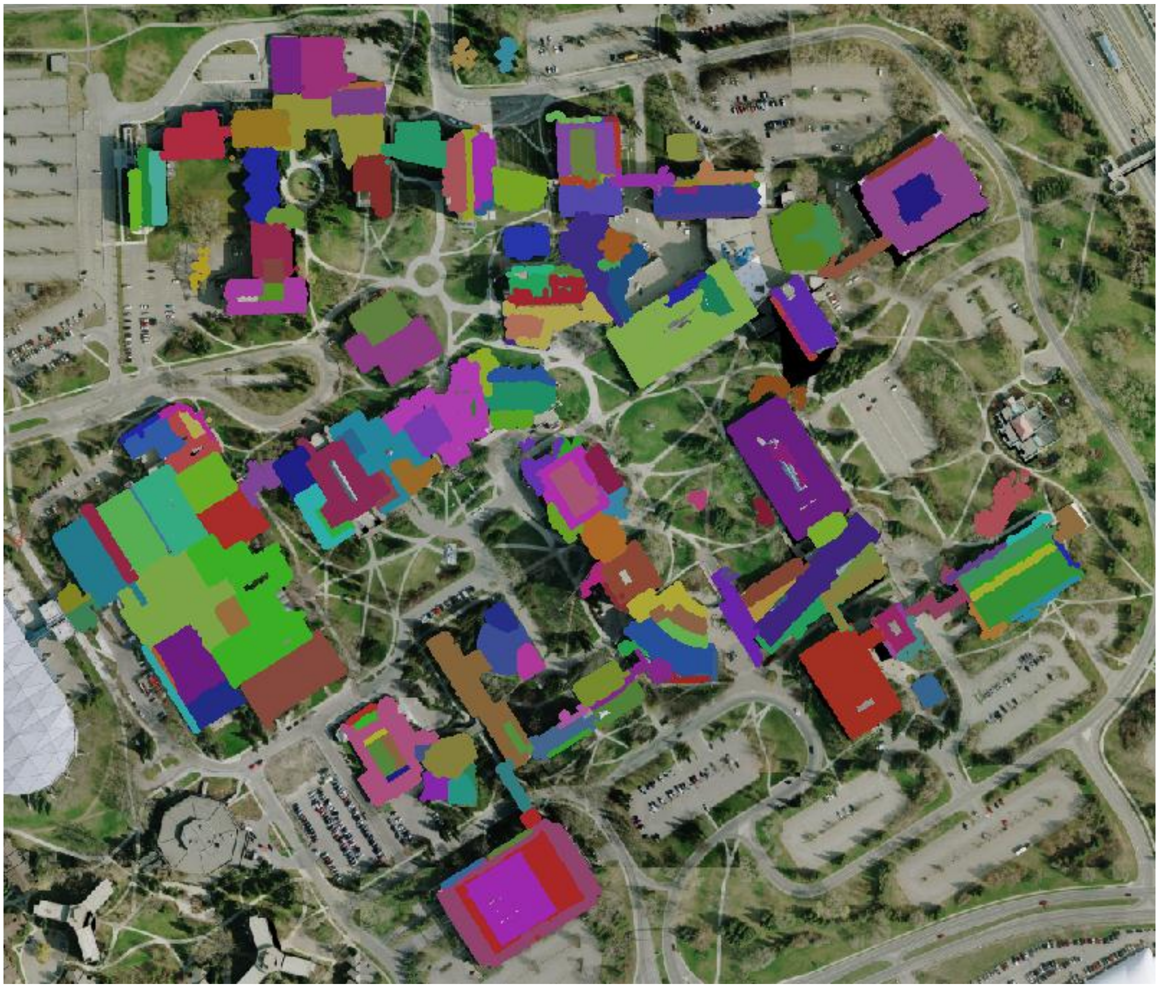


Figure 6.4: Generated building hypotheses from LiDAR data overlaid on top of an orthophoto

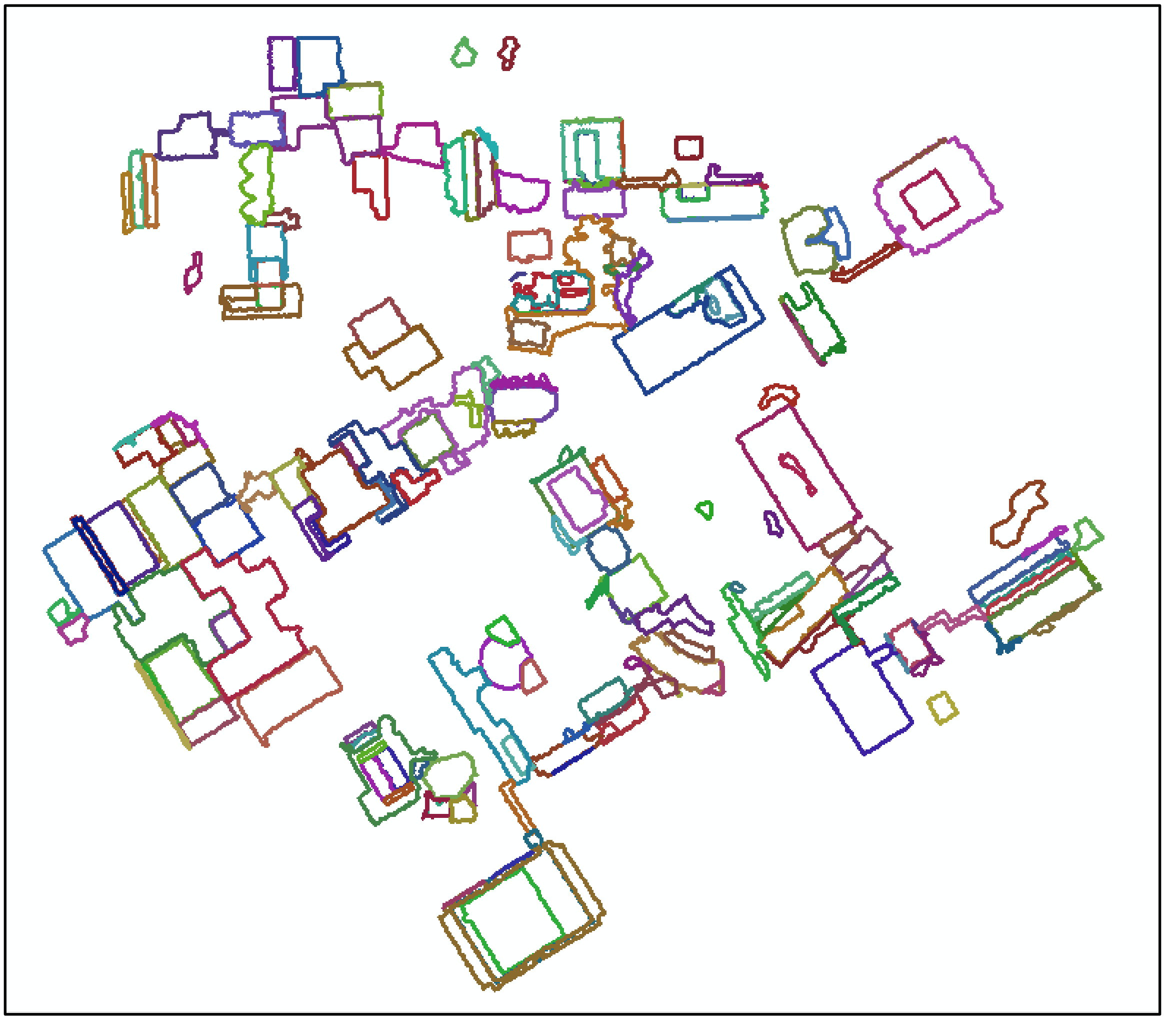


Figure 6.5: Boundaries of the building hypotheses

To evaluate the performance of the building detection (i.e., how correctly the buildings were identified), the identified building segments and missing building segments were counted. The available campus 2D map shown in Figure 6.6 was used as reference data to verify the existence of the buildings. Site visits were also available to confirm the presence of a given building.

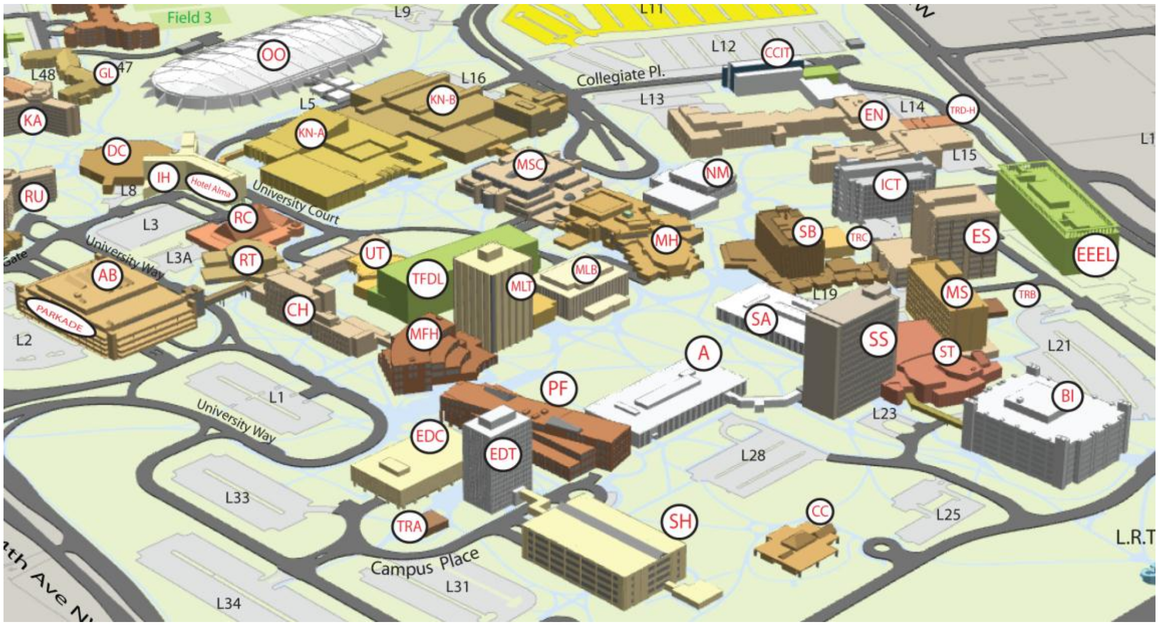


Figure 6.6: University of Calgary campus map used as reference data

The study area included a total of 32 buildings. While some buildings were characterized by a single plane, some of the buildings were represented by up to 12 planes. There were a total of 112 planes that constituted the buildings in the study area. Therefore, the quality of building detection was assessed by how many planes were segmented correctly. Table 6.3 presents the matrix utilized to evaluate the outcome of the building detection process. The proposed procedure produced 140 planes as building hypotheses. Among the produced building hypotheses, 112 planes were correctly detected (TP). Twenty-eight planes were identified as buildings that were, in reality, trees and other objects near buildings. However, the procedure identified all the buildings as buildings (i.e., zero false negative).

Table 6.3: The quality analysis of the building detection process

	Campus map (reference data)	
Results from the proposed methodology	Building	Non-building
Identified as building	112 (TP)	28 (FP)
Missing building segments	0 (FN)	N/A (TN)

Based on the above matrix, *completeness* was calculated as 100% according to Equation (3.2), which means that, using LiDAR data, one could identify all of the buildings as buildings. However, 20% of the non-building objects (e.g., trees) were included as building hypotheses, which demonstrated 80% of *correctness* measure. The 20% of the segments which were erroneously identified as buildings could be filtered out during the RMBR process since the boundaries of the trees were not regular in shape as seen in Figure 6.7.

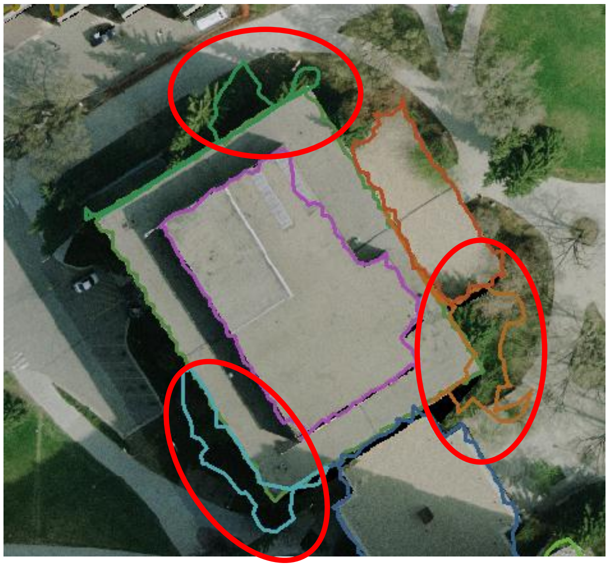
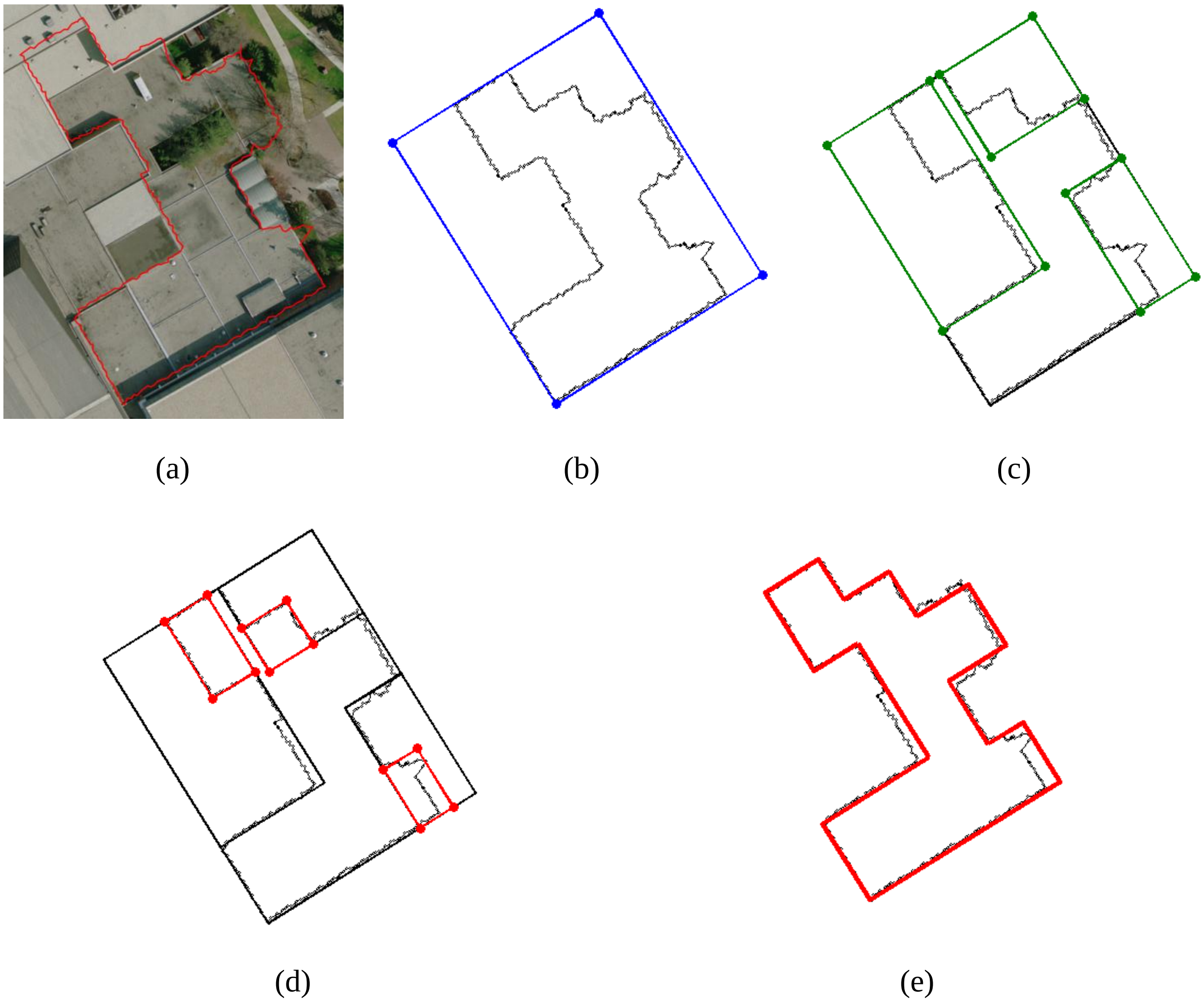


Figure 6.7: Example of erroneously identified building regions (FP)

6.4 Boundary decomposition experimental results

As evaluated in the previous section, 140 planes were derived as building hypotheses. The RMBR algorithm proposed in Chapter 4 was implemented in Microsoft Visual C# with Matlab display functions. The RMBR algorithm was applied on the generated boundaries of the building hypotheses. An example of applying the RMBR process on a building boundary is presented in Figure 6.8. Figure 6.8(a) shows the LiDAR derived boundary projected onto an image. The presented example includes one first level MBR (b), three second level MBRs (c), and three third level MBRs (d). Figure 6.8(e) shows the final shape of the building after the alternating series operation of each level based on Equation (4.1). The distance threshold used for the RMBR process is around two times the average point spacing of the LiDAR data, which is 2m.



