AUTOMATED 3D BUILDING MODELLING FROM AIRBORNE LiDAR DATA

Yuxiang He

Submitted in total fulfilment of the requirements for the degree of

Doctor of Philosophy

February, 2015

DEPARTMENT OF INFRASTRUCTURE ENGINEERING

SCHOOL OF ENGINEERING

THE UNIVERSITY OF MELBOURNE

PRODUCED ON ARCHIVAL QUALITY PAPER
Abstract

The modelling and representation of large-scale urban scenes in digital format is a fundamental problem in modern mapping. Concise and meaningful 3D building models provide a virtual environment for simulation applications involving the urban form, such as disaster management and mitigation, city planning and assisted navigation.

Various remote sensing datasets have being employed for this purpose. Point clouds from laser scanners and images from optical sensors are commonly used. These sensor systems are deployed on airborne or terrestrial mapping platforms. Particularly, airborne Light Detection and Ranging (LiDAR), or airborne laser scanning (ALS), has received a lot of attention, as it provides high-accuracy and high-resolution elevation data to represent city-scale landscapes.

This thesis presents a complete framework for 3D building reconstruction and modelling from airborne LiDAR data. The reconstruction procedure is decomposed into distinct stages and a workflow is designed by exploring a range of techniques. The workflow consists of three distinctive modules: 1) LiDAR data filtering, 2) building footprint extraction, and 3) building rooftop modelling. In the first module, an automated filtering algorithm is developed, with classification of the raw LiDAR data to identify terrain and non-terrain points. A powerful segmentation algorithm is proposed, in which a progressive energy minimization scheme is adopted over a graphic model based on terrain characteristics. In the second and foremost module, building regions are further detected from non-terrain data, and this is followed by building footprint extraction based on a hybrid reconstruction, both explicitly and implicitly. In the last module, LiDAR data inside of building footprints is further processed so as to extract roof facets using spectral clustering techniques. A reconstruction algorithm exploiting local topologies of planar patches, along with global adjustment of the footprint to achieve qualified polyhedral models, is then presented.

Experimental evaluation of the developed algorithms using LiDAR data over several urban scenes, from different cities and with various point densities, has demonstrated their utility, robustness and effectiveness for accurate and
comprehensive 3D city modelling. The experiments and the results of the evaluation according to several criteria show that more than 91\% of the roof facets over 10m² are correctly extracted, and the achieved accuracy of the building model is better than 1m and 0.35m in planimetry and height, respectively. The developed system can serve as a robust and automatic tool to extract buildings for urban modelling.
Declaration

This is to certify that

1) The thesis comprises only my original work towards the PhD,

2) Due acknowledgement has been made in the text to all other material used,

3) The thesis is less than 100,000 words in length exclusive of tables, figures, bibliographies and appendices.

Yuxiang He
Department of Infrastructure Engineering
University of Melbourne
Parkville, Victoria 3010
Australia
Acknowldgements

In writing this acknowledgement, many people and events occurred to me. This dissertation would not have been possible without the help or encouragement from them. I therefore wish to express my gratitude to all of them for their support and encouragement throughout my PhD study. Sincere appreciation also goes to colleagues who are not mentioned here.

First of all, I would like to express my greatest gratitude and thanks to both of my supervisors, Prof. Clive Fraser and Assoc Prof. Chunsun Zhang. Their insightful guidance, inspiring advice and scientific knowledge have been invaluable during my three years of study at the University of Melbourne. Many thanks are also due to Prof. Stephan Winter for his support.

Appreciation also goes to the staff and students of the Department of Infrastructure Engineering at the University of Melbourne and to members of the Cooperative Research Centre for Spatial Information (CRCSI) for inspiration, administrative assistance and friendship. Special thanks go to Drs. Christos Stamatopoulos, Mohammad Awrangjeb, Graeme Kernich and Nathan Quadros, and to Wendy Jackson, Jessica Keysers, Jane Inall, Melanie Plumb, Rose Macey and Pauline Woolcock. My CRCSI lab mates: Grant Hausler, Ebadat Ghanbari Parmehr, Joost Kuckartz, and Zhaokun Zhu are also thanked for their friendship and constructive advice.

I am thankful to the University of Melbourne and the CRCSI for supporting my research and for providing an excellent work environment. Thanks also go to 43pl members for the provision of LiDAR data, and for their feedback to the CRCSI regarding the progress of my research.

I feel lucky to have friends at the University of Melbourne who make my life a precious experience. These include Drs. Pan Wang, Yiqun Chen, Yin-Yu Lu, Linjie Guan, Mingzheng Shi, Haohui Chen and many others. Most important of all, I am grateful for the support from my parents in China. Finally, very special thanks go to my wife Zixue Qiu, a beautiful, smart and generous companion; I love you.
## Abbreviations

<table>
<thead>
<tr>
<th>Abbreviation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ALS</td>
<td>Airborne Laser Scanning</td>
</tr>
<tr>
<td>B-rep</td>
<td>Boundary representation</td>
</tr>
<tr>
<td>BSP</td>
<td>Binary Space Partitioning</td>
</tr>
<tr>
<td>CE</td>
<td>Coverage Error</td>
</tr>
<tr>
<td>CIR</td>
<td>Colour infrared</td>
</tr>
<tr>
<td>CityGML</td>
<td>City Geographic Markup Language</td>
</tr>
<tr>
<td>CSG</td>
<td>Constructive Solid Geometry</td>
</tr>
<tr>
<td>DD</td>
<td>Direction Difference</td>
</tr>
<tr>
<td>DP</td>
<td>Douglas-Peucker</td>
</tr>
<tr>
<td>DSM</td>
<td>Digital Surface Model</td>
</tr>
<tr>
<td>DTM</td>
<td>Digital Terrain Model</td>
</tr>
<tr>
<td>GIS</td>
<td>Geospatial Information Systems</td>
</tr>
<tr>
<td>GPC</td>
<td>General Polygon Clipper</td>
</tr>
<tr>
<td>GPS</td>
<td>Global Positioning System</td>
</tr>
<tr>
<td>GSD</td>
<td>Ground Sample Distance</td>
</tr>
<tr>
<td>INS</td>
<td>Inertial Navigation System</td>
</tr>
<tr>
<td>ISPRS</td>
<td>International Society for Photogrammetry and Remote Sensing</td>
</tr>
<tr>
<td>LiDAR</td>
<td>Light Detection and Ranging</td>
</tr>
<tr>
<td>Acronym</td>
<td>Definition</td>
</tr>
<tr>
<td>---------</td>
<td>------------</td>
</tr>
<tr>
<td>LOD</td>
<td>Level of Detail</td>
</tr>
<tr>
<td>MCD</td>
<td>Model Complexity Difference</td>
</tr>
<tr>
<td>MDL</td>
<td>Minimum Description Length</td>
</tr>
<tr>
<td>MRF</td>
<td>Markov Random Field</td>
</tr>
<tr>
<td>Ncut</td>
<td>Normalized cut</td>
</tr>
<tr>
<td>PSGC</td>
<td>Progressive Segmentation via Graph Cuts</td>
</tr>
<tr>
<td>RANSAC</td>
<td>Random Sample Consensus</td>
</tr>
<tr>
<td>ROI</td>
<td>Region of Interest</td>
</tr>
<tr>
<td>RJMCMC</td>
<td>Reversible Jump Markov Chain Monte Carlo</td>
</tr>
<tr>
<td>RMSE</td>
<td>Root Mean Square Error</td>
</tr>
<tr>
<td>RTG</td>
<td>Roof Topology Graph</td>
</tr>
<tr>
<td>SVM</td>
<td>Support Vector Machine</td>
</tr>
<tr>
<td>TLS</td>
<td>Terrestrial Laser Scanning</td>
</tr>
<tr>
<td>TPS</td>
<td>Thin Plate Spline</td>
</tr>
<tr>
<td>TIN</td>
<td>Triangular Irregular Network</td>
</tr>
<tr>
<td>VD</td>
<td>Vertex Difference</td>
</tr>
<tr>
<td>VDP</td>
<td>Vertex-driven Douglas-Peucker</td>
</tr>
</tbody>
</table>
# Table of Contents

**Chapter 1 Introduction** ................................................................. 1  
1.1 Motivation ................................................................. 3  
1.2 Challenges in 3D Building Modelling .......................... 6  
1.3 Research Objectives .................................................. 9  
1.4 Thesis Outline ......................................................... 10  

**Chapter 2 Review of Existing Approaches** ................................. 13  
2.1 Data Sources .......................................................... 15  
2.1.1 Aerial Imagery .................................................. 15  
2.1.2 LiDAR Data ....................................................... 15  
2.1.3 Ground Plans ..................................................... 16  
2.2 Urban Object Detection ............................................... 18  
2.2.1 Morphological Filtering ...................................... 18  
2.2.2 Interpolation Filtering .......................................... 19  
2.2.3 Progressive Densification Filtering ......................... 21  
2.2.4 Segment-based Filtering ...................................... 22  
2.2.5 Building Detection ............................................... 23  
2.3 3D Building Reconstruction ......................................... 26  
2.3.1 Building Footprint Extraction ............................ 26  
2.3.2 Planar Model Segmentation .................................. 27  
2.3.3 Rooftop Reconstruction ...................................... 31  
2.3.4 Building Representation ...................................... 33  
2.4 Summary of Existing Approaches .................................... 35  

**Chapter 3 Research Strategy** ....................................................... 37  
3.1 Strategy Introduction ................................................ 39  
3.2 Modelling Complexity ............................................. 39  
3.3 3D Building Reconstruction Strategy .......................... 40  

**Chapter 4 LiDAR Data Filtering** .................................................. 47  
4.1 Introduction to Filtering ............................................ 49  
4.2 Progressive Segmentation via Graph Cuts ...................... 49  
4.2.1 Outlier Removal and Initial Terrain Surface Generation 51  
4.2.2 Energy Minimization and Graph Cuts ..................... 52  
4.2.3 Definition of the Energy Function ......................... 54  
4.2.4 Segmentation of LiDAR Point Cloud with Iterated Graph Cuts 57  
4.3 Experiments ........................................................... 59  
4.3.1 Test Datasets .................................................. 59  
4.3.2 Results .......................................................... 60  
4.3.3 Criteria for Quality Assessment ............................ 64  
4.3.4 Comparison with Existing Approaches .................. 65  
4.4 Chapter Summary .................................................... 72  