(a) (b) (c)

(d) (e)

Figure 6.8: The RMBR procedure: projected boundary onto an image (a), first level MBR (b), second level MBRs (c), third level MBRs (d), and the final shape (e)

Figure 6.9 shows an example of the projection of the final regularized shapes from the decomposed primitives of the building hypotheses onto an image.

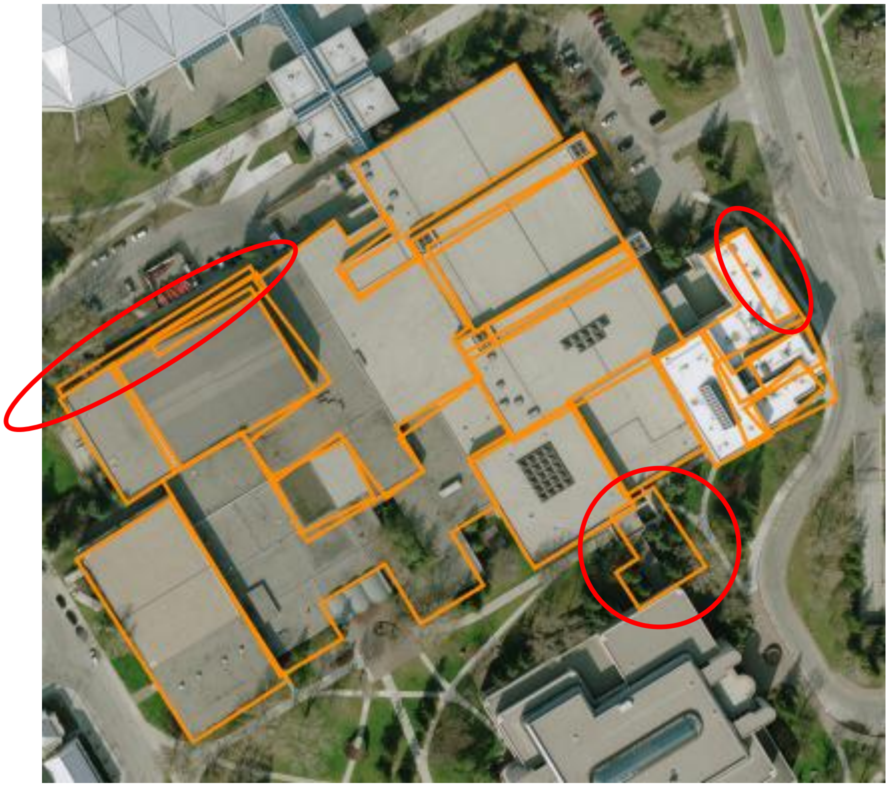


Figure 6.9: An example of the final shapes derived as a result of the RMBR process projected onto an image

As mentioned before, some of the non-building objects were falsely identified as buildings (red circles in Figure 6.9). To choose the buildings that were comprised of rectangles only (i.e., right-angled-corner buildings), the two criteria that compare the LiDAR boundary points and the derived MBRs were used. The two criteria included the area ratio and the RMSE of the normal distance between the LiDAR boundary points and the derived MBRs as discussed in Chapter 4. In this research, the MBRs with an area ratio larger than 0.6 and a RMSE value less than the used distance threshold (i.e., 2m) were selected as the model primitives for further processing. Those selected were believed to belong to right-angled-corner buildings. Figure 6.10 shows the final shapes of the selected MBRs after the evaluation, and it can be seen that the irregular boundaries from falsely identified buildings were removed through this evaluation process.

To determine how correctly the right-angled-corner buildings were selected and that the falsely identified buildings were filtered out during this process, the completeness and correctness of the evaluation process were calculated. The number of right-angled-corner building planes in the test area was 92 out of 112 building planes. Out of the 92 planes, 92 were determined to be right-angled-corner buildings (TP). There was no right-angled-corner planes identified as non-right angled-corner building planes (zero false negative). Out of the 28 falsely identified planes from the building hypotheses, 21 planes were correctly identified as non-right angled corner building (TN). Seven planes were selected as right-angled-corner buildings (FP). Therefore, the completeness measure after this process was 100% and the correctness measure was 93% since all the right-angled-corner building planes were correctly selected for further processing and only 7% of the planes were falsely selected. This number confirms that the majority of the erroneously identified building planes were filtered out. The selected MBRs were stored in Extensible Markup Language (XML) format to deal with the different levels of MBRs efficiently for the sequential model-based image fitting process.

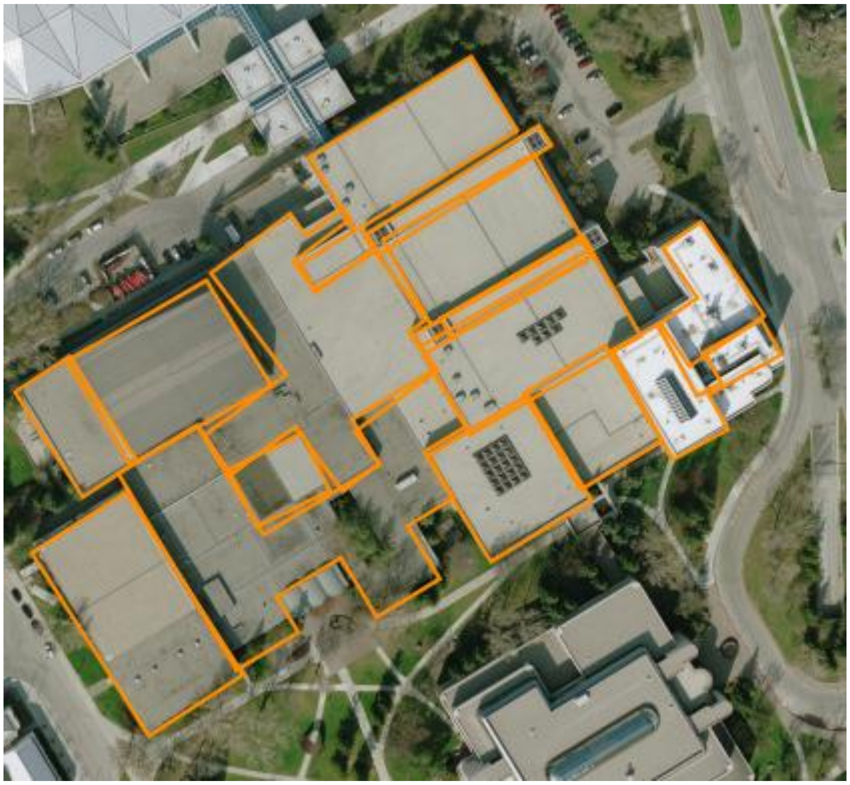


Figure 6.10: Selected MBRs projected onto an image after the right-angled-corner buildings' check

6.5 Model-based reconstruction experimental results

From the LiDAR-derived boundary, the approximate rectangular model parameters were established using the proposed RMBR algorithm. The rectangular model parameters were adjusted during the model-image fitting process, which can be applied on a single image; however, using more images captured from different views can reduce the chance of occlusions. Figure 6.11 shows an example of a simple rectangular building adjusted using three images within which the building appears. The quality of LiDAR-derived MBR depends on the planimetric LiDAR accuracy and average point spacing. Since the distance threshold to derive the initial model primitives during the RMBR process was 2m and the expected planimetric accuracy of the used LiDAR data was 0.6m, the initial MBR was assumed to be accurate within those ranges. Therefore, the buffer for the edge detection/filtering around the initial projected model translated into 33 pixels, which was equivalent to around 2m (the GSD of the image is 6cm). As the number of iterations increased, the buffer size for edge detection/filtering decreased until it reached five pixels, which is equivalent to 30cm on the ground. Figure 6.12 shows the adjusted MBR projected onto the three images after the model-based image fitting. The model-based image fitting software with its visualization was developed in Microsoft Visual C#.

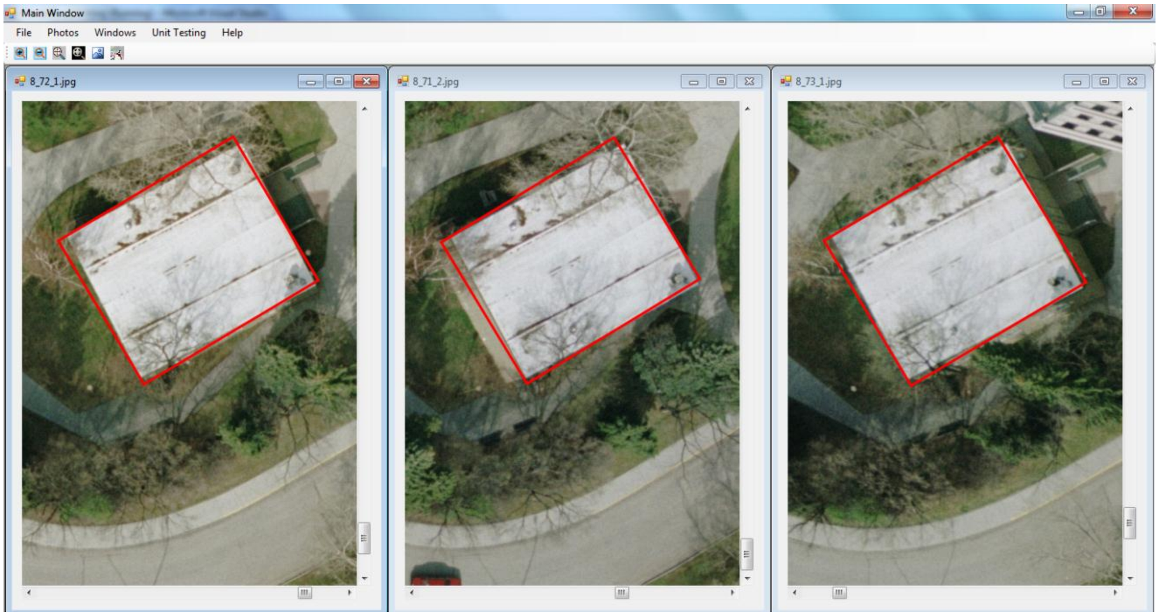


Figure 6.11: Projection of LiDAR-derived MBR onto three images

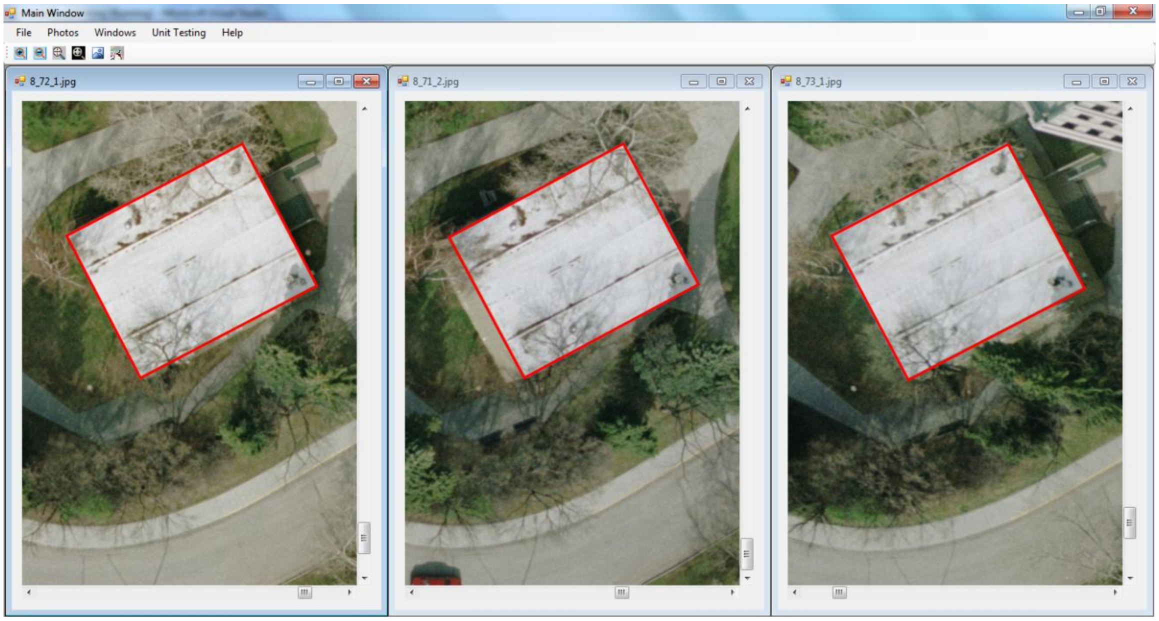


Figure 6.12: Projection of the adjusted MBR onto three images

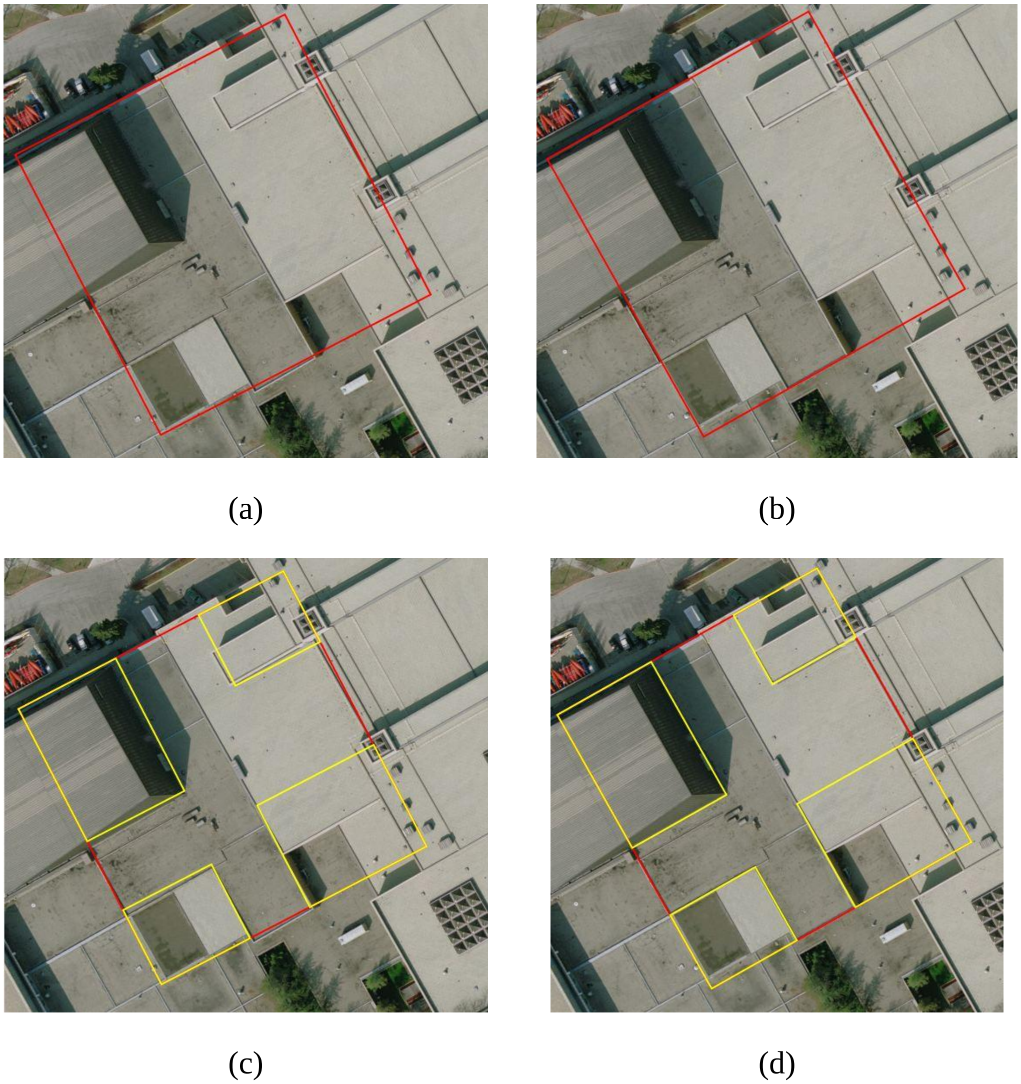
To demonstrate the quality of the model-based image fitting process, the stochastic properties of the LSA are presented in Table 6.4. The overall quality of the adjustment was confirmed by the a-posterior variance factor. The standard deviations of the adjusted parameters show that the precisions of the adjusted parameters were quite good. It is worth mentioning that the estimated values look too optimistic because of huge redundancy.

Table 6.4: The stochastic properties of the model-based image fitting

	$\widehat{\sigma_0}$	σ_{X_0} [m]	σ_{Y_0} [m]	σ_{d_w} [m]	σ_{d_l} [m]	σ_{κ_m} [sec]
values	0.0004	±0.004	±0.003	±0.004	±0.004	±35.108

Figure 6.13 presents an example of a more complex building which can be decomposed into multiple levels of MBRs to show how the rectangular primitives were sequentially adjusted. As mentioned previously, only the edges extracted from the building boundaries for each level were used for the model-image fitting. Figure 6.13(a), (c), (e), and (g) show the initial MBRs derived

from the LiDAR data projected onto the image with different level of detail. Figure 6.13(b), (d), (f), and (h) show the projection of the adjusted MBRs. In this case, two images were used during the adjustment. Depending on the desired level of detail, the MBRs shown vary from a single level MBR, which is the lowest level of detail corresponding to LoD1, to four MBR levels with the highest level of detail (i.e., LoD2). Each MBR level is shown in a different color (red, yellow, magenta, and blue). Visual inspection of the MBRs' projection onto an image confirms the proper alignment of the MBRs with the building's edges.

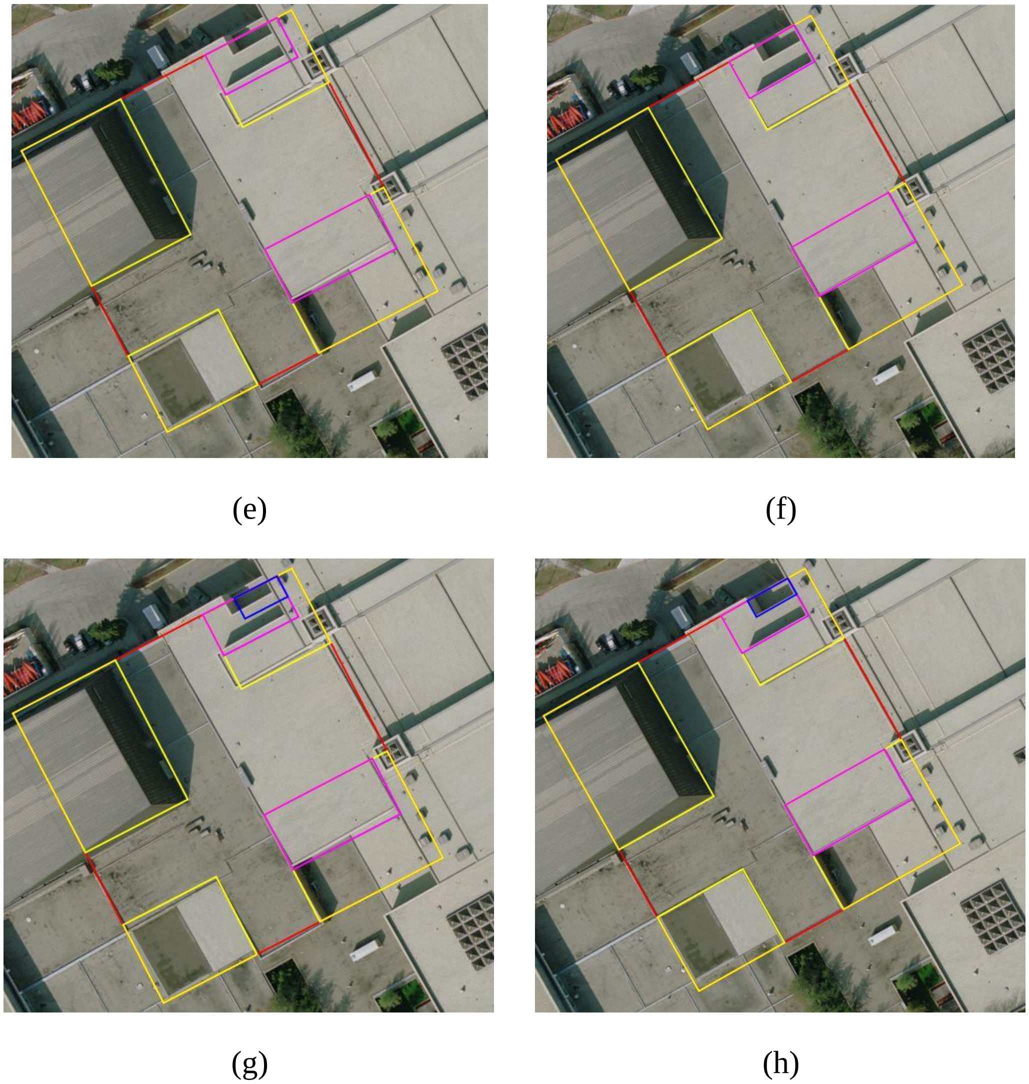


(a)

(b)

(c)

(d)

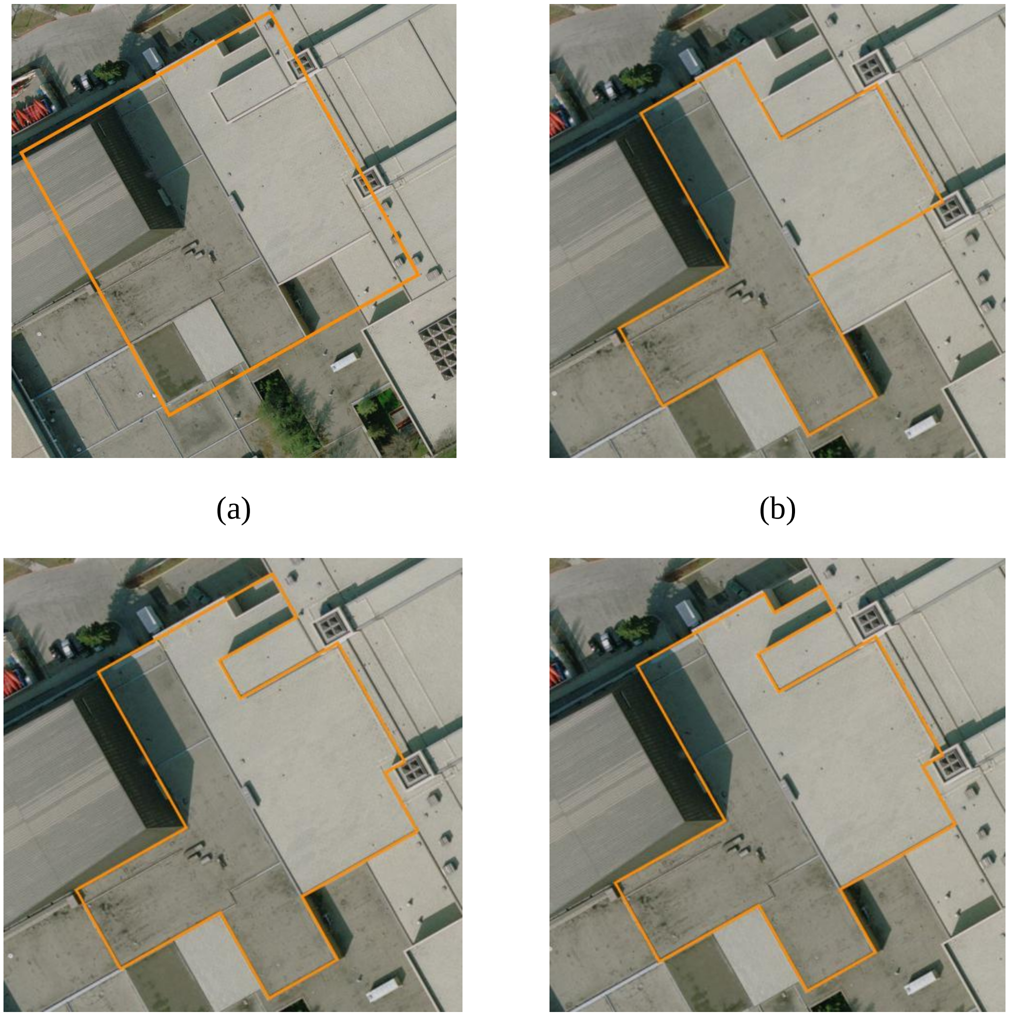


(e) (f) (g) (h)

Figure 6.13: MBRs projected onto the image: initial (a) and adjusted (b) first level MBR in red color, initial (c) and adjusted (d) second level MBRs in yellow, initial (e) and adjusted (f) third level MBRs in magenta, and initial (g) and adjusted (h) fourth level MBR in blue

The final constructed boundaries from the MBRs using an alternating series operation are shown in Figure 6.14. A single level MBR (Figure 6.14(a)) was constructed from the first level MBR shown in Figure 6.13(b), and two levels of MBR (Figure 6.14(b)) were derived by subtracting the second level MBR from the first level shown in Figure 6.13(d). Three levels of MBR (Figure

6.14(c)) and four levels of MBR (Figure 5(d)) were constructed in the same manner as the previous level of MBRs. The level of detail was determined by considering the resolution of the image and the LiDAR point density.


(a) (b)

(c) (d)

Figure 6.14: Final display of MBRs from: single level (red) (a), two levels (red - yellow) (b), three levels (red – yellow + magenta) (c), and four levels (red – yellow + magenta - blue) (d)

Figure 6.15 shows an example of a gable-roof building hypothesis generated from LiDAR data. The gable-roof case was modeled by adding the constraint that the reference point should be along the intersection line between the two planes. Figure 6.16 shows the projection of the initial derived models through a neighborhood analysis and the plane intersection onto four images; and the initial reference point (i.e., blue circle) can be seen to exist along the intersection of the two planes. The adjusted models after the model-based image fitting procedure are projected onto the images (Figure 6.17), which shows that the position of the reference point is moved only along the intersection line and the dimension of the models were accurately refined.