**Chapter 5 Building Footprint Extraction** ...................................... 75  
5.1 Introduction to Footprint Extraction ............................ 77
5.2 Building Detection ........................................................................................................................................ 77
  5.2.1 Extraction of Straight Line Segments ....................................................................................................... 77
  5.2.2 Line-based Building Detection .................................................................................................................. 79
  5.2.3 Initial Building Boundary Determination ................................................................................................. 80
5.3 Building Footprint Extraction ....................................................................................................................... 81
  5.3.1 Boundary Simplification ............................................................................................................................ 81
  5.3.2 Optimal Polygon Selection ........................................................................................................................ 83
5.4 Hybrid Reconstruction ..................................................................................................................................... 84
  5.4.1 Explicit Reconstruction .............................................................................................................................. 84
  5.4.2 Implicit Reconstruction ............................................................................................................................. 86
5.5 Experiments ...................................................................................................................................................... 88
  5.5.1 Criteria for Quality Assessment .............................................................................................................. 88
  5.5.2 Performance on Simulated Data .............................................................................................................. 89
  5.5.3 Performance on Real Data ...................................................................................................................... 92
  5.5.4 Discussion .............................................................................................................................................. 95
5.6 Chapter Summary ......................................................................................................................................... 98

Chapter 6 Building Rooftop Modelling ........................................................................................................... 99
6.1 Introduction to Roof Modelling ..................................................................................................................... 101
6.2 Spectral Clustering ....................................................................................................................................... 101
  6.2.1 Principle of Spectral Clustering ................................................................................................................ 101
  6.2.2 $K$-way Partitioning of Spectral Clustering .............................................................................................. 104
6.3 Roof Plane Segmentation .............................................................................................................................. 105
  6.3.1 Local Best-fit Plane Retrieval .................................................................................................................. 106
  6.3.2 Graph Laplacian Construction ................................................................................................................ 107
  6.3.3 Spectral Subspace Clustering ................................................................................................................ 108
6.4 Building Rooftop Reconstruction ................................................................................................................ 110
  6.4.1 Extraction of Roof Ridge and Internal Vertices ....................................................................................... 110
  6.4.2 Completing the Boundary using the Footprint ......................................................................................... 112
6.5 Experiments .................................................................................................................................................... 116
  6.5.1 Test Datasets ....................................................................................................................................... 116
  6.5.2 Results ................................................................................................................................................. 116
  6.5.3 Criteria for Quality Assessment ............................................................................................................ 119
  6.5.4 Evaluation .......................................................................................................................................... 121
6.6 Chapter Summary ....................................................................................................................................... 123

Chapter 7 Conclusion and Future Work ......................................................................................................... 125
7.1 Conclusion ..................................................................................................................................................... 127
7.2 Contributions ............................................................................................................................................... 129
7.3 Future Work ............................................................................................................................................... 130

Bibliography ..................................................................................................................................................... 133

List of Publications ............................................................................................................................................ 145

Curriculum Vitae .............................................................................................................................................. 147
List of Figures

Figure 1.1 High-resolution virtual city scene of Melbourne CBD, Australia, generated from aero3Dpro. ................................................................. 4

Figure 1.2 Data representation of the Engineering building at the University of Melbourne: (a) aerial imagery and (b) airborne LiDAR point cloud................................................................. 6

Figure 1.3 Scene appearance challenge - three landscapes with different urban landscapes: (a) a flat scene in a residential area of Germany, (b) a hilly scene in the city of Hong Kong and (c) a vegetation scene with sparse buildings in Australia. .............................. 7

Figure 1.4 Model quality assessments - two building reconstruction results regarding geometry and topology quality, respectively: a data-driven result with most data fitting in terms of geometry (left) and the alignment of a building model based on regularity considerations (right). ......................................................... 8

Figure 3.1 The five levels of detail defined by CityGML (Kolbe et al., 2005). ................................................................. 40

Figure 3.2 General strategy for 3D building modelling................................................................. 43

Figure 3.3 Input data, processing methods and achievements in the proposed building modelling system................................................................. 45

Figure 4.1 Proposed scheme of Progressive Segmentation via Graph Cuts (PSGC). ................................................................. 50

Figure 4.2 Example of graphical model................................................................. 54

Figure 4.3 Weights of points linking with the preliminary terrain surface........... 55

Figure 4.4 Weights of pair-wise linking................................................................. 56

Figure 4.5 Filtering result of Unimelb: (a) aerial image, (b) shaded TIN of DSM, (c) initial ground points, (d) iterative segmentation, (e) classification result and (f) shaded TIN of DTM............................................. 61

Figure 4.6 Filtering result of Eltham: (a) aerial image, (b) shaded TIN of DSM, (c) initial ground points, (d) iterative segmentation, (e) classification result and (f) shaded TIN of DTM............................................. 62

Figure 4.7 Iterative segmentation results................................................................. 63

Figure 4.8 Filtering results in detailed................................................................. 64

Figure 4.9 Reference data and filtering errors of Sample11: (a) aerial imagery, (b) generated DTM, (c) – (f) error distribution of PSGC, ALDPAT, terrascan and upward-fusion, respectively (Type I in green and Type II in red)................................................................. 66

Figure 4.10 Reference data and filtering errors of Sample12: (a) aerial imagery, (b) generated DTM, (c) – (f) error distribution of
PSGC, ALDPAT, terrascan and upward-fusion, respectively (Type I in green and Type II in red). .......................................................... 66

Figure 4.11 Reference data and filtering errors of Sample21: (a) aerial imagery, (b) generated DTM, (c) – (f) error distribution of PSGC, ALDPAT, terrascan and upward-fusion, respectively (Type I in green and Type II in red). .......................................................... 67

Figure 4.12 Reference data and filtering errors of Sample22: (a) aerial imagery, (b) generated DTM, (c) – (f) error distribution of PSGC, ALDPAT, terrascan and upward-fusion, respectively (Type I in green and Type II in red). .......................................................... 67

Figure 4.13 Reference data and filtering errors of Sample23: (a) aerial imagery, (b) generated DTM, (c) – (f) error distribution of PSGC, ALDPAT, terrascan and upward-fusion, respectively (Type I in green and Type II in red). .......................................................... 68

Figure 4.14 Comparison of errors and kappa coefficients for the four approaches: (a) type I errors of different approaches for the samples, (b) type II errors of different approaches for the samples, (c) total errors of different approaches for the samples and (d) kappa coefficients of different approaches for the samples. ............................................................................. 70

Figure 4.15 Error distribution of generated DTMs: (a) distribution of ground control points, (b) accuracy assessment from PSGC, (c) accuracy assessment from ALDPAT, (d) accuracy assessment from Terrascan, (e) accuracy assessment from Upward-fusion. ............................................................................. 71

Figure 5.1 A demonstration of the line extraction algorithm: (a) input LiDAR point cloud, (b) the separated scan lines, (c) extracted line segments are rendered in different colour and (d) close-up of a raw scan line and segmentation result. ............................................................................. 78

Figure 5.2 Results for building detection: result from point-based classification (left), and result from line-based classification (right). ............................................................................. 80

Figure 5.3 Extracted building patches (left) and initial boundaries (right). ..................... 81

Figure 5.4 Polygonal hypotheses with different numbers of vertices. ............................. 82

Figure 5.5 Energies for different vertex numbers. ........................................................... 84

Figure 5.6 Explicit reconstruction of a building footprint: (a) local fitting and (b) regularization of remaining line segments. ........................................................... 86

Figure 5.7 Connector hypotheses: (a) no additional line, (b) one additional line and (c) two additional lines. ........................................................... 87

Figure 5.8 Watertight building footprint. ............................................................................. 88

Figure 5.9 Comparison of performance under various Gaussian noise levels. .......... 90

Figure 5.10 Building footprint extraction applied to 17 buildings in the Eltham dataset: (a) detected building regions, (b) extracted footprints and (c) ground truth. ............................................................................. 93
Figure 5.11 Experiments and quality assessments on several building footprints; DSM on the left, aerial image on the right. ................................95

Figure 5.12 Difference between raw boundary and ground truth. .................................96

Figure 5.13 Ortho image, extracted footprint, and extraction error from improper lambda value A2. .................................................................96

Figure 5.14 Footprint reconstruction of complex building shapes with high density LiDAR data .............................................................97

Figure 6.1 Flow chart of spectral clustering .................................................................106

Figure 6.2 Illustration of spectral clustering process: (a) raw point cloud, (b) extracted line segments, (c) constructed adjacency matrix, (d) ideal adjacency matrix and (e) line-based clustering result. .................................109

Figure 6.3 Initial boundary of each planar patch .........................................................110

Figure 6.4 Determination of roof ridge and inner vertices: (a) planar patches and intersection ridge, (b) roof topology graph, (c) four closed cycles and (d) corresponding internal vertices and snapped result. ..112

Figure 6.5 Direction regularization: (a) before adjustment and (b) after adjustment .................................................................114

Figure 6.6 Snapping of vertices: (a) before adjustment and (b) after adjustment. ..............................................................................114

Figure 6.7 Consistency of vertices consistency: (a) before adjustment and (b) after adjustment. ..............................................................................115

Figure 6.8 Modelling dormer structures: (a) combination of footprint and internal structures, (b) Determination of 3D bounding box for substructure and (c) building model with dormers. ..................................115

Figure 6.9 Experimental results from ISPRS test data. The first column shows the extracted plane structures overlaid on the true orthoimage. The second column illustrates the reconstruction results and the last column shows the true orthoimages. .................117

Figure 6.10 Highlighted building models. The first column shows clustering results. The second column illustrates the reconstruction results and the last column shows referenced roof patches on the true orthoimages. ..............................................................................118

Figure 6.11 Extraction errors: (a) absence of data; (b) low vegetation is detected as a building; (c) low building is detected as terrain; (d) small planar patches fail to be extracted. .................................................................119

Figure 6.12 Topological differences: reference planes versus extracted planes in the three test areas . .................................................................122
List of Tables

Table 2.1 Summary of data sources and data formats in the reviewed methods................................................................. 17
Table 2.2 Summary of urban object detection methods................................................................. 25
Table 2.3 Summary of reconstruction methods. ................................................................. 34
Table 4.1 Characteristics of two test sites......................................................................................... 59
Table 4.2 Summary of filtering results. .......................................................................................... 60
Table 4.3 Comparison of filtering errors and kappa coefficients of PSGC, ALDPAT, Terrascan, and Upward-fusion for all sample sites............. 69
Table 4.4 Error statistics from generated DTMs of PSGC, ALDPAT, Terrascan and upward-fusion. ........................................................................ 72
Table 5.1 Statistical assessment under various Gaussian noise levels............................. 91
Table 5.2 Statistical assessment of building footprint extraction. .................................. 94
Table 6.1 Summary of the test sites. ................................................................................................. 116
Table 6.2 Statistical evaluation of the building reconstruction results. ................. 121
Chapter 1 Introduction
1.1 Motivation

Formerly, Geospatial Information Systems (GIS) contained only 2D datasets in the form of cadastral plans or contour maps. Recently, 3D landscape models that conform to reality have become more important. A good example is that a large amount of residential houses and commercial buildings have been constructed since the resident population had increased nearly 10% in the period from 2006 to 2011 in the city of greater Melbourne, Australia. Driven by the rapid urbanization, a real need for frequently refreshed 3D GIS is emerging in response to the problem of efficient city management (Landa et al., 2013). 3D building models are acknowledged as the essential part in such systems.