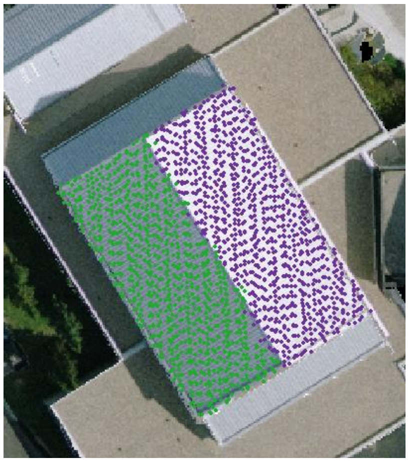


Figure 6.15: Segmented planes of a gable-roof building

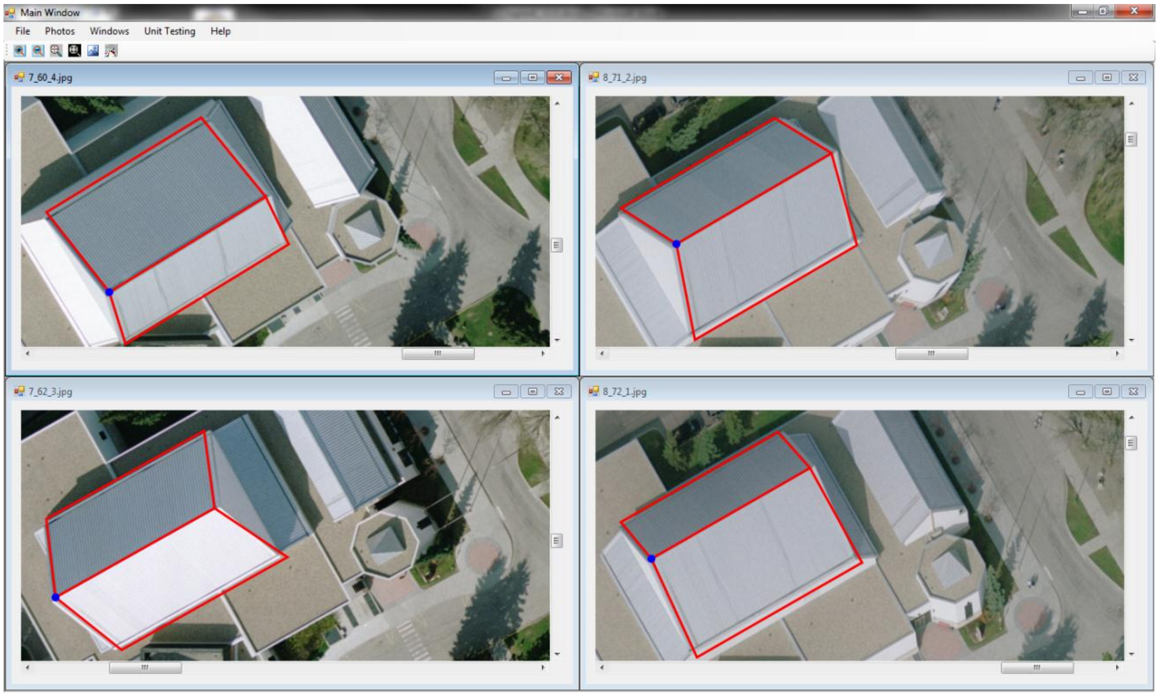


Figure 6.16: Projection of initial models derived from LiDAR data with the reference point (blue circle) along the intersection of two planes onto four images

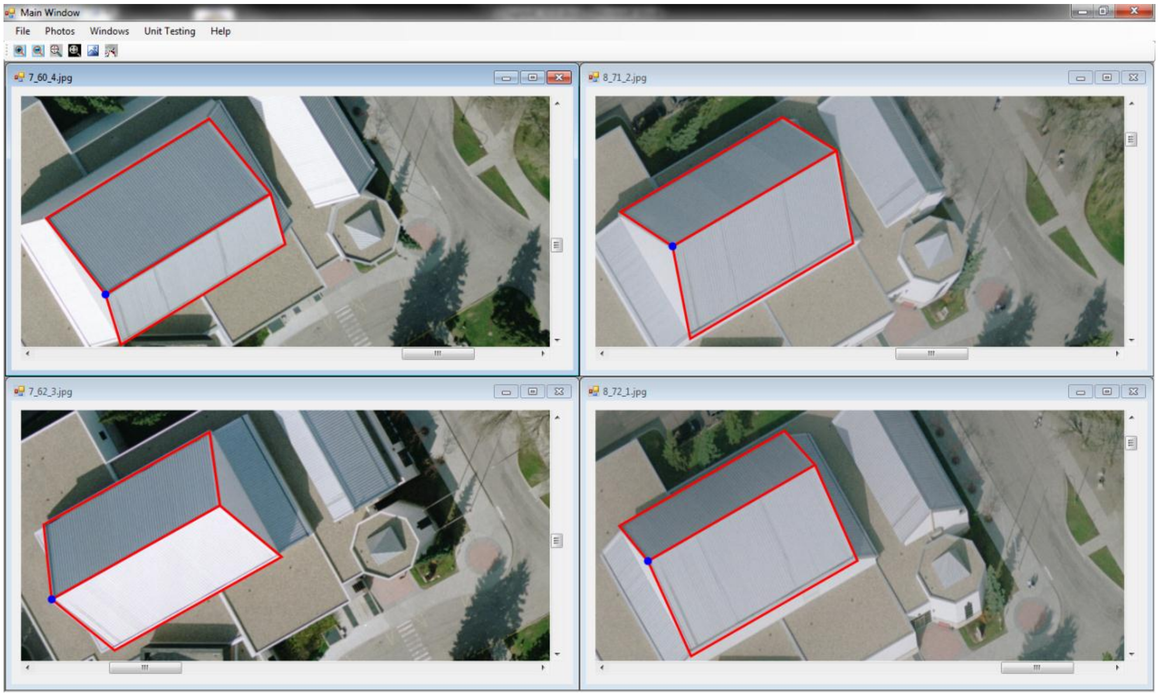
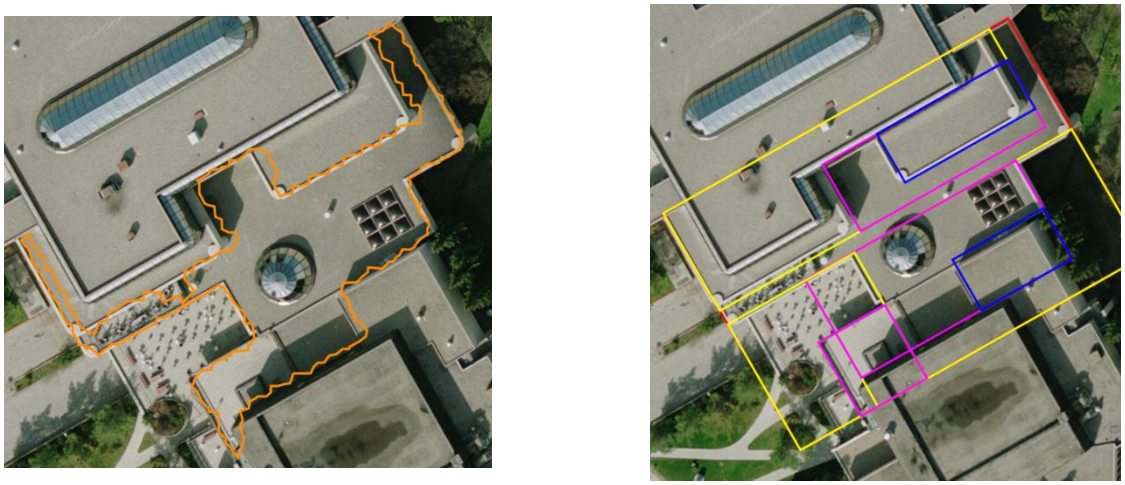


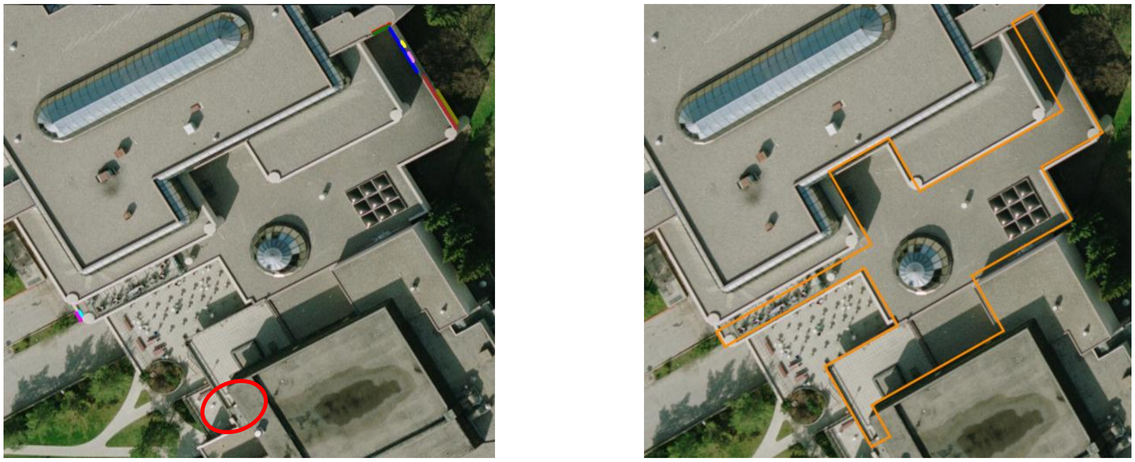
Figure 6.17: Projection of adjusted models with the reference point (blue circle) along the intersection of two planes onto four images

Even when the building is a right-angled-corner building and the RMBR procedure decomposes them correctly, the edges that represent the building might not be extracted properly depending on the surrounding structures and image condition. Figure 6.18 shows an example where the building boundaries (Figure 6.18(a)) were decomposed into multiple rectangles through the RMBR process (Figure 6.18(b)), but it was not possible to extract a sufficient number of edges to adjust the model parameters. As seen in Figure 6.18(c), one side of the building was too short (red circle) and the edges that ordinarily could represent the building boundary were missing. Hence, the first level MBR could not be properly adjusted during the model-based image fitting process. It produced an error message with the lack of information to determine the length of the rectangle. In this case, an alternative representation of the buildings could be sought. One alternative could be to use the final shape derived from the LiDAR data only without adjustment through the model-based image fitting process as seen in Figure 6.18(d).



(a)

(b)



(c) (d)

Figure 6.18: LiDAR building boundary (a), different levels of MBRs (b), extracted/filtered edges for the first level MBR (c), and final shape derived from LiDAR data only (d)

After the LiDAR-derived primitives are adjusted through the sequential model-based image fitting, the final shape of the building can be derived using the refined rectangular primitives. Figure 6.19 presents the final derived building models of the study area displayed on top of the orthophoto. Figure 6.20 displays the final building models in 3D. It is important to mention that the presented results were reconstructed in a fully automatic way.

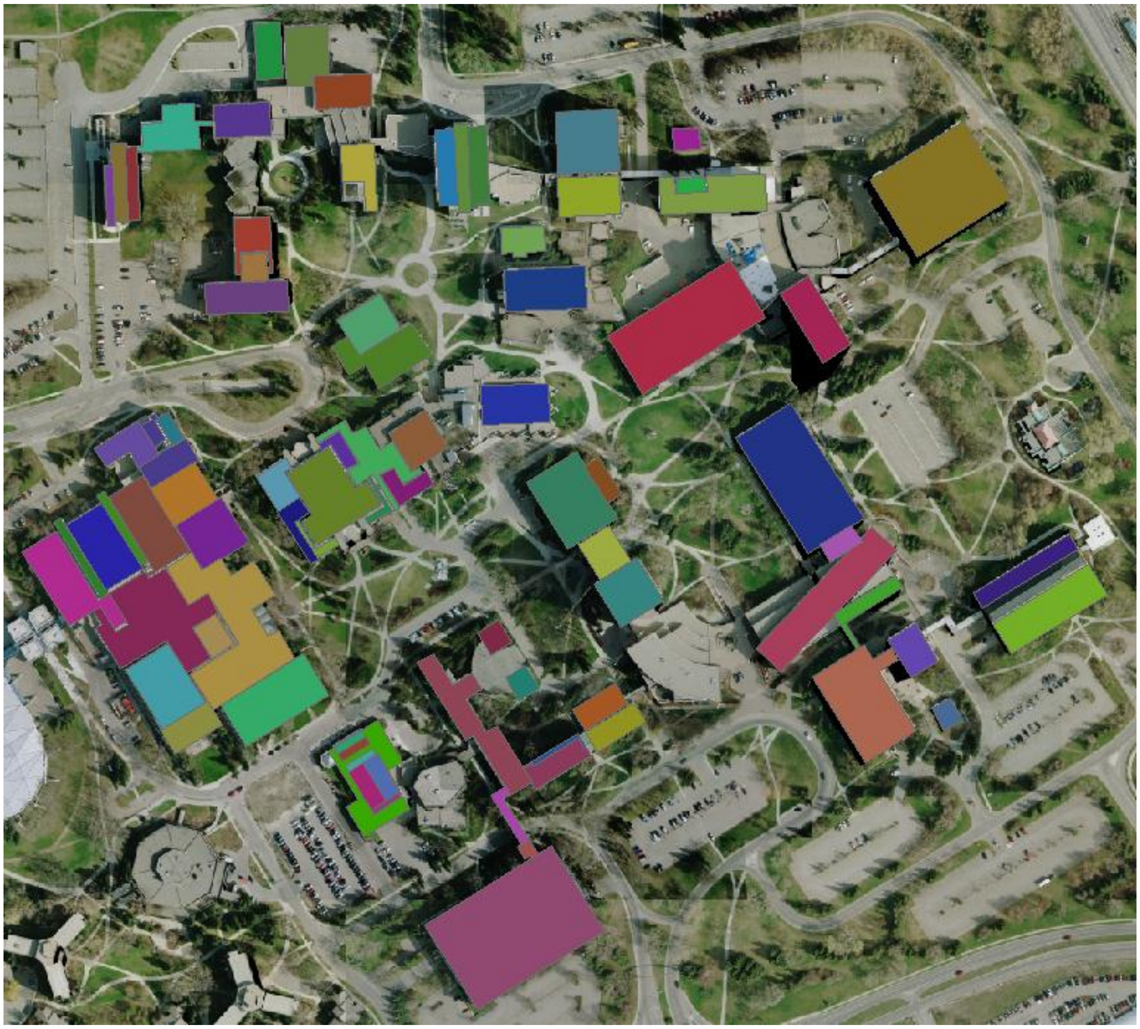


Figure 6.19: Final derived building models on top of orthophoto

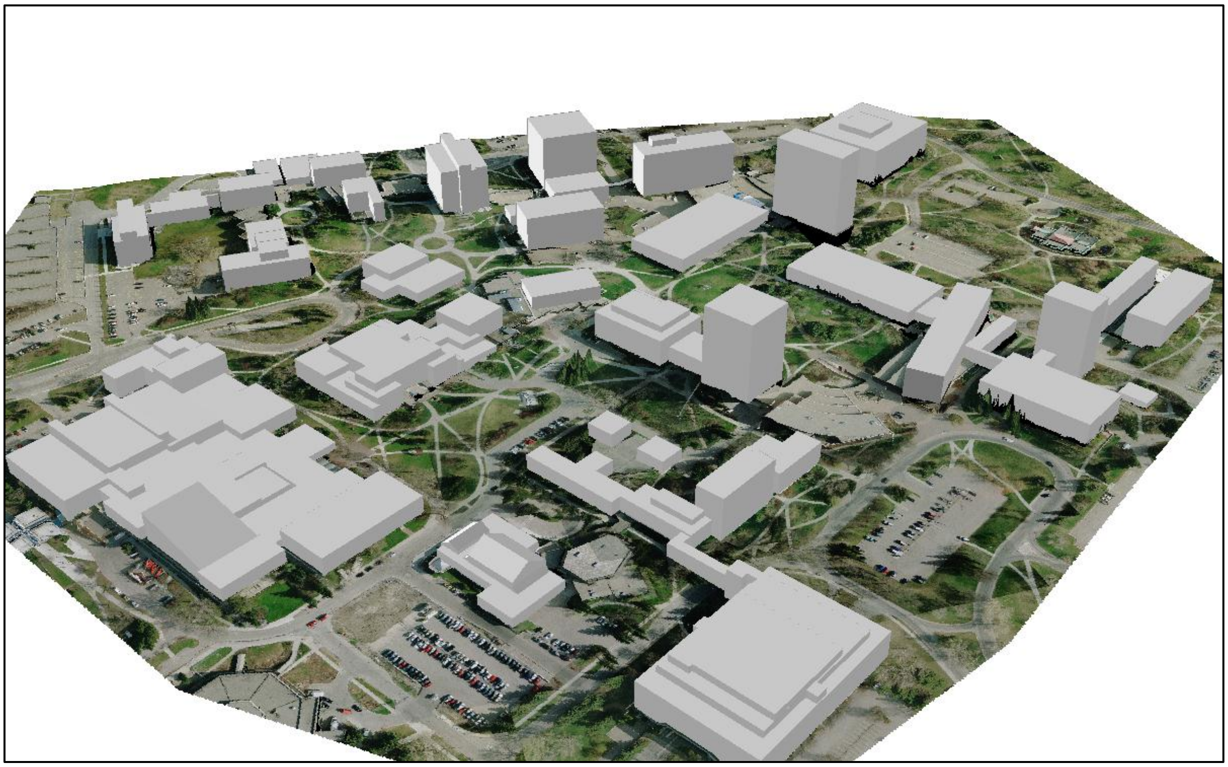


Figure 6.20: Final derived building models in 3D

To demonstrate how applicable the proposed methodology was for the test area, the right-angled-corner buildings/planes were counted. The test area included 32 buildings. As seen in Table 6.5, out of the 32 total buildings, 72% of them have right-angled-corner rooftops which could be reconstructed using the proposed methodology. Nineteen percent of the buildings could be partially reconstructed, which means that some of the planes were not right-angled-corners in the case of buildings that are comprised of multiple planes. Less than 10% of the buildings could not be reconstructed using the proposed methodology because they were not right-angled-corner buildings as seen in Figure 6.21.

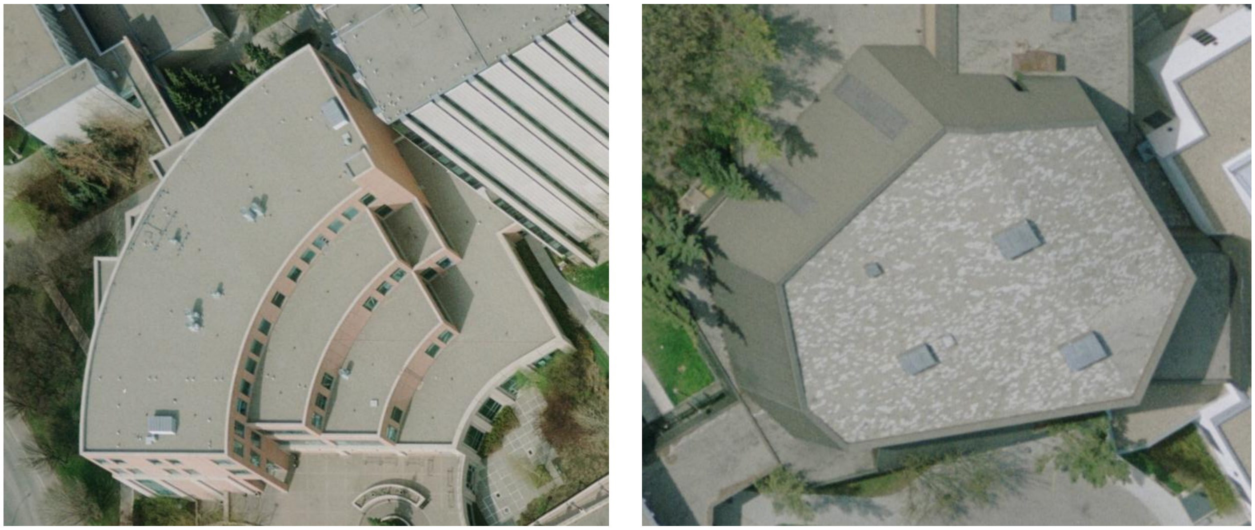


Figure 6.21: Examples of non-right-angled-corner building

Table 6.6 presents the analysis in terms of the number of planes. Of the 112 total planes, 82% were right-angled-corner planes and 18% of the planes were non-right-angled-corner planes that could not be reconstructed using the proposed methodology. In this experiment, all the right-angled-corner planes were reconstructed, with the exception of two that were derived from LiDAR data only.

Table 6.5: Applicability of the proposed methodology to the test area in terms of number of buildings

	Right-angled-corner buildings	**Partially right-angled-corner buildings**	**Non-right-angled-corner buildings**	**Total**
# of buildings	23	6	3	32
Percentage of buildings	72%	19%	9%	100%

Table 6.6: Applicability of the proposed methodology to the test area in terms of number of planes

	Right-angled-corner planes	Non-right-angled-corner planes	Total
# of planes	92	20	112
Percentage of planes	82%	18%	100%

As a final product of the automated building reconstruction process, the generated DBM was converted into a Keyhole Markup Language (KML) format, which can be directly imported to Google Earth, as shown in Figure 6.22.

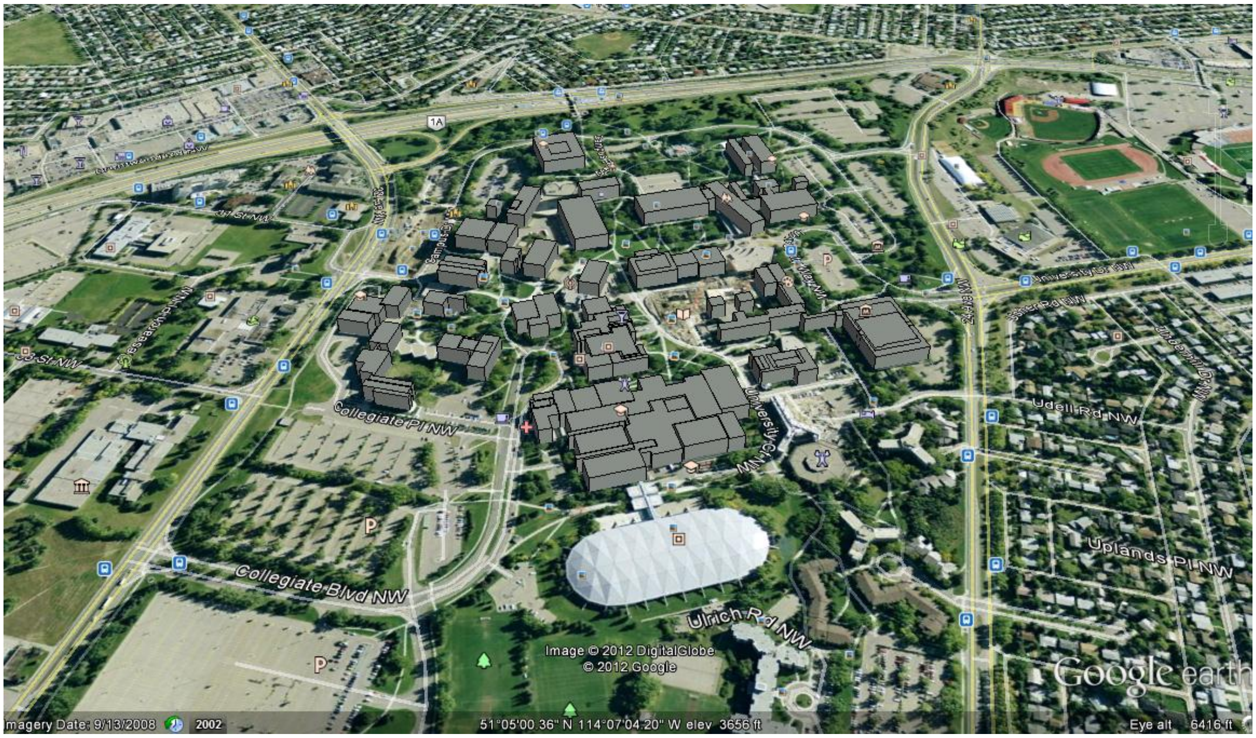


Figure 6.22: Produced DBM from the proposed methodology in KML format

The qualitative evaluation was performed by projecting the buildings' derived boundaries onto corresponding images after the model-based image fitting. Checking whether the adjusted model is aligned properly with edges that represent building boundaries confirms the quality of the final model. To assess the accuracy of the established DBM in a quantitative way, the coordinates of the DBM corner points were compared with those derived manually using the photogrammetric reconstruction procedure. 110 building corner points were measured manually on the multiple overlapping images and then the bundle adjustment process was performed. The quality of photogrammetric reconstruction is defined by its bundle adjustment precision; and the a-posteriori variance factor is 4μm, which is equivalent to 1/3 pixel size. The expected accuracies (σ_X, σ_Y, and σ_Z) of the reconstructed points can be calculated based on the assumed image measurement accuracy, the height, the camera's focal length, and the height-base ratio. Using one pixel image measurement accuracy, the expected planimetric and vertical accuracy were 6cm and 14cm, respectively. The standard deviations of the reconstructed points were minimum ±2cm and maximum ±5cm, which are within the expected accuracy. The statistical properties for the differences in the corner point coordinates between the manual photogrammetric reconstruction and the proposed methodology are shown in Table 6.7.

Table 6.7: Statistical properties for the comparison between the manual photogrammetric reconstruction and the proposed methodology (110 check points)

[Unit: m]	X	Y	Z
Mean	0.07	0.12	0.48
Standard deviation	0.35	0.30	0.43
RMSE	0.35	0.32	0.64

As seen in Table 6.7, the planimetric accuracy was 30cm, which is equivalent to five pixels in the image space. Previous research on the same dataset reported that the RMSE between the manual photogrammetric reconstructions from different operators was also 30cm (Kim, 2008). Given the fact that the buffer size for edge detection/filtering to represent the building boundaries is five pixels, which is equivalent to 30cm GSD, the planimetric accuracy was considered to be numerically high. The mean (i.e., bias) ranged from 7 to 12cm for the planimetric coordinates. The standard deviations for the X and Y coordinates were around 30cm, which was higher than expected. These high deviations were caused by the problem of identifying the corner points of a building rooftop during the manual image measurements. As seen in Figure 6.23, when the corner points were occluded by trees or corner points were not clearly visible due to textures on the wall, it was difficult for the operators to select the exact same corner points in overlapping images. However, the proposed methodology adjusted the model primitives based on the automatic edge extraction. As seen in Figure 6.24(a), a building is represented by the automatically extracted/filtered edges and the model parameters can be adjusted accordingly, which eliminates the need to select the corner point (Figure 6.24(b)).

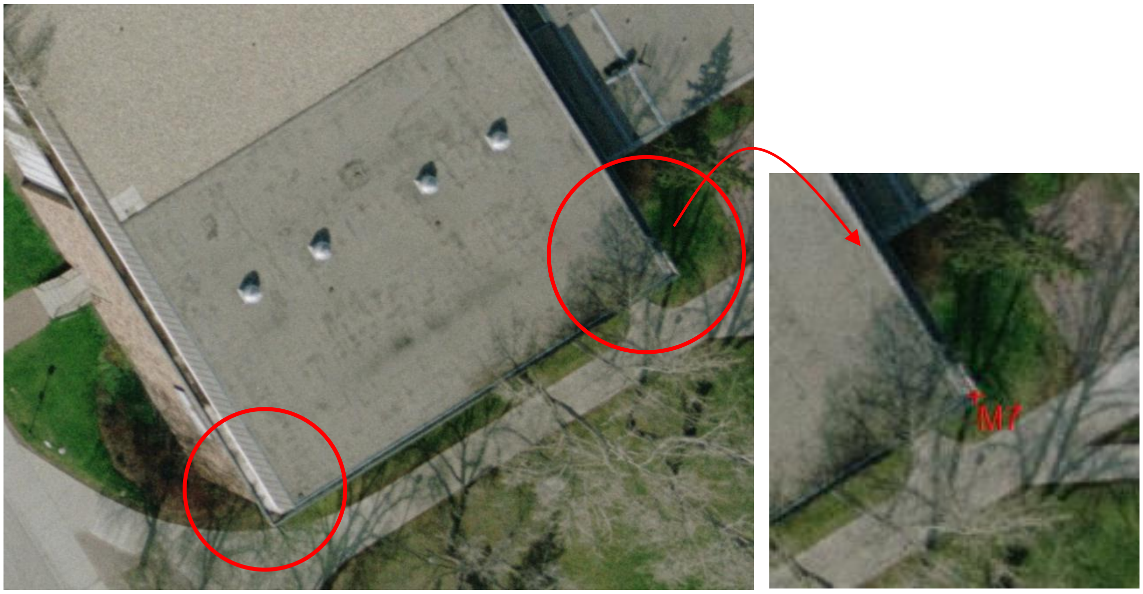
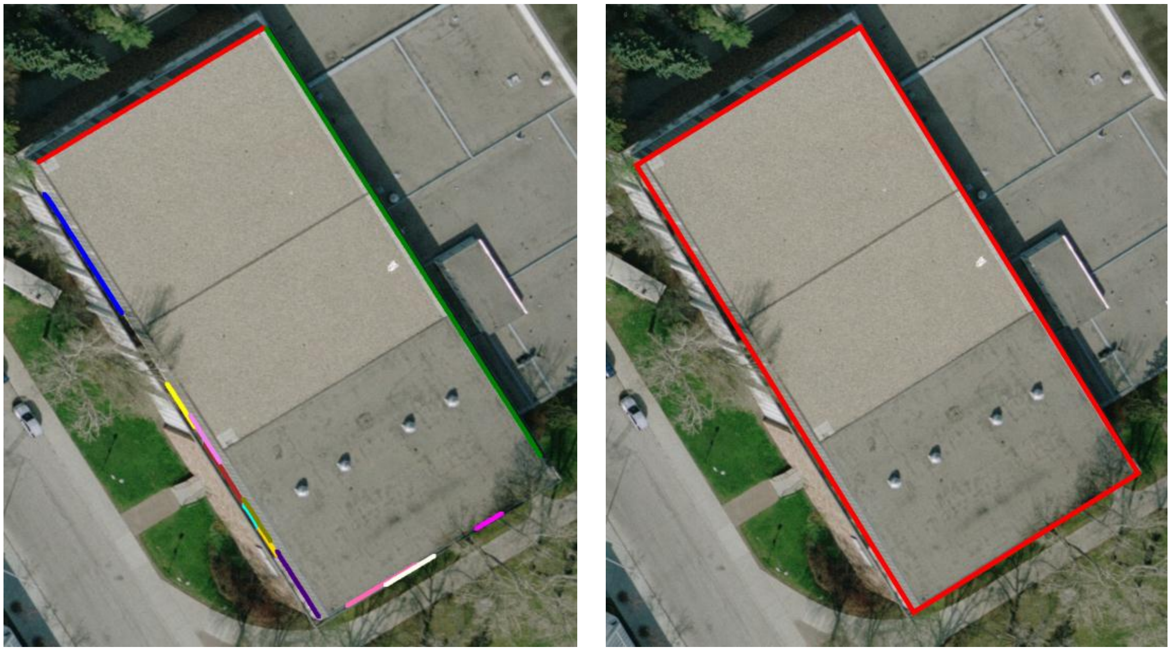


Figure 6.23: Problems with manually-identified corner points



(a) (b)

Figure 6.24: Building representation by automatically-extracted/filtered edges (a) and projection of automatically-reconstructed boundary onto an image

On the other hand, it can be observed that the vertical accuracy was worse than the planimetric accuracy as reported in Table 6.7. This accuracy difference was due to the fact that operator identifies the corner point on top of fences while the proposed methodology reconstructs the corner points on the rooftop plane using the derived plane parameters from the LiDAR segmentation. Therefore, the proposed methodology ensures that the vertical coordinates of the corner points are always defined on the rooftop plane. Figure 6.25 demonstrates the difference between the manually and automatically reconstructed points.