3D building models are the computer-based representation of city landscapes. They deliver spatial and geometric descriptions in terms of location, pattern, volume and shape. They also provide virtual reality in such a way as to allow simulation applications in a variety of fields involving the city form. For disaster control, simulating flood physics is much more efficient than imaging the flow information on elevation contour maps. For public safety, precise 3D building models are often employed for emergency response. Shiode (2000) grouped these applications into four categories: (1) planning and design, (2) infrastructure and facility services, (3) commercial sector and marketing, and (4) promotion and learning of information on cities. Satisfying all these applications requires accurate, concise and up-to-date building models (Zhou and Neumann, 2008).

According to an early survey on 3D city modelling conducted by the European Organization for Experimental Photogrammetric Research (Fuchs et al., 1998), over 90% of the respondents expressed their interest in realistic 3D city models consisting of information on building, traffic network, and vegetation. To cope with higher application demands, 3D visualization projects are carried out by governments, especially in cities with a population of greater than one million people, such as Tokyo, New York, Los Angeles, Chicago, Mexico, Berlin and Paris (Batty et al., 2001). To date, the requirement for such models is also apparent amongst private clients. Many prominent companies such as Google, Apple and Microsoft are now heavily involved in generating free access and interactive platforms for world-scale 3D landscapes (Leberl, 2010).
Therefore, modern map production within the last two decades has undergone a paradigm shift from 2D topographic mapping to 3D landscape modelling (Haala and Kada, 2010). One of the most routine tasks is the procedure of converting a city landscape into building models. Traditionally, the most direct route to obtain such models relied on manual land surveying. Every main feature is obtained by survey devices, such as total station or GPS receiver. However, the performance greatly depends on the surveyor’s knowledge of landscape. Due to the significant heterogeneity in appearance and complexity in landscapes, manual delineation of the entire city scene is a rather labour-intensive and time-consuming task, and it is thus unsuitable for large-scale modelling. Since the 1990s, automating this process has drastically minimized human resources and fulfilled up-to-date supply. Remote sensing datasets have great potential to provide a more reliable representation of city scenes. Building model generation is usually performed in two phases: (1) sensor data acquisition; (2) building detection and reconstruction.

![Figure 1.1 High-resolution virtual city scene of Melbourne CBD, Australia, generated from aero3Dpro.](image)

Many advances in the fields of sensor hardware and acquisition techniques have made it possible to collect precise geometric information and to have the capability of representing detailed 3D city environments. For instance, Figure 1.1 illustrates a
A powerful representation of Melbourne CBD in Australia from remote sensing data. The fact of employing sensor data also owes much to the nowadays more accessible datasets at affordable prices.

To date, two 3D sensor systems have been distinguished for building modelling:

- **Camera systems** detect radiation from an external source of energy, such as sunlight. The camera systems recover the local depth from a sequence of 2D images. The datasets, such as images or videos, have been served as the traditional source for 3D measurement.
- **Light Detection And Ranging (LiDAR)** systems emit laser pulses towards an object surface and subsequently measure time-of-flight between sensor and target to obtain range measurement. LiDAR is an emerging 3D sampling technology which opens up an extensive range of applications for 3D reconstruction.

In comparison to image analysis, LiDAR has proven to be a well-established technique in the collection of spatial information for at least four related reasons. First of all, LiDAR is an active remote sensing technology, as the system uses an on-board laser sensor to transmit the active source of energy, which allows data acquisition at any time of the day or night independently of natural lighting conditions. Second, laser pulses can partially penetrate vegetation, which makes data underneath the trees cover available. Third, LiDAR directly produces georeferenced information through the integration of the *Global Positioning System* (GPS) and *Inertial Navigation System* (INS) sensors. While imagery records the spectral properties of the landscape, the LiDAR system scans the underlying sampled surfaces, resulting in discrete and dense arrays of 3D coordinates, also known as point clouds. Figure 1.2 shows the Engineering building at the University of Melbourne with imagery and point clouds representations. Lastly, besides the range information, modern laser scanners support the recording of pulse information, such as amplitude and return count, to enrich coordinate attributes. Overall, the LiDAR system has become a state-of-the-art tool which enables the generation of highly precise, high resolution, and large-scale elevation information (Baltsavias, 1999). Such advances have offered researchers much potential to address a variety of issues ranging from detailed
terrain model extraction and forestry canopy management to heritage applications (Vosselman and Maas, 2010).

![Image](image_url)

(a) Imagery

(b) LiDAR point cloud

Figure 1.2 Data representation of the Engineering building at the University of Melbourne: (a) aerial imagery and (b) airborne LiDAR point cloud.

Although acquisition of dense 3D point cloud data has become routine, the “detection and reconstruction” of building models from such digital surface models is still an active research field (Schnabel et al., 2008). This thesis details research into the automated extraction and advanced object modelling from LiDAR point clouds.

1.2 Challenges in 3D Building Modelling

The main difficulty of building detection and reconstruction lies in the fact that raw LiDAR data provides only Cartesian coordinate measurements without any semantic
or structural information of the elements it contains, i.e. which parts of the data belong to which entities, and which parts are from which geometric shapes. In this section, several challenging issues in building modelling using airborne LiDAR data are highlighted.

Although many efforts and different strategies have been developed over the last two decades, showing various degrees of success (Brenner, 2005; Grün et al., 1995; Haala and Kada, 2010), existing approaches for the automation of building modelling from airborne LiDAR data usually fail to provide comprehensive results and they fall short of practical requirements, mostly due to the intrinsic characteristics of point clouds (i.e., discrete and irregular distribution) and the very large data size in a large-scale scene. Moreover, the scene heterogeneity in appearance, the unavoidable noise due to the environment, the varieties of structures, and the indefinite number of possible structures make the robust modelling task extremely challenging.

- Complex appearance is critical for urban object detection. For instance, Figure 1.3 shows three LiDAR datasets representing typical landscapes in Europe, Asia, and Australia, respectively. Architectural style comes with a diversity of shapes and components. Humans perceiving a city scene can often easily recognize various terrain trends and building objects even though ambiguous appearances in all data are quite different. Computer vision must be capable of achieving a similar level of performance in complex environments.

Figure 1.3 Scene appearance challenge - three landscapes with different urban landscapes: (a) a flat scene in a residential area of Germany, (b) a hilly scene in the city of Hong Kong and (c) a vegetation scene with sparse buildings in Australia.

- Huge size of data is another major difficulty. The amount of acquired LiDAR data is massive, partially because of a high sampling rate for the description of
detailed structures. To cover an entire city scene, overlapped strips from multiple scan lines will be required. This generates huge point cloud for processing. The handling of unorganized point clouds is often computationally expensive because a great number of elements must be accommodated. This is especially true when working with datasets that are larger than the PC’s physical memory. An out-of-core segmentation step that compacts raw data into meaningful objects is important to reduce the number of processing elements.

- **Limited resolution of LiDAR data** is also a challenge for the feature extraction and reconstruction task. For instance, important information such as the number of building corners and their locations are often unknown. An extraction method is required to extract every corner from LiDAR data to complete the building footprint. If one or more corners are missing, then the obtained footprint will have unreliable geometry.

- **Model quality assessment** within the model is also a critical issue. Human vision is more sensitive to topological regularity. For instance, Figure 1.4 shows a prismatic building model and its regularized form. Although the left model best fits the LiDAR data in terms of geometry, the missing parallel and perpendicular constraints on building outlines are conspicuous to human vision. Herein, the reconstructed model requires not only geometric fitness to the observation data, but also the need to preserve necessary regularity.

![Figure 1.4 Model quality assessments - two building reconstruction results regarding geometry and topology quality, respectively: a data-driven result with most data fitting in terms of geometry (left) and the alignment of a building model based on regularity considerations (right).](image)
1.3 Research Objectives

The presented study has been achieved within the framework of an urban feature extraction project. The project is a cooperation of the University of Melbourne, Cooperative Research Centres for Spatial Information (CRCSI) and 43pl members. The aim is to metrically combine the imagery and LiDAR point clouds to form accurate 3D images. The combined 3D images provide additional information through each of the data sources. These 3D images will be used to automatically extract features for modelling objects such as buildings and vegetation parameters.

This thesis deals with building extraction in the project CRCSI 2.02. The main task is to develop and evaluate a building extraction system which includes a set of robust and efficient methods to process the sensor datasets. This work centres on employing airborne LiDAR point clouds because LiDAR datasets are available for large-scale production purposes. The results of this system will be concise 3D vector models describing the geometric structures of buildings as precisely as possible. The results can be used for end users who can obtain value-added information from 3D datasets.

To address problems in this specific task requires solving a series of processing steps spanning many techniques, such as object recognition, feature extraction and regularity. A wide range of properties is also required to better understand the framework system. Of these, specific objectives should be borne in mind for the proposed system:

1. **Scene interpretation**: The system should be able to detect three dominant urban objects from LiDAR data, namely, terrain, buildings and vegetation. Scene interpretation is crucial to obtain good accuracy of the derived models.

2. **Robustness**: The system should be able to perform on various LiDAR datasets regardless of data resolution and noise content.

3. **Efficiency**: The system should deal with large data sets to reconstruct city-scale building models within reasonable time and computing resources.

4. **Accuracy**: The system should produce precise building models from the input data, which are as accurate as possible. The following requirements should also be addressed in regard to accuracy and geometric completeness:
i. The reconstructed building models should contain a set of simple roof facets in an explicit manner, to enable applications such as solar collection analysis.

ii. Building models should adequately represent the raw data.

iii. Building roofs should be assembled in a reasonable manner to preserve design and fabrication considerations.