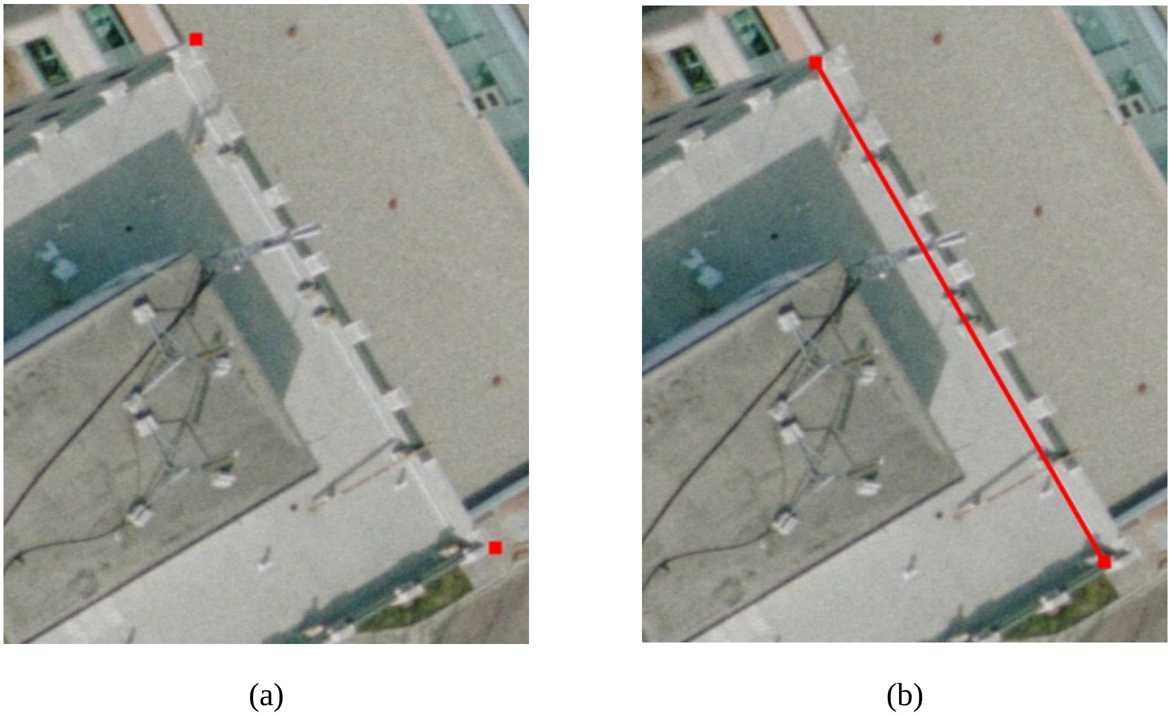


Figure 6.25: An example of a building with a fence: projection of reconstructed points from manual photogrammetric method (a) and projection of reconstructed points from the proposed methodology (b)

The projection of the reconstructed points onto an image in Figure 6.25 shows that the manual reconstruction was carried out on top of the fence while the automatic process was completed on the rooftop plane. The reported mean values for the vertical coordinate comparison (i.e., 48cm in Table 6.7) reveal the existence of height-bias between the two reconstruction processes, which indicates that there was a constant shift due to the anticipated heights of the fences.

To further investigate the improvement of the horizontal accuracy of the derived building models through the model-based image fitting compared to the LiDAR-based one, the accuracy of the building model derived from LiDAR data only was also compared to the manual photogrammetric reconstruction. More specifically, the coordinates of the regularized building model after the RMBR procedure, which were presented in Section 6.4, were compared with the manual photogrammetric reconstruction. The same 110 building corner points used as check points during the final DBM analysis were employed. Table 6.8 presents the statistical properties of the differences between the results from the RMBR process and the manual reconstruction. One can observe that while the vertical accuracy was quite similar to that derived using the proposed methodology reported in Table 6.7, the planimetric accuracy was much worse. This was expected since the photogrammetric data provide better horizontal accuracy compared to LiDAR data. Also, as seen in the circled area in Figure 6.26, the regularized MBR shows varying qualities of the corner points, depending on the structure of the building and the quality of the LiDAR segmentation. This result can be easily improved using photogrammetric data through model-based image fitting.

Table 6.8: Statistical properties for the comparison between the manual photogrammetric reconstruction and the reconstruction using LiDAR data only (110 check points)

[Unit: m]	X	Y	Z
Mean	0.09	0.35	0.48
Standard deviation	0.81	0.78	0.50
RMSE	0.81	0.85	0.69

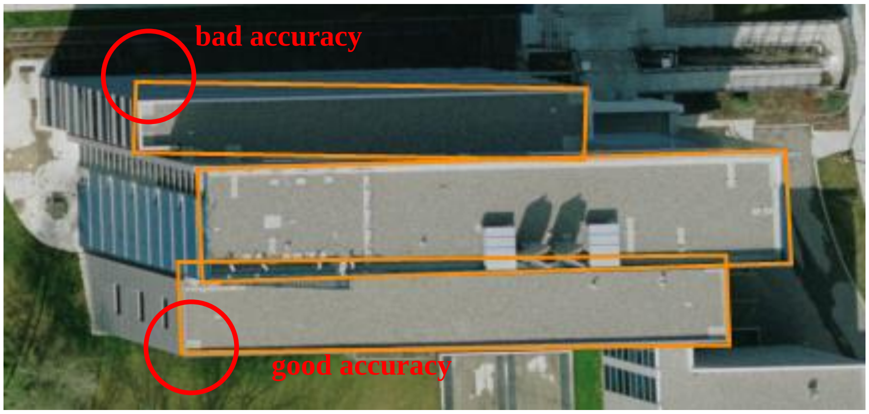


Figure 6.26: Regularized building boundaries through the RMBR process

To demonstrate the impact of the number of utilized images on the quality of the derived models, another dataset which includes more number of images over the same area are used. The dataset used is captured over British Colombia Institute of Technology (BCIT) in Vancouver, Canada. The selected buildings appear in multiple aerial images and airborne LiDAR data are available over the same area. Both datasets were captured from two different flying height (540 and 1,150m), which leads to ground sampling distance of 5 and 10cm for images, and LiDAR point density of 1.5 and 4.0pts/m^2. Since the dataset includes mostly simple rectangular shape buildings and gable roof buildings, two test buildings that are the most common shapes in the data have been selected. The first building is a simple rectangular shape building and the building is partially occluded by trees. The second building is a gable roof shape. Five images which captured the test buildings from different locations of camera exposures are used. Figure 6.27 illustrates approximate locations of camera exposures of five images including flying directions and heights. This figure also includes approximate locations of the test buildings. Figure 6.28 and Figure 6.29 show the test buildings in the different images used in this experiment.

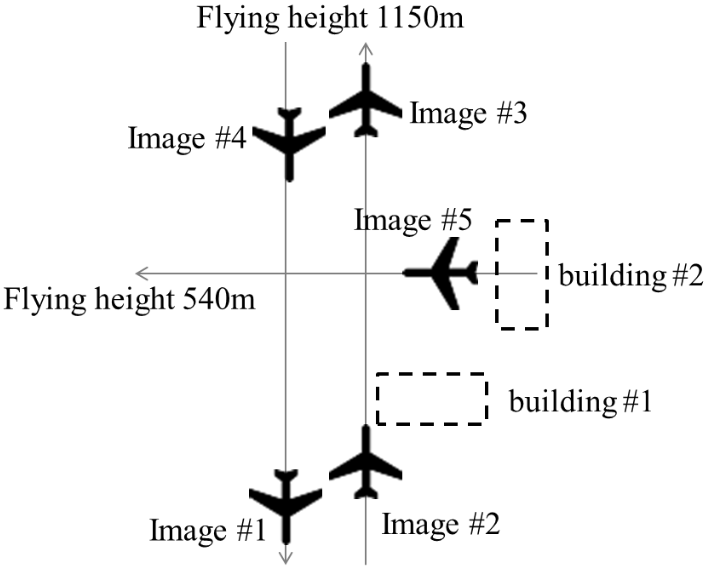


Figure 6.27: Illustration of approximate locations of camera exposures of the imagery and buildings

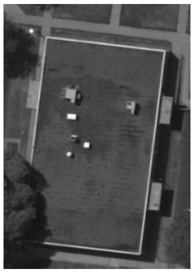
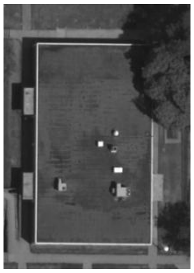
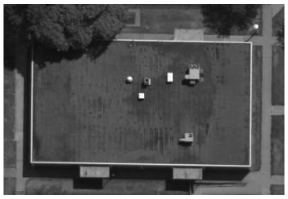
	#1	#2	#3	#4	#5
Test building 1					

Figure 6.28: The first test building appeared in the different images

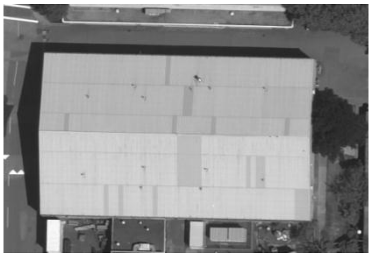
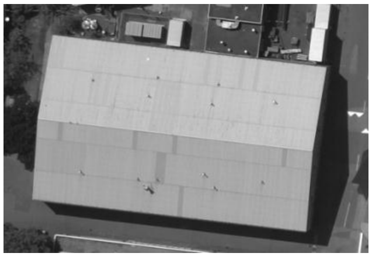
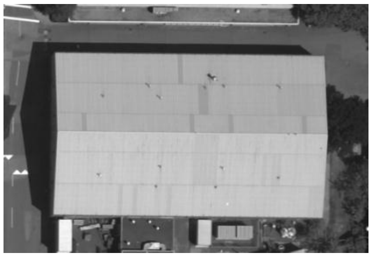
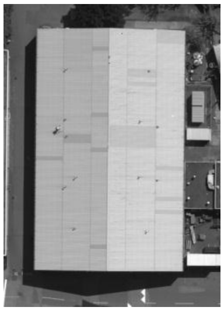
<table>
<tr><td rowspan="4">Test building 2</td><td colspan="2">#1</td><td colspan="2">#2</td></tr>
<tr><td colspan="2"></td><td colspan="2"> </td></tr>
<tr><td>#3</td><td colspan="2">#4</td><td>#5</td></tr>
<tr><td></td><td colspan="2"></td><td></td></tr>
</table>

Figure 6.29: The second test building appeared in the different images

The proposed building detection procedures were performed on the LiDAR dataset and the MBR algorithm was applied to derive the initial information about the building. The initial models were adjusted through the proposed model-based image fitting procedure. The model-based image fitting was applied on using single image (i.e., image #1), two images (i.e., image #1 and #2), four images (i.e., image #1, # 2, #3, and #4), and five images (i.e., image #1, #2, #3, #4, and

#5). To provide a reference for the quality comparison, corner points of the test buildings are reconstructed through multiple light ray intersections of manually measured points.

For a quantitative analysis on the quality of the models, the RMSE between the reconstructed corner points using different number of images and the manual reconstruction are calculated. Table 6.9 and Table 6.10 report the RMSE in X, Y, and Z direction for the first and second test buildings, respectively.

Table 6.9: RMSE between the manual photogrammetric reconstruction and the proposed methodology of the first test building

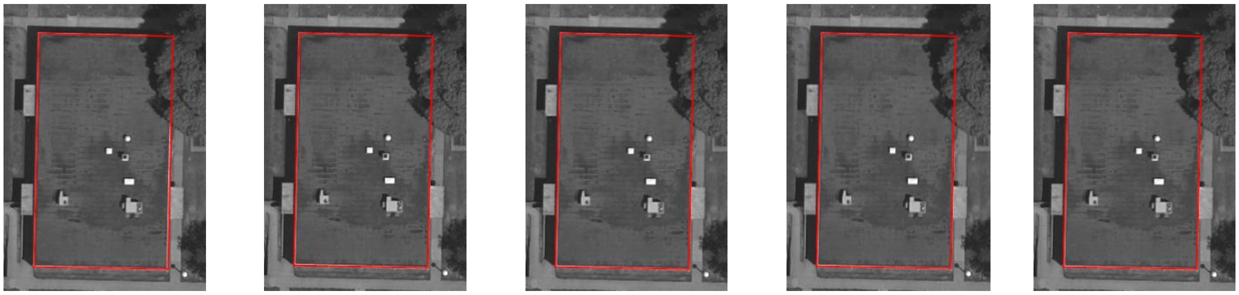
RMSE [m]	X	Y	Z	Total
single image	0.03	0.08	0.23	0.24
two images	0.03	0.09	0.23	0.25
four images	0.05	0.08	0.23	0.24
five images	0.05	0.07	0.23	0.24

Table 6.10: RMSE between the manual photogrammetric reconstruction and the proposed methodology of the second test building

RMSE [m]	X	Y	Z	Total
single image	0.17	0.05	0.39	0.43
two images	0.11	0.06	0.34	0.36
four images	0.10	0.08	0.34	0.36
five images	0.06	0.06	0.33	0.34

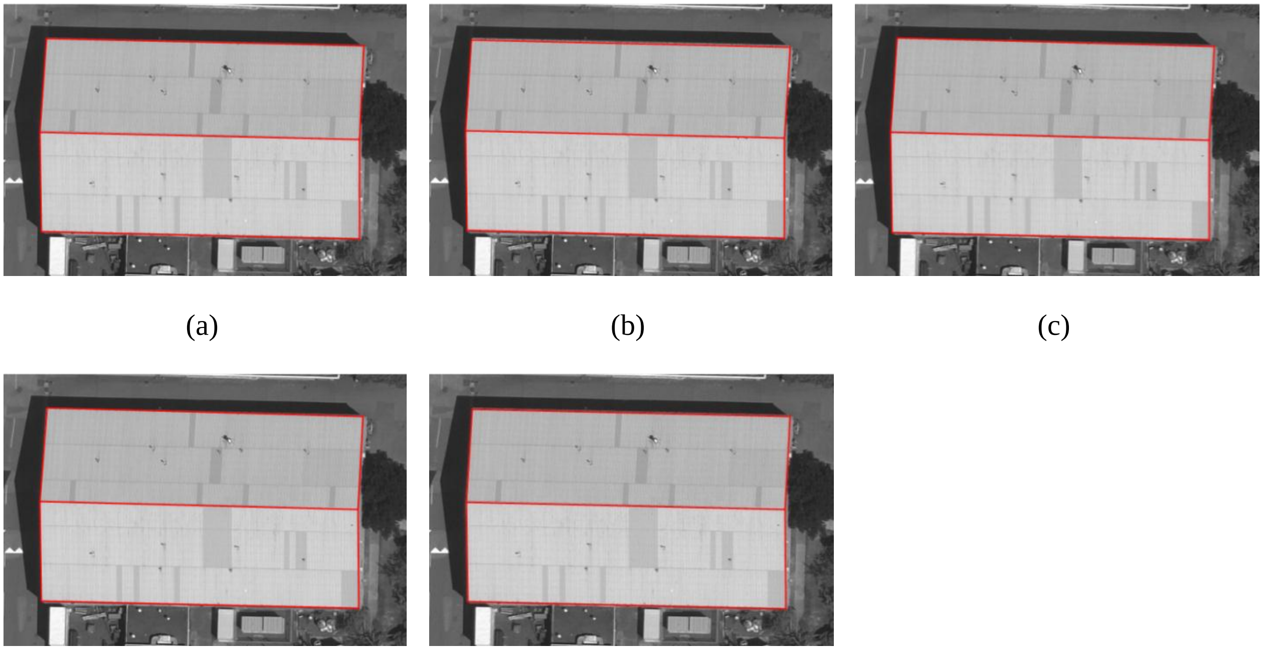
Using 1 pixel (0.006mm) for the image measurement accuracy, σ_x and σ_y is equivalent to 0.1m and σ_z is 0.47m. RMSE from both tests are within the range of expected precision of photogrammetric reconstruction using 1 pixel accuracy. It is worth noting that both results show lower RMSE in X, Y directions than Z direction. It is well known that planimetric precision of the photogrammetric reconstruction is higher than vertical one and it was confirmed by previous RMSE analysis using University of Calgary campus dataset. Since the Z values from the proposed methodology come from LiDAR data which shows higher vertical accuracy, they are believed to be more accurate compared to the one from manual reconstruction. In case of simple rectangular building, the accuracies of the results are similar regardless of the number of images used, even when single image is used. However, the gable roof shows slightly higher RMSE when using one image, but it is still less than 2 pixels. This happens because not enough edges were selected in the image #1. Different images with different view angles provide more redundant observations and increase the chance of providing better results.

The quality of the models also can be compared in a qualitative way by projecting the reconstructed points onto one of the images. Figure 6.30 and Figure 6.31 show the projection of the reconstructed points of the test buildings onto one of the image (i.e., image #1). When visually checked, all of the projected lines coincide with building edge lines and all the cases show similarly good results. Thus, one can confirm that the proposed methodology works well using a single image as discussed, but for complex buildings, using more images would increase the chance of more accurate building models.



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

Figure 6.30: Projection of the reconstructed corner points of the first building onto image #1 using: the manual reconstruction (a), the proposed methodology with single image (b), with two images (c), with four images (d), with five images (e)



(a) (b) (c)

(d) (e)

Figure 6.31: Projection of the reconstructed corner points of the second building onto image #1 using: the manual reconstruction (a), the proposed methodology with single image (b), with two images (c), with four images (d), with five images (e)

6.6 Summary

This chapter presented the experimental results using a real dataset comprised of multiple LiDAR strips and images to demonstrate the feasibility of the proposed methodology. Building hypotheses were generated from the LiDAR data through plane segmentation and ground/non-ground classification. The employed building detection methodology in this research demonstrated a high level of completeness and correctness. However, the derived building boundaries from the LiDAR data show irregular characteristics and therefore need to be regularized. This research proposed rectangular model-based image fitting to improve the building boundaries. In order to decompose the LiDAR boundaries into rectangular primitives, the RMBR algorithm was applied. The derived MBRs from the boundaries that were falsely identified as buildings were successfully removed using two criteria which represent the similarity between the final shape and the LiDAR boundary. The horizontal accuracy of the derived building models from the LiDAR data as a result of the RMBR process was around 90cm when compared with the manual photogrammetric reconstruction. This reported RMSE value reveals that the horizontal accuracy of the building models needed improvement. Therefore, using the derived multiple levels of MBRs from LiDAR data as an initial model primitive, model-based image fitting was applied sequentially. The final shape was derived by an alternating series operation of addition and subtraction for the multiple levels and the results were evaluated qualitatively and quantitatively. Visual inspection of the projected boundary of the buildings onto images reveals improvements in the quality of the building boundary. The planimetric RMSE value between the final product and the manual photogrammetric reconstruction was 30cm, which confirmed that the horizontal accuracy of the building corner points was improved after the model-based image fitting.

Chapter Seven: **Conclusions and recommendations for future work**

This chapter discusses the conclusions from the research findings, which were confirmed by the experimental results using real datasets as well as how they met the objectives of the research. The major contributions of this research are summarized and also, recommendations/suggestions for future work are included.

7.1 Conclusions

As presented in Section 1.2, this research aimed at the development of a hybrid approach for automatic building model generation of right-angled-corner buildings. One of the most important objectives was to achieve a high level of automation with minimum human interaction. Accordingly it aimed at reducing the number of model primitives and deriving the initial model parameters automatically. Another objective of the proposed research is obtaining a high level of accuracy in terms of both horizontal and vertical accuracies. In addition, buildings with different level of details could be reconstructed while generalizing the building models automatically. This research is expected to eliminate the limitation of number of images that can be used and overcome the effect of the partial occlusions. The following discussion will demonstrate how these intended objectives are met in this research and what are the advantages and limitations of the research.

This research presented a robust approach to generate building models automatically from LiDAR data and imagery by combining a data-driven approach and a model-based approach.

One of the most common problems encountered when using LiDAR data within a data-driven approach is determining a way to regularize the LiDAR-derived boundaries. On the other hand, a model-based approach implicitly constrains the rules, which eliminates the need for a regularization process. However, selecting accurate model primitives and determining their initial parameters involves manual interaction. To alleviate the problems of each approach and complement both datasets, the presented research takes advantage of overlapping imagery and LiDAR data that have undergone rigorous QA/QC and registration procedures.

First, LiDAR data were used to detect the buildings automatically; planar segmentation and ground/non-ground classification procedures were performed to differentiate buildings from ground and non-planar objects. The thresholds used for building detection process were logical and justifiable which are not based on a trial and error process. The performance of the building detection process was evaluated by checking the completeness and correctness of its detection ability. The building detection results revealed a high level of completeness and correctness; however, the boundaries did not always represent the actual boundaries of the objects due to the relatively low sampling rate of the airborne LiDAR system. The derived boundaries of the buildings from the LiDAR data therefore indicated the need for improvement, and photogrammetric data were integrated in an attempt to improve the quality of the LiDAR-derived building boundaries. In this research, right-angled-corner buildings (e.g., box, shed-roof, and gable-roof buildings or a mixture thereof) were the main interest, which could be achieved by using rectangular primitives.

Secondly, to eliminate the need to determine the initial model parameters manually, the RMBR algorithm was proposed. The RMBR algorithm can decompose buildings into rectangular models automatically without human intervention or requiring any additional GIS data. To improve the correctness measure for further processing, the LiDAR boundaries were compared to the derived MBRs; and the developed criteria to select only right-angled-corner buildings removed the irregular boundaries from falsely identified buildings. This evaluation procedure thereby retained a high level of completeness and improved the correctness level.

Lastly, sequential model-based image fitting was undertaken to refine the model parameters of the multiple levels of MBRs. The final models were constructed by alternating the operation of subtraction and addition from each level of adjusted MBRs. The quality of the final building models was evaluated through qualitative and quantitative analysis. Visual inspection of the projected boundaries of the buildings onto the images revealed improvements in the quality of the building boundaries. To assess the accuracy of the established DBM in a quantitative way, the coordinates of the DBM corner points were compared with those derived manually using the photogrammetric reconstruction procedure. The RMSE analysis showed that the accuracy of the final DBM was very similar to the manual reconstruction. To investigate the improvement of the horizontal accuracy of the derived building models through the model-based image fitting compared to the LiDAR-based method, the accuracy of the building model derived from LiDAR data only was also compared to the manual photogrammetric reconstruction. Significant improvement of the horizontal accuracy was observed while the vertical accuracy was similar, which is due to the fact that photogrammetric data provide better horizontal accuracy compared to LiDAR data. As confirmed by the visual inspection and the RMSE analysis, the proposed approach produced DBMs with high accuracy. The results were obtained with a high level of automation and minimal human intervention.

One of the advantages of the proposed method, established from the experimental results, is that the amount of processing could be minimized for each step since appropriate quality control of each processing step was performed. Also, the presented research confirmed that there is no limitation in terms of the number of images that can be utilized.

Another area of improvement is minimizing the effect of partial occlusions, which is the most common problem for image-based approaches. Most of the data-driven approaches require applying constraints during the reconstruction process and the use of image matching from stereo images. Moreover, these approaches assume availability of complete edges from both images. This research proved the robustness against occlusions and when only partial edges exist, the building model can still be reconstructed with the proposed method.

The proposed method also can generate different levels of detail such that the model can be generalized to LoD 1 producing a prismatic building and can obtain details equivalent to LoD 2 as well. This process is affected by the LiDAR point density and the size of the target building, thus, when the LiDAR point density is low and/or the object is smaller than the utilized threshold, the model cannot be properly reconstructed. This research utilized only a rectangular model, which greatly reduced the number of models to deal with, but it still was able to model complex buildings automatically. Even though accurate final building models were reconstructed with a high level of automation using the rectangular model only, its biggest limitation is that it can model only the types of buildings which decompose into rectangles.

In addition, when LiDAR data are not available, the proposed method can be applied using only images. 3D point cloud can be generated using available image matching algorithms from multiple images and the proposed building detection process can be performed on them. Then, the model-based image fitting reconstruction can be employed to improve the image-matching-based photogrammetric reconstruction. The potentials of the proposed methodology using only images were demonstrated in the application of detecting the sub-millimeter level beam deformation and were presented in Kwak et al. (2012).

7.2 Summary of research contributions

In summary, this research introduces an innovative methodology for automatically reconstructing accurate right-angled-corner building models from LiDAR and photogrammetric data. A high level of automation is achieved by combining data-driven and model-driven processing strategies. These strategies are devised to take full advantage of both photogrammetric and LiDAR data (i.e., the highest level of information that could be reliably extracted from both datasets are utilized for building detection and reconstruction). An additional contribution is the automatic generation of reliable initial model parameters from LiDAR data through the newly developed RMBR process. The RMBR algorithm is capable of dealing with several building models while only relying on rectangular primitives to represent the buildings in question. This introduces further advantage by simplifying the processing procedure while being able to deal with complex and varying models. For medium-accuracy applications, the derived model from the RMBR process can be used as a digital building model, whose planimetric accuracy is equivalent to the point-spacing of the utilized LiDAR data. With the increase in the pulse repetition rate of modern LiDAR system, the planimetric accuracy of the RMBR-based building models will get better. Another advantage of the introduced RMBR process is the ability to automatically generalize the reconstructed building models according to the required level of detail.

To enhance the horizontal accuracy of the RMBR-based DBM, a sequential model-based image fitting is developed by considering derived edges from overlapping imagery. Contrary to previous literature, the model-based strategy can generate building models using only one image as well as any number of overlapping images. Moreover, the proposed procedure does not require the identification of conjugate features or building corners in overlapping imagery. This characteristic reduces the sensitivity of the proposed methodology to partial occlusions in the imagery. A modified least squares adjustment procedure has been utilized to simplify the model-based image fitting process.

Another advantage of the developed methodology is that the performance of the intermediate results is assessed to ensure that the buildings to be reconstructed belong to the category of right-angled-corner buildings through a sequence of quality control measures. The quality control at each step determines in advance whether the building in question can be reconstructed using the proposed methodology or not, thereby avoiding unnecessary processing of buildings that fall beyond the ones of interest. This QC procedure is useful for classifying the detected buildings into right-angled-corner buildings and non-right-angled-corner buildings. Once the building model generation is completed, the stochastic properties of the generated DBMs indicate the precision of the adjustment. The developed methodologies have been incorporated in a software package that allowed for a straightforward testing using real datasets.

7.3 Recommendations for future work

As previously mentioned, this research work only models buildings with right-angled-corners. Future work therefore could include increasing the applicability of the proposed algorithm for other building shapes in order to obtain more extensive building models while maintaining a high level of accuracy and automation. Since this research can automatically distinguish right-angled and non-right-angled-corner buildings, the future work can focus on the reconstruction of non-right-angled-corner buildings. As previously mentioned, the LiDAR regularized boundaries (i.e., the derived models from the RMBR process) can be used as DBMs for the non-right-angled corner buildings. Otherwise, LiDAR derived boundaries can be adjusted/improved by minimizing the normal distances between the boundaries and extracted edges from images or image-matching-based reconstruction. Another alternative can be an incorporation of different models primitives only for the non-right-angled building reconstruction. As mentioned, since majority of the buildings (i.e., right-angled-corner buildings) can be reconstructed using the proposed methodology, the amount of processing for the remaining buildings (i.e., non-right-angled-corner buildings) will be minimized.

Another area to investigate is the possibility of extending the proposed algorithm into terrestrial laser scanning data to include façade modeling, which could increase the level of detail of the buildings. The further investigation of reconstructing 3D building models using only images can be performed as well. The available dense matching algorithm can be incorporated and the quality of the final building models derived through this process can be compared. Future work could include carrying out more testing on different datasets with different point densities and image resolutions, as well as analyzing the effects of those characteristics on the accuracy, completeness, and correctness of the models. Finally, user interface development is required to show the instances where the approach is not applicable while allowing for semi-automated refinement of the DBM.

References

Arefi, H., Engels, J., and Hahn, M. (2008). "Levels of detail in 3D building reconstruction from LiDAR data." *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, Beijing, 485–490.

Awrangjeb, M., Ravanbakhsh, M., and Fraser, C. S. (2010). "Automatic detection of residential buildings using LIDAR data and multispectral imagery." *ISPRS Journal of Photogrammetry and Remote Sensing*, 65(5), 457–467.

Baillard, C., Schmid, C., Zisserman, A., and Fitzgibbon, A. (1999). "Automatic line matching and 3D reconstruction of buildings from multiple views." *ISPRS Conference on Automatic Extraction of GIS Objects from Digital Imagery*, 69–80.

Besl, P. J., and Jain, R. C. (1988). "Segmentation through variable-order surface fitting." *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 10(2), 167 –192.

Brenner, C. (2005). "Building reconstruction from images and laser scanning." *International Journal of Applied Earth Observation and Geoinformation*, 6(3–4), 187–198.

Brenner, C., Haala, N., and Fritsch, D. (2001). "Towards fully automated 3D city model generation."

Canny, J. (1986). "A Computational Approach to Edge Detection." *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, PAMI-8(6), 679 –698.

Chaudhuri, D., and Samal, A. (2007). "A simple method for fitting of bounding rectangle to closed regions." *Pattern Recognition*, 40(7), 1981–1989.

Chen, L. C., Teo, T. A., Shao, Y. C., Lai, Y. C., and Rau, J. Y. (2004). "Fusion of LIDAR data and optical imagery for building modeling." *International Archives of Photogrammetry and Remote Sensing*, 35(B4), 732–737.

Demir, N., and Baltsavias, E. (2012). "Automated modeling of 3D building roofs using Image and LiDAR data." *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, Melbourne, Australia, 35–40.