5. **Automation**: The developed building modelling process should be designed as an automated operation.

### 1.4 Thesis Outline

This thesis is organized into seven main chapters. After this introduction, the thesis is structured as follows:

- **Chapter 2**: *Review of existing approaches* provides an overview in the scope of 3D building reconstruction from imagery and LiDAR data. The chapter provides a description of different current techniques. Readers can catch up on the state-of-the-art methods and details from references. Furthermore, the chapter offers a comparison of current approaches with the system to be developed.

- **Chapter 3**: *Research strategy* explains the basic notation in term of modelling complexity and energy minimization. Following that, the proposed strategy and system achievements are outlined.

- **Chapter 4**: *LiDAR data filtering* presents a novel graph-based approach for filtering LiDAR data. The LiDAR data is used to separate terrain and non-terrain points to support building detection. It is achieved by finding the minimal energy cuts over a graphical model. The graphical model encodes both soft point-wise closeness constraints as well as pair-wise smoothness constraints to alleviate filtering errors. A primary focus here is on the adoption of an iterative process to gradually refine the terrain surface, which propagates the filtering result to optimize the next graph cuts.

- **Chapter 5**: *Building footprint extraction* deals with the generation of building outlines. This chapter consists of three major parts: the detection of buildings is explained in Section 5.2. This includes line segment extraction and line-based
building detection. Line segments are used as the basic element to reduce the analysis volume. Next, the initial building boundary is employed to find the approximate polygon which contains only important vertices. This is detailed in Section 5.3. In Section 5.4, there is a description of the use of edges to enforce regularity constraints in terms of parallelism and perpendicularity.

- **Chapter 6**: *Building rooftop modelling* deals with the extraction and reconstruction of roof structures using line segments. Finding planar roof patches is a crucial component of the proposed system. The extraction of roof planes is explained in Section 6.1. This includes local plane retrieval, graph laplacian construction and spectral subspace clustering. The developed spectral clustering technique provides a new energy-preserving representation of data, which outperforms traditional clustering algorithms. Lastly, the rooftop model is reconstructed via the plane boundaries. This is realised with the aid of planar patch intersection and the extracted footprint, as much as possible, to provide reliable and consistent models.

- **Chapter 7**: *Conclusion and further work* summarizes the contributions of the research and suggests possible directions for future work.
Chapter 2 Review of Existing Approaches
Chapter 2 Review of Existing Approaches
In this chapter, the significant and recent building reconstruction methods will be reviewed in order to familiarize the reader with the employed datasets, and the proposed approaches and their underlying principles. The review starts with a description of the data sources employed. A detailed literature review related to each urban object detection and building reconstruction step is then provided in Sections 2.2 and 2.3.

### 2.1 Data Sources

#### 2.1.1 Aerial Imagery

As mentioned in Section 1.1, aerial imagery has traditionally been and still is the major data source for building detection. In the early stages, spectral information from a single intensity image was employed for buildings detection (Lin and Nevatia, 1998). Later, when two or more overlapping images were utilised, a number of common features could be detected on each image and then a depth map could be generated based on stereo triangulation (Davidson, 1985). Compared to the single images, 3D data from stereo image analysis provides more geometric information. As a highly cited approach, Weidner and Förstner (1995) derived height models, which will also be referred to here as height image, using Match-T software, and they identified terrain and building regions by employing morphological operations. Recent methods from Gerke and Xiao (2013) employed multiple aerial oblique images to obtain a dense voxel representation. This follows a scene interpretation based on the geometric, textural and colour information of the voxels. A comprehensive study and comparison of automated building detection methods can be found in Khoshelham et al. (2010).

#### 2.1.2 LiDAR Data

With the introduction of LiDAR datasets, the detailed reconstruction of complex building shapes became feasible. A comparison of the accuracy and robustness of extraction processing using LiDAR data and stereo photogrammetry has been reported in Baltsavias (1999). From contour lines derived from the two data sources, the author suggested that the quality of LiDAR data is much higher, especially on object edges.
Pioneering extraction approaches (Alharthy and Bethel, 2002; Rottensteiner and Briese, 2003) commenced by obtaining a height image with the help of interpolation algorithms such that traditional image toolboxes for feature extraction could be applied. Due to the fact that rasterized images drop LiDAR pulse information and reduce spatial precision (Axelsson, 1999), it is preferable to apply methods at the sample point level so that end-users can process or manually edit the raw data to fit their particular requirements.

Triangulation meshes generated from point clouds are also a popular input format for the processing of LiDAR datasets (Tse et al., 2005). Meshes consist of a set of triangles formed by connecting of their common corners or edges to provide connectivity information. Nevertheless, geometric features from spurious triangles are often unreliable due to the presence of noise (Maas and Vosselman, 1999). Besides, triangulation meshes are derived from point clouds, so processing directly on point cloud is preferable.

Airborne LiDAR point clouds with nadir viewing are widely used for city-scale modelling (Haala and Kada, 2010). Airborne point clouds describe visible surfaces, such as building rooftops, tree crowns, and terrain structure, while information on vertical structures, such as building façade or tree branch, is generally too sparse for adequate representation. Therefore, the derived models are more commonly adopted for applications where a bird’s eye view of the building model is desirable. A comparison of vertical façade extraction from airborne and terrestrial LiDAR point clouds, which reported that less details and accuracy can be extracted from airborne data, has been presented in Rutzinger et al. (2009). To complete the building façade information, airborne LiDAR data is usually supplemented by terrestrial data as LiDAR can have considerable variations in scan view and sampling rates. Fruh and Zakhor (2003) have employed terrestrial LiDAR data to fill the data gap in LiDAR coverage. In recent years, oblique-view airborne LiDAR data has also provided sufficient information for modelling vertical structures in open areas (Tuttas and Stilla, 2013).

### 2.1.3 Ground Plans

Some researchers exploit the integrated information of remote sensing data and ground plans. Ground plans in existing topographic databases are often available as cadastral data. They invariably contain 2D representations of building footprints,
which play an important role in building detection and structure decomposition, and they provide initial values for primitive hypotheses (Vosselman, 2002). In previous work (Brenner et al., 2001; Vosselman and Dijkman, 2001; Vosselman and Suveg, 2001), the employment of existing 2D ground plans produced better success rates in feature hypothesis, especially in areas with small roof surfaces and flat roofs because of the uncertainty of the roof structure derived from LiDAR and imagery-based point clouds.

The data source and data format used in the reviewed literature are listed in Table 2.1. Since the early 2000s, airborne LiDAR data has been regularly adopted. High accuracy and high resolution is the main reasons why LiDAR point clouds are often preferred over height images from photogrammetry. Moreover, building reconstruction is more likely to be performed on LiDAR data alone, without the assistance of ground plans, which are mostly accessible in cities, and in general are updated infrequently.

Table 2.1 Summary of data sources and data formats in the reviewed methods.

<table>
<thead>
<tr>
<th>Authors</th>
<th>Data source</th>
<th>Data format</th>
</tr>
</thead>
<tbody>
<tr>
<td>Weidner and Förstner (1995)</td>
<td>Aerial imagery</td>
<td>Height image</td>
</tr>
<tr>
<td>Maas and Vosselman (1999)</td>
<td>Airborne LiDAR</td>
<td>Point cloud</td>
</tr>
<tr>
<td>Brenner et al. (2001)</td>
<td>Airborne LiDAR, ground plan</td>
<td>Point cloud, vector map</td>
</tr>
<tr>
<td>Vosselman and Dijkman (2001)</td>
<td>Airborne LiDAR, ground plan</td>
<td>Point cloud, vector map</td>
</tr>
<tr>
<td>Vosselman and Suveg (2001)</td>
<td>Aerial imagery, ground plan</td>
<td>Colour image, vector map</td>
</tr>
<tr>
<td>Alharthy and Bethel (2002)</td>
<td>Airborne LiDAR</td>
<td>Height image</td>
</tr>
<tr>
<td>Rottensteiner and Briese (2003)</td>
<td>Airborne LiDAR</td>
<td>Height image</td>
</tr>
<tr>
<td>Tse et al. (2005)</td>
<td>Airborne LiDAR</td>
<td>Triangulation mesh</td>
</tr>
<tr>
<td>Rutzinger et al. (2009)</td>
<td>Airborne LiDAR, terrestrial LiDAR</td>
<td>Point cloud</td>
</tr>
<tr>
<td>Fruh and Zakhor (2003)</td>
<td>Airborne LiDAR, terrestrial LiDAR</td>
<td>Point cloud</td>
</tr>
<tr>
<td>Tuttas and Stilla (2013)</td>
<td>Airborne oblique LiDAR</td>
<td>Point cloud</td>
</tr>
<tr>
<td>Gerke and Xiao (2013)</td>
<td>Aerial oblique imagery</td>
<td>Voxel</td>
</tr>
</tbody>
</table>
2.2 Urban Object Detection

No matter which specific dataset is employed, most of modelling methods start with object detection to identify which part of the data belongs to which entity. Terrain (ground surface), buildings and vegetation are the three urban objects of most interest in geospatial applications.

Building detection is usually considered as a two-step binary classification: 1) separation of terrain and non-terrain objects; 2) post-processing of non-terrain objects to detect buildings. Particularly, the first binary classification is also known as LiDAR filtering for Digital Terrain Model (DTM) extraction. According to the filtering concept of Vosselman and Maas (2010), four filtering classes can be categorized: morphological filtering, surface interpolation filtering, progressive densification filtering and segment-based filtering.

2.2.1 Morphological Filtering

In the field of image processing, mathematical morphology is used to describe the spatial structure based on a set of morphological operators. A morphological operator changes an image into regions of homogeneity, where structures of interest are emphasized. Two basic operators are designed in terms of erosion and dilation, where erosion shrinks a region while dilation expands the region. In addition, the two operations are combined together to define two more morphological operations, the so-called opening and closing operators. The opening operation is achieved by applying an erosion of the dataset followed by dilation, while the closing operation is carried out in the reverse order. These kernel elements can also be employed to extract terrain regions from LiDAR data. To start the morphology processing, potential regions and an operator are essential. Lindenberger (1993) is the pioneer of using morphology operators on point clouds. The seed regions are first detected by the lowest (minimum elevation) points within each structure window moving along a profile. Then the terrain regions are extended to neighbouring points if the elevation differences are small. The neighbourhood can be defined either along LiDAR profiles (Shan and Sampath, 2005), in a multi-directional fashion in rasterized height image (Meng et al., 2009), or through neighbouring points within a certain distance (Vosselman, 2000).
One critical parameter for morphological filters is the selection of the window size to ensure that the lowest points are always on the terrain surface. If a large man-made homogenous structure, such as a train station covers more than the window size, then the initial derived ground region may mislead subsequent procedures. Therefore, various object regions require adaptive window sizes to identify ground regions. Zhang and Chen (2003) proposed a progressive morphological filter by gradually enlarging the window size. With iteratively increasing window size, the height difference is calculated between surfaces after morphological opening. Points within a certain height difference are identified from the terrain surface. Chen et al. (2007) extended adaptive morphological filtering by first performing an opening operation to remove tree points via a relatively small window. Then, building points are filtered with a larger window in a similar way to the Zhang and Chen (2003) approach. This method is adaptive to local terrain and can readily work over rugged areas.

Another critical parameter to be defined is the maximum allowable height difference between a seed terrain point and its neighbours. This parameter describes the slope of the terrain. Rather than using empirical knowledge about the terrain shape, Vosselman (2000) determined the allowable height difference from statistical analysis of a selected training data. However, this fixed function representing the terrain slope is not always realistic in complex landscapes since points on steep terrain are likely to be filtered out. In order to preserve local terrain characteristics, Sithole (2001) proposed an adaptive slope function by using a coarse DTM for estimation of local terrain slope, where the coarse DTM is derived from the lowest points within a certain cell. A similar approach has been proposed by Kim and Shan (2011a), where the discontinuity measurement is achieved by using the residuals to the line connecting the two consecutive points. The discontinuity threshold is determined adaptively as 50% of all residuals of the current profile. If the discontinuity is larger than the threshold, the corresponding point is assigned as a non-terrain point and is removed from the next iteration.

### 2.2.2 Interpolation Filtering

Terrain regions can also be extracted via surface interpolation based on the assumption that terrain is represented as a continuous surface. These algorithms start with setting whole points as terrain points and iteratively interpolating the surface
based on a weight function. The surface is confirmed when point weights are stable and then points close to the surface are treated as terrain points. Therefore, this method performs surface interpolation and filtering in the same process.

Kraus and Pfeifer (1998) proposed a weighted interpolation algorithm to approximate the terrain surface. All points are firstly assigned with the same weight to interpolate the initial surface. Based on the vertical residual to the surface, each point is reassigned with a different weight. Obviously points below the surface are more likely to be terrain points, and hence have more influence (high weight). Points above the surface are more likely to be non-terrain points, so they have less influence on the run of the surface (low weight). A new surface is interpolated based on the calculated weight of each point. After the interpolation, the weight of every point is recalculated for the next iteration.

Elmqvist et al. (2001) adopted the active shape models to determine terrain surface. In a similar way to active contours for finding continuous edges in images, continuous shape models can also be determined in point clouds of low elevation. The objective function minimizes the energy by summing up internal energy, potential energy and constraint energy to direct the membrane shape. The internal energy is a function of the smoothness of the surface; the potential energy is the potential field created from the DSM and the constraint energy is defined as a force constraint model to a preferred surface. The energy function guides the surface towards the low points and restricts the defined surface. The optimal shape is achieved in an iterative manner and any points within a buffer of the final surface are assigned as terrain points.

Many interpolation-based filters suffer from difficulties in high computation cost when processing large-scale LiDAR data. To speed up the filtering process, Pfeifer et al. (2001) embedded the weighted interpolation method in a hierarchical approach. They first filtered the lower resolution data to obtain a rough DTM. Then the derived surface was used to compare the data of higher resolution. Chen et al. (2013) also presented a multi-resolution hierarchical method for filtering. At each level, the surface was iteratively interpolated towards the ground using a Thin Plate Spline (TPS) approach and the residuals between higher resolution points and the surface were calculated. If the residual was larger than a pre-defined threshold, the point was excluded from terrain surface.
2.2.3 Progressive Densification Filtering

As with interpolation-based filters, progressive densification filters also iteratively update the surface model by considering height residual. However, rather than treating the terrain as a polynomial surface, progressive densification filters employ a triangulation mesh to approximate the surface as a polyhedron. Furthermore, progressive densification filters proceed in a coarse-to-fine manner. Particularly, a small set of lowest points are first selected to obtain the sparse and preliminary terrain, and then more and more points are classified into terrain points to refine the derived surface.

Axelsson (2000) first divided the whole point dataset into tiles, where the lowest point in each tile is selected as the seed terrain point. The preliminary terrain surface is obtained as the Triangular Irregular Network (TIN) of those lowest points. For each triangle, one of the remaining points, which inside the triangle has a small height offset, will be assigned as a terrain point. Before continuing the next iteration, the TIN is updated by adding the new terrain points. To avoid an edge cutting effect, mirror points are introduced to retain qualified edge points. The accuracy of this method, however, strongly depends on the initial selection of seed points. In undulating terrain, the selected points need to be dense enough to approximate the bare earth. On the other hand, in urban area, the dense initial terrain points may be incorrectly selected, which results in many non-terrain points being classified as terrain points.

Sohn and Dowman (2002) proposed a two-step densification approach. Firstly, the coarse terrain surface is established by the lowest points in the four corners of the whole area. In the following downward step, the TIN is updated by adding the lowest point within each triangle. This step is repeated until no point below the TIN can be added. In the final upward step, the TIN is further refined by identifying ground points above the current TIN. Within this step they suggest adding in each triangle those points which are in a certain buffer with respect to the surface defined be the triangle and which fulfil a certain Minimum Description Length (MDL) criterion. All points in the TIN are classified as terrain points and others are classified as non-terrain points.

In order to robustly extract an accurate terrain model in the presence of objects such as large buildings, Chen et al. (2012) proposed an upward-fusion method. They
first generated a set of preliminary DTMs of different resolutions using a local-
minimum method. Then, upward fusion was conducted between successive DTMs.
This process began with the DTM of the lowest resolution, which was treated as a
trend surface. A finer DTM was compared with this large-scale DTM. If the points in
the finer DTM have a smaller height difference than the trend surface, those points
are added to trend surface. This process repeats until all preliminary DTMs had been
included in the upward fusion and a refined DTM of high resolution is achieved.

2.2.4 Segment-based Filtering

The bare earth can be assumed to consist of piecewise continuous regions
decomposed by objects. Segment-based filters attempt to identify terrain regions so
that a segment rather than a point is the basic element for filtering. As a pre-
processing step, segmentation means aggregating points into several regions such
that each region is homogeneous without overlap. Points in each segment share
similar properties and thus have the same label. In the domain of terrain filtering,
analysing segments rather than individual points assures that a point and its
neighbouring points belonging to the same region are labelled consistently. As a
matter of fact, segmentation can provide more reliable results since both geometric
and topological information is considered in the filtering procedure (Vosselman,
2009).

Sithole and Vosselman (2005) proposed one representative approach for the
filtering of airborne laser scanning data based on segmented point clouds. The point
clouds are firstly partitioned into a series of various orientated profiles and points
within each profile are further segmented when height difference criteria are
fulfilled. Subsequently, the segments of the whole point clouds are derived by
grouping line segments from different profiles. The subsequent classification of the
segments is based on an analysis of the neighbourhood relation of the segments. One
remarkable advantage of profile analysis is that ambiguous features inside terrain
segments, such as a bridge, can be easily detected.

Zhou and Neumann (2008) presented another segment-based approach for filtering
airborne LiDAR point clouds. They proposed a segmentation of the point cloud
based on region growing. Starting from one single point, points with similar
characteristics are grouping to the segment in a step-by-step process. The LiDAR point clouds are divided into different patches, where the largest segment is treated as terrain. To improve the initial result, a post-processing algorithm which merges low-height patches into terrain is employed.

To automatically extract a DTM from voluminous LiDAR data, He (2010) presented a splitting and merging strategy for filtering segments. First of all, the whole dataset is split into several median-size overlapping tiles. For each tile, accurate segments are extracted. In the following step, segment distribution and the height jump of adjacent segments are the two criteria to identify terrain segments. At the last step, segments on the edge of tiles are located and reclassified to reduce the edge cutting effect.

2.2.5 Building Detection

The identified non-terrain points naturally include returns from building roofs as well as vegetation. Building detection is a binary classification task to separate points on building roofs from vegetation. Many efforts have been addressed to this task. In general, methods extract geometry and reflectance attributes within the data and then employ a classification algorithm to distinguish buildings and vegetation.

Early research work directly employed point-based information to calculate the discriminative features at each point, such as point distribution (Zhou and Neumann, 2008), height residual (Lafarge and Mallet, 2012), flatness (Verma et al., 2006), first-last pulse difference (Alharthy and Bethel, 2002) and intensity analysis (Song et al., 2002). The classification algorithms then utilized the derived features to assign a class label to the corresponding point. The detection algorithm can be a rule-based decision tree or a machine learning classifier, such as random forests (Chehata et al., 2009), Support Vector Machine (SVM) (Zhan and Yu, 2011), or Adaboost (Wang et al., 2006). However, point-based methods only make use of local features without considering contextual information. As a result, these methods are subject to accuracy limitations since building ridges and vegetation can share the same geometric properties. To further alleviate the detection errors, Zhou and Neumann (2008) presented a post-processing voting algorithm. The point is assigned to a vegetation class only if a certain percentage of its neighbouring points from the labelled data are vegetation. Lafarge and Mallet (2012) encoded the spatial coherence in a graphical model such that the un-supervised classifier contains the
energies from both data terms (combination of normalized features) and the pair-wise terms (spatial coherence between neighbouring points). A multi-label graph cuts algorithm is then employed for a fast optimization of label energies.

Instead of directly working on individual points, segment-based classification initially transforms the raw points into regions that share similar characteristics. The extracted regions are then employed for classification in the subsequent step. In this way, the classification task can be strengthened by first aggregating points and then analysing segment attributes rather than individual points (Filin and Pfeifer, 2006). Segmentation enriches geometric and topological information to provide more reliable classification results (Vosselman, 2009). Sithole and Vosselman (2005) proposed the scan line split-and-merge method to extract smooth segments where the segment boundary has a distinguishable height difference from its surroundings. The segments are then labelled depending on their geometrical relationships. Zhang et al. (2013) adopted the region growing algorithm to extract plane structures. A total of 13 features including geometrical features, echo features, radiometric features and topological features are then employed from the derived segments in a SVM classifier. Shapovalov et al. (2010) split point clouds by $k$-means clustering in the form of over-segmentation. For each segment, a total of 68 features are employed in a random forest classification to derive likelihood labelling.