Faig, W., and Widmer, T. (2000). "Automatic building extraction from aerial images." *International Archives of Photogrammetry and Remote Sensing*, 33(B7), 1708–1715.

Filin, S., and Pfeifer, N. (2005). "Neighborhood systems for airborne laser data." *Photogrammetric Engineering & Remote Sensing*, 71(6), 743–755.

Filin, S., and Pfeifer, N. (2006). “Segmentation of airborne laser scanning data using a slope adaptive neighborhood.” *ISPRS journal of Photogrammetry and Remote Sensing*, 60(2), 71–80.

Fischer, A., Kolbe, T., Lang, F., Cremers, A. ., F_orstner, W., Pl_umer, L., and Steinhage, V. (1998). “Extracting Buildings from Aerial Images using Hierarchical Aggregation in 2_D and 3_D.” *Computer Vision and Image Understanding*, 72(2), 195–203.

Flamanc, D., Maillet, G., and Jibrini, H. (2003). “3D city models: an operational approach using aerial images and cadastral maps.” *International Archives of Photogrammetry Remote Sensing and Spatial Information Sciences*, 34(3/W8), 53–58.

Freeman, H., and Shapira, R. (1975). “Determining the minimum-area encasing rectangle for an arbitrary closed curve.” *Communications of the ACM*, 18(7), 409–413.

Friedman, J. H., Bentley, J. L., and Finkel, R. A. (1977). “An Algorithm for Finding Best Matches in Logarithmic Expected Time.” *ACM Transactions on Mathematical Software*, 3(3), 209–226.

Fuchs, C., Förstner, W., Gülch, E., Heipke, C., and Eder, K. (1998). *OEEPE Survey on 3D-city Models*. Bundesamt für Kartographie und Geodäsie.

Gröger, G., Kolbe, T. ., Nagel, C., and Häfele, K. (Eds.). (2012). “OGC City Geography Markup Language (CityGML) En-coding Standard.” 2012 Open Geospatial Consortium.

Gruber, G., Menard, C., and Schachinger, B. (2008). “Evaluation of the Geometric Accuracy of Automatically Recorded 3D – City Models Compared to GIS-Data.” *The European Information Society*, Lecture Notes in Geoinformation and Cartography, L. Bernard, A. Friis-Christensen, and H. Pundt, eds., Springer Berlin Heidelberg, 67–78.

Gruen, A., Kuebler, O., and Agouris, P. (Eds.). (1995). *Automatic Extraction of Man-Made Objects from Aerial Space Images*. Springer.

Gruen, A., and Wang, X. (1998). “CC-Modeler: a topology generator for 3-D city models.” *ISPRS Journal of Photogrammetry and Remote Sensing*, 53(5), 286–295.

Guercke, R., Götzelmann, T., Brenner, C., and Sester, M. (2011). “Aggregation of LoD 1 building models as an optimization problem.” *ISPRS Journal of Photogrammetry and Remote Sensing*, 66(2), 209–222.

Haala, N. (1995). “3D Building Reconstruction using Linear Edge Segments.” Photogrammetric Week, D. Fritsch and D. Hobbie, eds., Wichmann, Karlsruhe, 19–28.

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

Habib, A., Al-Durgham, M., Kersting, A. P., and Quackenbush, P. (2009). “Error budget of LiDAR systems and quality control of the derived point cloud.” *Photogrammetric Engineering and Remote Sensing*, 75(9), 1093–1108.

Habib, A., Bang, K. I., Kersting, A. P., and Chow, J. (2010). “Alternative Methodologies for LiDAR System Calibration.” *Remote Sensing*, 2(3), 874–907.

Habib, A., Jarvis, A., Kersting, A. P., and Alghamdi, Y. (2008). “Comparative analysis of georeferencing procedures using various sources of control data.” *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, Beijing, China, 1147–1152.

Habib, A., Kwak, E., and Al-Durgham, M. (2011). “Model-Based Automatic 3D Building Model Generation by Integrating LiDAR and Aerial Images.” *Archives of Photogrammetry, Cartography and Remote Sensing*, 22, 187–200.

Habib, A., and Morgan, M. F. (2005). “Stability analysis and geometric calibration of off-the-shelf digital cameras.” *Photogrammetric Engineering & Remote Sensing*, 71(6), 733–741.

Hammoudi, K., and Dornaika, F. (2011). “A Featureless Approach to 3D Polyhedral Building Modeling from Aerial Images.” *Sensors*, 11(1), 228–259.

Heipke, C., Mayer, H., Wiedemann, C., and Jamet, O. (1997). “Evaluation of Automatic Road Extraction.” *In: International Archives of Photogrammetry and Remote Sensing*, 47–56.

Henricsson, O., and Baltsavias, E. (1997). “3-D Building Reconstruction with ARUBA: A Qualitative and Quantitative Evaluation.” *Automatic Extraction of Man-Made Objects from Aerial and Space Images (II)*, Verlag, 65–76.

Hermosilla, T., Ruiz, L. A., Recio, J. A., and Estornell, J. (2011). “Evaluation of Automatic Building Detection Approaches Combining High Resolution Images and LiDAR Data.” *Remote Sensing*, 3(6), 1188–1210.

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

Huertas, A., and Nevatia, R. (1988). “Detecting Buildings in Aerial Images.” *Computer Vision, Graphics, and Image Processing*, 41(2), 131–152.

Kaartinen, H., and Hyyppä, J. (2006). “Sensor and Data Fusion Contest: Information for Mapping from Airborne SAR and Optical Imagery (Phase I).”

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(Part 3), W4.

Karner, K., Hesina, G., Maierhofer, S., and Tobler, R. F. (2006). “Improved reconstruction and rendering of cities and terrains based on multispectral digital aerial images.” *Proceedings of CORP*, 299–304.

Kim, C. (2008). “Object-based Integration of Photogrammetric and LiDAR Data for Accurate Reconstruction and Visualization of Building Models.” Ph.D. thesis, University of Calgary.

Kim, C., and Habib, A. (2009). “Object-Based Integration of Photogrammetric and LiDAR Data for Automated Generation of Complex Polyhedral Building Models.” *Sensors*, 9(7), 5679–5701.

Kim, Z. W., and Nevatia, R. (2004). “Automatic description of complex buildings from multiple images.” *Computer Vision and Image Understanding*, 96(1), 60–95.

Koch, K.-R. (1988). *Parameter Estimation and Hypothesis Testing in Linear Models*. Springer-Verlag New York, Inc., New York, NY, USA.

Kolbe, T. H., Gröger, G., and Plümer, L. (2005). “CityGML–3D city models and their potential for emergency response.” *Geospatial information technology for emergency response*, 257.

Kraus, K. (1993). *Photogrammetry*. Ferdinand Dummlers Verlag.

Kwak, E., Detchev, I., Ayman, H., El-Badry, M., and Hughes, C. (2012). “Precise photogrammetric reconstruction using model-based image fitting for 3D beam deformation monitoring.” *Journal of Surveying Engineering*, (Accepted).

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

Lari, Z., and Habib, A. (2012). “Segmentation-Based Classification of 3D Laser Data.” *ASPRS Annual Conference*, Sacramento, California.

Lari, Z., and Habib, A. (2013). “New Approaches for Estimating the Local Point Density and its Impact on Lidar Data Segmentation.” *Photogrammetric Engineering & Remote Sensing*, 79(2), Accepted.

Lari, Z., Habib, A., and Kwak, E. (2011). "An Adaptive Approach for Segmentation of 3D Laser Point Cloud." *International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, Calgary, Canada.

Lee, D. H., Lee, K. M., and Lee, S. U. (2008). "Fusion of lidar and imagery for reliable building extraction." *Photogrammetric Engineering and Remote Sensing*, 74(2), 215–225.

Lee, I., and Schenk, T. (2001). "Autonomous extraction of planar surfaces from airborne laser scanning data." *ASPRS annual conference*, St. Louis, MO, USA.

Lin, C., and Nevatia, R. (1998). "Building detection and description from a single intensity image." *Computer vision and image understanding*, 72(2), 101–121.

Lowe, D. G. (1991). "Fitting parameterized three-dimensional models to images." *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 13(5), 441–450.

Ma, R. (2004). "Building model reconstruction from lidar data and aerial photographs." Ph.D. thesis, The Ohio State University.

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.

Mayer, H. (1999). "Automatic Object Extraction from Aerial Imagery—A Survey Focusing on Buildings." *Computer Vision and Image Understanding*, 74(2), 138–149.

McGlone, J. C., and Shufelt, J. A. (1994). "Projective and object space geometry for monocular building extraction." *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 54 –61.

Mikhail, E. M. (1976). *Observations and least squares*. University Press of America.

Oude Elberink, S., and Vosselman, G. (2011). "Quality analysis on 3D building models reconstructed from airborne laser scanning data." *ISPRS Journal of Photogrammetry and Remote Sensing*, 66(2), 157–165.

Rottensteiner, F., Baillard, C., Sohn, G., and Gerke, M. (2011). "ISPRS Test Project on Urban Classification and 3D Building Reconstruction."

Rottensteiner, F., and Jansa, J. (2002). "Automatic extraction of buildings from LIDAR data and aerial images." *International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, 34(4), 569–574.

Rottensteiner, F., Sohn, G., Jung, J., Gerke, M., Baillard, C., Benitez, S., and Breitkopf, U. (2012). "The ISPRS benchmark on urban object classification and 3D building reconstruction." *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, Melbourne, Australia.

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.

Rottensteiner, F., Trinder, J., Clode, S., and Kubik, K. (2007). "Building detection by fusion of airborne laser scanner data and multi-spectral images: Performance evaluation and sensitivity analysis." *ISPRS Journal of Photogrammetry and Remote Sensing*, 62(2), 135–149.

Roux, M., and McKeown, D. M. (1994). "Feature matching for building extraction from multiple views." *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 46 –53.

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.

Samet, H. (1984). "The Quadtree and Related Hierarchical Data Structures." *Journal ACM Computing Surveys*, 16(2), 187–260.

Sampath, A., and Shan, J. (2007). "Building boundary tracing and regularization from airborne LiDAR point clouds." *Photogrammetric Engineering & 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.

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), 1425–1440.

Suveg, I., and Vosselman, G. (2000). "3D reconstruction of Building Models." *International archives of photogrammetry and remote sensing*, 33(B2; PART 2), 538–545.

Suveg, I., and Vosselman, G. (2004). "Reconstruction of 3D building models from aerial images and maps." *ISPRS Journal of Photogrammetry and Remote Sensing*, 58(3–4), 202–224.

Tseng, Y. H., and Wang, S. (2003). “Semiautomated building extraction based on CSG model-image fitting.” *Photogrammetric engineering and remote sensing*, 69(2), 171–180.

UN-Habitat. (2012). *State of the world’s cities 2012/2013: Prosperity of Cities*. Earthscan/James & James.

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

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

Vosselman, G., and Veldhuis, H. (1999). “Mapping by dragging and fitting of wire-frame models.” *Photogrammetric Engineering & Remote Sensing*, 65(7), 769–776.

Vosselman, V., and Maas, H.-G. (Eds.). (2010). *Airborne and Terrestrial Laser Scanning*. Taylor & Francis.

Vu, T. T., Yamazaki, F., and Matsuoka, M. (2009). “Multi-scale solution for building extraction from LiDAR and image data.” *International Journal of Applied Earth Observation and Geoinformation*, 11(4), 281–289.

Weidner, U., and Förstner, W. (1995). “Towards automatic building extraction from high-resolution digital elevation models.” *ISPRS Journal of Photogrammetry and Remote Sensing*, 50(4), 38–49.

Xie, W., Zhou, G., and Xue, Y. (2009). “3D detailed building model extraction for urban large-scale orthorectification.” *Urban Remote Sensing Event, 2009 Joint*, 1 –6.

Zebedin, L., Bauer, J., Karner, K., and Bischof, H. (2008). “Fusion of Feature- and Area-Based Information for Urban Buildings Modeling from Aerial Imagery.” *Proceedings of the 10th European Conference on Computer Vision: Part IV*, ECCV ’08, Springer-Verlag, Berlin, Heidelberg, 873–886.

Appendix A:

The partial derivatives of each rectangle vertex with respect to each model parameter

	v_1	v_2	v_3	v_4
$\left(\frac{\partial X_i}{\partial X_0}, \frac{\partial Y_i}{\partial X_0}, \frac{\partial Z_i}{\partial X_0}\right)$	(1, 0, a)	(1, 0, a)	(1, 0, a)	(1, 0, a)
$\left(\frac{\partial X_i}{\partial Y_0}, \frac{\partial Y_i}{\partial Y_0}, \frac{\partial Z_i}{\partial Y_0}\right)$	(0, 1, b)	(0, 1, b)	(0, 1, b)	(0, 1, b)
$\left(\frac{\partial X_i}{\partial d_w}, \frac{\partial Y_i}{\partial d_w}, \frac{\partial Z_i}{\partial d_w}\right)$	(0, 0, 0)	(0, 0, 0)	(r_{12}, r_{22}, r_{32})	(r_{12}, r_{22}, r_{32})
$\left(\frac{\partial X_i}{\partial d_l}, \frac{\partial Y_i}{\partial d_l}, \frac{\partial Z_i}{\partial d_l}\right)$	(0, 0, 0)	(r_{12}, r_{22}, r_{32})	(r_{11}, r_{21}, r_{31})	(0, 0, 0)
$\left(\frac{\partial X_i}{\partial \kappa_m}, \frac{\partial Y_i}{\partial \kappa_m}, \frac{\partial Z_i}{\partial \kappa_m}\right)$	(0, 0, 0)	$(-\ell r_{12}, \ell r_{22}, \ell r_{32})$	$(-\ell r_{12} - \mathscr{w} r_{11}, \ell r_{22} - r_{21}, \ell r_{32} - \mathscr{w} r_{31})$	$(-\mathscr{w} r_{11}, -\mathscr{w} r_{21}, -\mathscr{w} r_{31})$

The partial derivatives of collinearity equations with respect to $\boldsymbol{X_i, Y_i, Z_i}$

vertex	$\frac{\partial f}{\partial X_i}$	$\frac{\partial f}{\partial Y_i}$	$\frac{\partial f}{\partial Z_i}$
x	$-\mathrm{c}\frac{r_{11}D - r_{13}N_x}{D^2}$	$-\mathrm{c}\frac{r_{21}D - r_{23}N_x}{D^2}$	$-\mathrm{c}\frac{r_{31}D - r_{33}N_x}{D^2}$
y	$-\mathrm{c}\frac{r_{12}D - r_{13}N_y}{D^2}$	$-\mathrm{c}\frac{r_{22}D - r_{23}N_y}{D^2}$	$-\mathrm{c}\frac{r_{32}D - r_{33}N_y}{D^2}$

Where,

$$N_x = r_{11}(X - X_0) + r_{21}(Y - Y_0) + r_{31}(Z - Z_0)$$

$$N_y = r_{12}(X - X_0) + r_{22}(Y - Y_0) + r_{32}(Z - Z_0)$$

$$D = r_{13}(X - X_0) + r_{23}(Y - Y_0) + r_{33}(Z - Z_0)$$