Recent work of Xu et al. (2014) employs multiple-entities and their corresponding attributes to distinguish classes. Particularly, attributes from points, planar segments and segments from mean shift are all input into a four-step classifier. After filtering terrain points, segmentation is applied on non-terrain points. Planar attributes allow the water surfaces, ground surfaces, building roofs, vegetation and undefined objects to be distinguished. In the next step, points on walls or roof elements are classified based on the contextual information of a building using point-wise attributes. The last step is designed to eliminate errors, in which features from mean shift segments are employed to identify the vegetation points on the rooftop.

The detected urban objects and principal methods in the reviewed methods are summarized in Table 2.2. Due to the terrain regularly serving as contextual information, many efforts on data filtering have been made. This table is useful for the recognition of detected urban objects via the reviewed methods.
Table 2.2 Summary of urban object detection methods.

<table>
<thead>
<tr>
<th>Authors</th>
<th>Detected object</th>
<th>Method</th>
</tr>
</thead>
<tbody>
<tr>
<td>Lindenberger (1993)</td>
<td>Terrain</td>
<td>Morphology filtering</td>
</tr>
<tr>
<td>Vosselman (2000)</td>
<td>Terrain</td>
<td>Morphology filtering</td>
</tr>
<tr>
<td>Shan and Sampath (2005)</td>
<td>Terrain</td>
<td>Morphology filtering</td>
</tr>
<tr>
<td>Meng et al. (2009)</td>
<td>Terrain</td>
<td>Morphology filtering</td>
</tr>
<tr>
<td>Zhang and Chen (2003)</td>
<td>Terrain</td>
<td>Progressive morphology filtering</td>
</tr>
<tr>
<td>Chen et al. (2007)</td>
<td>Terrain</td>
<td>Progressive morphology filtering</td>
</tr>
<tr>
<td>Sithole (2001)</td>
<td>Terrain</td>
<td>Adaptive morphology filtering</td>
</tr>
<tr>
<td>Kim and Shan (2011a)</td>
<td>Terrain</td>
<td>Adaptive morphology filtering</td>
</tr>
<tr>
<td>Kraus and Pfeifer (1998)</td>
<td>Terrain</td>
<td>Interpolation-based filtering</td>
</tr>
<tr>
<td>Elmqvist et al. (2001)</td>
<td>Terrain</td>
<td>Active contour</td>
</tr>
<tr>
<td>Pfeifer et al. (2001)</td>
<td>Terrain</td>
<td>Hierarchical interpolation-based filtering</td>
</tr>
<tr>
<td>Chen et al. (2013)</td>
<td>Terrain</td>
<td>Hierarchical interpolation-based filtering</td>
</tr>
<tr>
<td>Sohn and Dowman (2002)</td>
<td>Terrain</td>
<td>Downward and upward TIN densification filtering</td>
</tr>
<tr>
<td>Chen et al. (2012)</td>
<td>Terrain</td>
<td>Upward DTM densification filtering</td>
</tr>
<tr>
<td>He (2010)</td>
<td>Terrain</td>
<td>Smoothness segment-based filtering</td>
</tr>
<tr>
<td>Sithole and Vosselman (2005)</td>
<td>Non-terrain object, bridge and terrain</td>
<td>Scan-line segment-based filtering</td>
</tr>
<tr>
<td>Zhou and Neumann (2008)</td>
<td>Terrain, vegetation and building</td>
<td>SVM classifier and segment-based classification</td>
</tr>
<tr>
<td>Lafarge and Mallet (2012)</td>
<td>Terrain, vegetation, building and noise</td>
<td>Multi-labelling graph cut</td>
</tr>
<tr>
<td>Zhang et al. (2013)</td>
<td>Terrain, vegetation, building, power line and vehicle</td>
<td>Segment-based classification</td>
</tr>
<tr>
<td>Shapovalov et al. (2010)</td>
<td>Terrain, vegetation, building and vehicle</td>
<td>Segment-based classification</td>
</tr>
<tr>
<td>Xu et al. (2014)</td>
<td>terrain, water, roof, roof element, vegetation, wall and others</td>
<td>Multiple-entity classification</td>
</tr>
</tbody>
</table>
2.3 3D Building Reconstruction

2.3.1 Building Footprint Extraction

The main problem in building footprint extraction is that sampled points fail to preserve accurate edge information. Boundaries directly from LiDAR data often exhibit a zigzag pattern, partially due to either occlusions from neighbouring high objects, such as overhanging trees, or to inherent deficiencies in the LiDAR data itself, such as low sampling rate, unavoidable noise, and irregular point distribution. Auxiliary information, such as building topology maps (Vosselman and Dijkman, 2001) or high-resolution images (Cheng et al., 2008) help in determining footprints with higher accuracy. When auxiliary information is absent or out-of-date compared to the LiDAR data, the LiDAR data alone must suffice for building outline generation.

To recover regular shape, some methods determine dominant directions from boundary points. Alharthy and Bethel (2002) measured two dominant directions by determining peaks in the histogram of angles. Zhou and Neumann (2008) extended the work by finding multiple dominant directions from histogram analysis using tangent directions of boundary points. Other methods for determining principal directions are based on Mean-shift clustering algorithms (Dorninger and Pfeifer, 2008) or minimum bounding rectangles (Arefi and Reinartz, 2013). These methods employ a rectilinear prior to regularize polygonal shape. Such an assumption is not appropriate in many cases, especially when buildings exhibit complex structure.

As an alternative strategy, Sester and Neidhart (2008) relied on an explicit representation of boundary shape using a Random Sample Consensus (RANSAC) method for line segment extraction. The extracted segments provided information on angle transition, which was used to impose constraints of parallelism or perpendicularity. Another strategy initially represents raw building boundaries as preliminary polygons, and then the outlines are regularized by direction alignment. Sampath and Shan (2007) built the preliminary boundary via a Douglas-Peucker (DP) method (Douglas and Peucker, 1973) and the regularized polygon was processed using a rule-based alignment. As suggested in Neidhart and Sester (2008), results from DP simplification can be unsatisfactory because building characteristics are not necessarily retained. Weidner and Förstner (1995) employed a local MDL approach to approximate the DP result and meanwhile imposed soft constraints to regularize
polygonal shape. A similar approach can be found in Jwa et al. (2008) where MDL is extended by adding a global directional constraint. Wang et al. (2006) also followed this general workflow for preliminary boundary extraction, leading to a new regularization method where the building footprint is determined by maximizing the posterior expected value. The prior is formulated as the belief in the likelihood of various hypotheses and the fitting error was employed to encode the probability of boundary points belonging to a particular building footprint model.

### 2.3.2 Planar Model Segmentation

An essential part of rooftop reconstruction is the clustering of the unorganized point cloud into planar patches. Each detected plane serves as a proxy for a set of corresponding points to provide subsequent modelling with a high-level of abstraction about rooftop structure. The importance of LiDAR data segmentation has resulted in extensive research and segmentation has been widely investigated in photogrammetry and computer vision. According to the processing element, two major classes of segmentation techniques can be pointed out: point-based segmentation and line-based segmentation. The following subsection details popular approaches for segmentation of LiDAR data into planar patches.

In quite a few segmentation approaches, coordinates from point clouds are directly employed to recognize the underlying shapes. The Hough Transform (Hough, 1962) is a proven way to successfully detect parameterized shapes in 2D as well as 3D. The Hough Transform is wildly used for straight line detection in digital images in a 2D domain. The traditional Hough Transform has been extended by Borrmann et al. (2011) to point clouds in 3D so as to recognize planar shapes. The principle of the Hough Transform is to determine geometric shapes in the Hough feature space rather than in the object space. For instance, a plane defined in 3D object space is formulated as $A \cdot x + B \cdot y + C \cdot z + D = 0$. To avoid the problem of infinite slopes when describing vertical planes, a plane can be represented in polar coordinate form as $\cos \phi \cos \theta \cdot x + \sin \phi \cos \theta \cdot y + \sin \theta \cdot z = \rho$, with $\phi$ the azimuth, $\theta$ the elevation, and $\rho$ the normal distance from the origin to the plane. The three parameters determine the characteristics of the plane and also define the 3D Hough space spanned by $\phi$, $\theta$ and $\rho$ so that each position in the feature space corresponds to a unique plane in the object space. On the other hand, each point in the object space
can be transformed as a sinusoidal surface in the Hough space. The sinusoidal surface describes all possible variants of the plane that contain the original point. Supposing a set of coplanar points in the object space, a unique position intersection from the transformed surfaces in the Hough space should exist. This scheme works globally on the data and thereby may result in spurious planes if some noise points exhibit coincidence in one plane (Vosselman and Dijkman, 2001). Overby et al. (2004) rejected a spurious plane if the projected plane failed to span a considerable area. Vosselman and Dijkman (2001) employed ground plans to separate spurious planes in the same bin of Hough space. Huang and Brenner (2011) proposed constraints that roof planes sharing a horizontal ridge have opposite azimuths and planes sharing a diagonal ridge have perpendicular azimuths. In searching the Hough space with constraints, joint planes can be extracted simultaneously.

Another well-known point-based approach for extraction of planar shapes from LiDAR data utilizes the RANSAC (Fischler and Bolles, 1981). RANSAC extracts a model by randomly drawing minimal sets from the point cloud and estimating corresponding shape parameters. A minimal set is the smallest number of points required to uniquely define a given type of geometric primitive. For example, the minimal set to define a plane shape need three points. The resulting candidate shapes are verified against the entire points in dataset to determine how many points belong to the shape. After a given number of trials, the shape which approximates the most points is extracted and the algorithm continues on the rest of data. RANSAC-based methods are limited by high computational complexity for large-scale point clouds because they need to apply to the entire dataset. Furthermore, the original RANSAC algorithm was designed to handle single structure. It can only extract one model at a time, resulting in an iterative-and-subtractive manner to extract multiple models from a dataset. Huang and Brenner (2011) pointed out the iteration problem that points belonging to two adjacent planes may be removed too early with the initially found plane. To address this issue, Schnabel et al. (2007) framed the shape extraction as an optimization problem defined by a score function. In their definition, the deviation of the point normal from the shape surface is used to formulate a score function. Tarsha-Kurdi et al. (2007a) solved this problem by applying mathematical morphology procedures for the detected plane to eliminate isolated points which should belong to other roof planes. Although it is an iterative process, Tarsha-Kurdi
et al. (2007a) suggest that RANSAC has better performance than the Hough Transform in terms of efficiency and quality.

Region growing is also a popular method for segmentation of large-scale data. Region growing approaches exploit the connectivity information. A planar region is aggregated from a subset of its neighbouring points when they are coplanar. Selecting a suitable subset is critical for region growing, and a robust subset can be picked manually or automatically. Leonardis et al. (1995) detected the potential roof regions based on the Minimum Description Length principle. Verma et al. (2006) selected the potential roof regions through local flatness estimation. After identifying seed regions, the growing procedure is carried out by comparing the similarity between the seeds and their direct neighbours. If a certain criterion is satisfied, the neighbour point is recognized as the same shape and the region will expand to find more neighbours. Normal derivation is one of the most-used criteria to measure the similarity. Zhou and Neumann (2008) used covariance analysis to determine the normal direction of each point. Region growing starts by selecting a seed point classified as non-vegetation point and the growing process continually adds points to the same region if the normal difference is small. As a drawback, the normal estimation is often ambiguous in areas of overlapping LiDAR strips. Vosselman and Dijkman (2001) explored region growing based on a height residual analysis from a prior primitive. The 3D Hough Transform is firstly applied on a local region to estimate parameters of the seed plane. Then the seed plane is extended to a neighbouring region if the residual from the neighbouring point to the plane is small. Wang and Tseng (2004) adopted a split-and-merge paradigm as another region growing strategy. The whole data is split using octree data structure until all points inside each sub-node are coplanar. After that a merge process is performed on adjacent nodes when the two neighbouring planes share similar normals and displacement values.

While all the segmentation techniques mentioned above employ an iteration scheme for identifying multiple structures, the clustering technique represents another segmentation scheme to detect multiple structures simultaneously. The principle of clustering is that a dataset with different properties will be partitioned into a number of disjoint subsets in feature space such that each subset shares homogeneous properties. In all clustering algorithms, two key parts have to be
considered, namely, the feature space in which the clustering takes place, and a method of partitioning the feature space into subspaces without a priori information about the number of clusters or the cluster centres. This process is also referred to as unsupervised subspace learning. Sampath and Shan (2010) defined the feature vector as the surface normal of each planar point and then mapped them to Gaussian Sphere space where fuzzy $k$-mean clustering is utilized for subspace partitioning. Dorninger and Pfeifer (2008) utilised the normal vector as well as the normal distance as the feature vector in extended Gaussian space and used density-based clustering to identify subspaces so that parallel segments can be separated simultaneously. To estimate precise surface normal, Filin and Pfeifer (2006) used a slope adaptive neighbourhood to automatically set the optimal neighbourhoods whilst Liu and Pomerleau (2012) used tensor voting to enhance normal coherence.

Rather than using points for segmentation, the other group employs straight line segments as the basic element for planar surface extraction. A significant reason for the adoption of line segments as the intermediate object is based on the observation that, in a scan line, the points on a planar surface form a straight line segment. Moreover, all points belonging to a straight line lie on the same planar surface. The main advantage of the line-based algorithm is that it greatly reduces the data volume to handle when processing, such that the use of line segments makes the segmentation much faster (Jiang and Bunke, 1994).

Line-based segmentation was introduced in Jiang and Bunke (1994) for use in range images, and it was then adopted for building reconstruction (Hebel and Stilla, 2008), road mapping (Miyazaki et al., 2014) and robot navigation (Georgiev et al., 2011) employing LiDAR point clouds. Generally, the methods followed a typical split-and-merge paradigm. In the step of splitting, each scan line is interpreted as several straight line segments which have higher level geometric information than points. The segmentation will consider only the set of line segments as the basic elements. In the merging step of region growing, a seed plane is derived by similarity measurement among neighbouring line segments.
2.3.3 Rooftop Reconstruction

The main task of rooftop reconstruction is the extraction and arrangement of building primitives. Primitive extraction is the fundamental transforming step from low-level data to a high-level model description. Theoretically, a primitive extraction task can be interpreted as partitioning the whole dataset into a distinct model. In practice, different building reconstruction methods have their own definitions of meaningful primitives. Existing methods of building segmentation and reconstruction from LiDAR data can be classified into two broad categories: the explicit and implicit approaches.

Most of the existing techniques make the assumption that the building is a polyhedron model containing parameterized surface shapes. Most common are planes in the context of modelling building rooftops. This method is also termed a data-driven approach because a segmentation module interprets the building points as roof patches. From the information of segments, the data-driven approach follows a typical pipeline: 1) extraction of line features from segments; 2) extension of derived lines to recover corner features from insufficient sampling; and 3) arranging features to form a regular building model. A line feature is extracted as the boundary of a segment. While inner bounds are extracted from the intersection of adjacent segments (Oude Elberink and Vosselman, 2009), outer bounds are derived from roof steps (Rottensteiner, 2005). In order to guarantee a water-tight model, Sohn et al. (2008) proposed a Binary Space Partitioning (BSP) tree to globally recognize roof corners. Lafarge and Mallet (2012) propagated primitive labels using a Markov Random Field (MRF) approach to recover plane adjacency. A common issue is the alignment of the planar segments in order to impose geometric regularity on the final rooftop models. Brenner (2000) adjusted planes through prior orientation knowledge from existing topographic maps. Vosselman (1999) restricted building outlines to the dominant building orientation. Sohn et al. (2012) achieved implicit vector regularization using the MDL principle; the data-driven approach being adaptive to complex shapes. However, the main limitation is that features from local extraction are unstable because data shows noise or irregular sampling (Huang and Brenner, 2011).

Alternate explicit strategies have been proposed to assume that a building consists of rectangular building roof primitives such as hipped, gable, mansard, etc. Herein,
the explicit strategy falls into the model-driven category. This assumption is especially true for simple buildings in rural or suburban areas. To extract the building shape, Maas and Vosselman (1999) employed invariant moments of rooftop points to recognize a standard gable roof with small dormers. In order to cope with more complicated models, extended generative shapes in a pre-defined building library are hypothesised and verified (Kada and McKinley, 2009; Lafarge et al., 2008). Huang et al. (2013) employed the Reversible Jump Markov Chain Monte Carlo (RJMCMC) method for shape detection and variable dimensions searching. Rather than using the point clouds, robust shape recognition can also be achieved from the topological relationships of segments. Verma et al. (2006) introduced a topological graph of roof segments, which establishes the link between planar structures and building primitives. Schnabel et al. (2008) detected semantic entities by using a topographic graph. Oude Elberink and Vosselman (2009) presented a sub-graph matching approach to determine building primitives. Milde et al. (2008) presented a formal grammar based on graph interpretation, by which means more complex roof patterns can be inferred. The major advantage of the model-driven strategy is that it can always generate topologically consistent models even from sparse LiDAR point clouds (Henn et al., 2013). However, a practical challenge faced is to divide a building into sub-model shapes when dealing with complex building structure (Sohn et al., 2012). Some methods (Kada and McKinley, 2009; Vosselman and Dijkman, 2001) tackle this issue by decomposing the footprint into cells. Due to the limited details of the given footprint, an accurate partition is often hard to achieve. The recent work of Lin et al. (2013) addresses the decomposition problem by introducing planarity, symmetry and convexity constraints such that the building point clouds are decomposed into basic blocks. Nevertheless, these standard building primitives are often not expressive enough to generate arbitrarily buildings, which subsequently require manual modification.

In contrast to the explicit reconstruction approaches described above, which construct parameterized models to best fit the LiDAR data (Haala and Kada, 2010), the following techniques employ smoothness prior to approximation of the piecewise surface from a triangular mesh. The approach of Hoppe et al. (1992) computes a signed geometric distance field to the underlying surface, and extracts an isosurface from the distance field using the marching cubes algorithm. The generated surfaces
implicitly represent the building roofs as multiple non-parametric height layers. Zhou and Neumann (2011) employed a dual contouring method on hermite data to extract structural pieces. By exploring structural features, roof surfaces are generated via topology-preserving simplification. Implicit approaches provide a robust way of crack-free models with arbitrary shapes. Nevertheless, a low-level model is unsuited for representing a building rooftop, which often has characteristic sharp features.

2.3.4 Building Representation

A geometric building model can be represented with a triangle mesh, *Boundary representation* (B-rep) and *constructive solid geometry* (CSG) (Brenner, 2005; Haala and Kada, 2010). Generally, a building model is generated by combinations of extracted building parts which are mostly characterized by their roof structures. In reality, the categorization is closely related to the method used for segmentation. Implicit building models represent their roofs and boundary polygons by multiple triangles because they are derived from mesh simplification (Zhou and Neumann, 2011). The B-rep method represents a building by indirectly representing its bounding surface as well as topological information. Compared to triangles, B-rep has a smaller model description, as only the vertices and their topological relationships are stored. Most prismatic models and polyhedral models favour B-rep to represent planar roof polygons (Sampath and Shan, 2010). Since few points on building walls are collected, wall polygons are generated by extruding the building footprint to the ground. In CSG, the building is represented by combining basic primitives using Boolean operators (union, difference, and intersection). Usually, the primitives are parametric models, typically including cuboids, Cylinders, prisms, pyramids, cones. CSG guarantees the model is watertight and requires fewer parameters than B-rep. In addition, it can represent more complex structure, such as domes. However, with CSG it is hard to represent complex buildings because they are not easily ‘built’ from limited shape primitives. A comparison between data-driven and model-driven methods for building reconstruction has been made by Tarsha-Kurdi et al. (2007b), who state that the model-driven approach is faster but does not visually deform the building model. In contrast, the data-driven approach tends to model each building detail to obtain the nearest polyhedral model, but it usually visually deforms the real shape of the building. Therefore, only a few model-
driven methods (Verma et al., 2006; Vosselman and Suveg, 2001) fit CSG building models.

The principle methods, model details and representation for the reviewed approaches are listed in Table 2.3. Related to the applied method developed here, representation is made from B-rep, CSG and a mesh. This table is useful for an understanding of representation choices made in different approaches.

Table 2.3 Summary of reconstruction methods.

<table>
<thead>
<tr>
<th>Authors</th>
<th>Method</th>
<th>Level of detail</th>
<th>Representation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Alharthy and Bethel</td>
<td>Histogram analysis</td>
<td>LOD 1</td>
<td>B-rep</td>
</tr>
<tr>
<td>(2002)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Zhou and Neumann</td>
<td>Histogram analysis</td>
<td>LOD 1</td>
<td>B-rep</td>
</tr>
<tr>
<td>(2008)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Neidhart and Sester</td>
<td>RANSAC line fitting</td>
<td>LOD 1</td>
<td>B-rep</td>
</tr>
<tr>
<td>(2008)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Sampath and Shan</td>
<td>Explicit reconstruction</td>
<td>LOD 1</td>
<td>B-rep</td>
</tr>
<tr>
<td>(2007)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>(Weidner and Förstner, 1995)</td>
<td>Local MDL</td>
<td>LOD 1</td>
<td>B-rep</td>
</tr>
<tr>
<td>Jwa et al. (2008)</td>
<td>Global MDL</td>
<td>LOD 1</td>
<td>B-rep</td>
</tr>
<tr>
<td>Wang et al. (2006)</td>
<td>Bayesian statistics</td>
<td>LOD 1</td>
<td>B-rep</td>
</tr>
<tr>
<td>Oude Elberink and Vosselman (2009)</td>
<td>Sub-Graph fitting</td>
<td>LOD 2</td>
<td>CSG</td>
</tr>
<tr>
<td>Sohn et al. (2008)</td>
<td>BSP tree</td>
<td>LOD 2</td>
<td>B-rep</td>
</tr>
<tr>
<td>Lafarge and Mallet (2012)</td>
<td>Primitive fitting</td>
<td>LOD 2</td>
<td>Mesh</td>
</tr>
<tr>
<td>Brenner et al. (2001)</td>
<td>Primitive fitting</td>
<td>LOD 2</td>
<td>B-rep</td>
</tr>
<tr>
<td>Vosselman (1999)</td>
<td>Primitive fitting</td>
<td>LOD 2</td>
<td>B-rep</td>
</tr>
<tr>
<td>Sohn et al., 2012</td>
<td>Primitive fitting + MDL</td>
<td>LOD 2</td>
<td>B-rep</td>
</tr>
<tr>
<td>(Maas and Vosselman, 1999)</td>
<td>Model-driven</td>
<td>LOD 2</td>
<td>CSG</td>
</tr>
<tr>
<td>(Huang et al., 2013)</td>
<td>Model-driven</td>
<td>LOD 2</td>
<td>CSG</td>
</tr>
<tr>
<td>(Verma et al., 2006)</td>
<td>Model-driven</td>
<td>LOD 2</td>
<td>CSG</td>
</tr>
<tr>
<td>(Schnabel et al., 2008)</td>
<td>Model-driven</td>
<td>LOD 2</td>
<td>B-rep</td>
</tr>
<tr>
<td>(Henn et al., 2013)</td>
<td>Model-driven</td>
<td>LOD 2</td>
<td>CSG</td>
</tr>
<tr>
<td>(Lin et al., 2013)</td>
<td>Model-driven</td>
<td>LOD 2</td>
<td>B-rep</td>
</tr>
<tr>
<td>(Zhou and Neumann, 2011)</td>
<td>Mesh simplification</td>
<td>LOD 2</td>
<td>Mesh</td>
</tr>
</tbody>
</table>
2.4 Summary of Existing Approaches

The reviewed approaches individually demonstrate that the employment of various datasets and strategies for different landscapes is promising. In the detection phase, the use of 3D coordinate information helps to alleviate the difficulties of image-based interpretation because elevation information is reliable for the inference of geometric properties. Since the early 2000s, the success of detection approaches based on LiDAR data has been reinforced by the constantly increasing accuracy and density of LiDAR measurements (Haala and Kada, 2010). In the reconstruction phase, the building is often reconstructed as a prismatic model when the resolution of sensor data is low, whilst in high-resolution data, the building is treated as a polyhedral model and consequently the rooftop is usually reconstructed through a combination of multiple primitive structures. Most present methods have been developed to reconstruct buildings with relatively simple structure, since the handling of buildings with complex structures is an enduring challenge (Habib et al., 2010). Semi-automated systems with more or less manual user interaction are still required to recognize complex building, e.g. CyberCityModeller (Grün and Wang, 1999), O-Snap (Arikan et al., 2013). Since a great number of buildings exit in city-scale mapping and costs of interactive recognition are high, fully automated tools are still sought to satisfy the need for efficient model generation (Förstner, 1999). Recent research has proposed modelling complicated buildings using prior knowledge of building shape so as to develop more robust methods to tackle complex cases. Results show that the combination of shape knowledge and geometry extractors performs better than either extractor alone (Xiong et al., 2014).

Due to the absence of modern benchmark data sets, researchers have usually evaluated their own results over various test data sets and study areas having different building type configurations. Moreover, criteria for quality evaluation are quite various, which make a statistical comparison of existing methods more difficult. In order to make the results of reconstruction methods more comparable, benchmark datasets are now provided by the International Society for Photogrammetry and Remote Sensing (ISPRS). Public accessible pan-sharpened colour infrared (CIR) images with a Ground Sample Distance (GSD) of 8 cm and airborne laser scanning (ALS) data of 4-7 points/m² are available for the purpose of urban object detection and 3D building reconstruction. Such data sets with ground truth help to clarify
research techniques, quality assessment and existing problems. Rottensteiner et al. (2014) have reported the research results from several research groups on building modelling. Benchmark tests indicate that LiDAR is currently the preferred data source for building reconstruction. A few of the methods use the fusion of LiDAR and images, but few rely solely on images. An interesting observation is that methods using LiDAR data have the same performance in planimetric accuracy as those based on images, even though the images generally have a higher sampling rate (Rottensteiner et al., 2014). In the case of height accuracy, reported methods based on LiDAR data have exhibited smaller height errors than image-based methods. The large variations of completeness between 68.5% and 82.8% indicate that small roof planes remain undetected from some reported methods. All the methods have good performance of correctness, and some achieve 100% correctness. Over-segmentation also has large variation, between 0 and 11, while under-segmentation is relatively large for all methods.
Chapter 3 Research Strategy
3.1 Strategy Introduction

In this chapter, a newly designed and developed strategy for accurate 3D building generation using airborne LiDAR data will be presented. Based on the objectives, a potential workflow has been proposed that responds to the requirements for automated extraction and reconstruction of buildings from LiDAR data. In order to increase success rate and reliability of the results, the system needs to contain a set of geometric constraints for feature extraction, and it needs to make use of available observations as much as possible. In the following, the model complexity is firstly described in Section 3.2. Based on different levels of details, the strategy for 3D building generation is detailed in Section 3.3.

3.2 Modelling Complexity

In the context of model generation, it is essential to choose the appropriate model complexity because urban objects come with a diversity of shapes and detail. The detail of a model varies with particular requirements. For example, flood disaster analysis or infrastructure planning focuses on terrain shape, while solar power potential assessment is more concerned with the orientation of roof facets. In addition, the level of detail within a 3D model is subject to the resolution of the input LiDAR data. When given data with large point spacing, only rough and generalized features can be extracted since tiny structures fail to be measured. Therefore, there will probably be no single level of detail that fulfils all application requirements.

Different typologies of building models can be produced according to their Level of Detail (LOD). The international LOD standard has been developed by the Open Geospatial Consortium, as described in OpenGIS City Geographic Markup Language (CityGML) (Kolbe et al., 2005). CityGML provides a hierarchical description of city entities in five different levels (from LOD0 to LOD4), where a higher level corresponds to a more detailed representation of the building model. The lowest level of detail (LOD0) is essentially a DTMs over which an aerial image or a map may be draped, and thus there is no direct relationship with buildings. The first level of detail (LOD1) presents buildings as a simple box representation with a flat roof structure. The extracted outlines are employed to define the building footprint and the flat roof is extruded from the DTM to a certain height by a mean or
maximum height of LiDAR points within the footprint. In contrast, a building generated to the second level of detail (LOD2) has a more discernable roof structure and thematically differentiated boundary surfaces. LOD3 is defined by adding architectural elements with detailed façade structures potentially including doors and windows. Finally, LOD4 completes a LOD3 model by adding interior structures, such as rooms, stairs, interior doors and even furniture. Figure 3.1 illustrates the different levels of detail in an urban scene. From the definition, the representation of LOD0 to LOD2 shows strong 2.5D characteristics and their data source is mainly from airborne laser scanning or detailed aerial photogrammetry. LOD3 and LOD4 exhibit full 3D information and thus the data source is from façade photographs and/or Terrestrial Laser Scanning (TLS).

![LOD0, LOD1, LOD2, LOD3, LOD4](image)

Figure 3.1 The five levels of detail defined by CityGML (Kolbe et al., 2005).

### 3.3 3D Building Reconstruction Strategy

Figure 3.1 indicates the details in building generation from LiDAR data. Buildings have various forms of representation, depending on their own characteristics and the data resolution. It has been noted in Chapter 2 that many general models for building reconstruction have been developed. These models are linked to the characteristics of simple buildings. Most often the general models chosen would not be suitable for another type of building or in another type of landscape. Buildings which fit the
models are extracted. If the building, as it appears in LiDAR data, does not correspond to the pre-defined model, the building generation systems may yield many omissions and false extractions. Thus, the defined roof models should be generic enough to reach a good rate of completeness but precise enough to avoid false extraction. In manual generation, the reconstruction is based on operator experience and involves a high level of prior knowledge about topographic objects. One of the simplest solutions to build a robust model is to use 3D geometric constraints from prior knowledge with fabrication considerations.

As can be seen from the literature review in Chapter 2, due to the complexity of LiDAR data, many building generation algorithms become inefficient. In this research, two approaches are taken to overcome this issue. On the one hand, a more reliable element is employed to represent the LiDAR data covering buildings. More specifically, straight line segments are derived from point clouds as the basic elements for data processing. The reason for working with line segments instead of points is that a line segment is a simple and robust geometric primitive to describe most man-made environments where straight edged objects comprise many of the environmental features. In addition, line segments are able to provide a wealth of information such as geometric properties. By converting raw point clouds into line segments, an abstract of point clouds is obtained with a minimal loss of information, which reduces much of the processing elements and requires less memory for loading. Therefore, by choosing straight line segment as the starting point, the proposed system is able to perform building detection and rooftop reconstruction over a large area. On the other hand, a more robust modelling approach is designed by making use of both observation data and prior constraints. In particular, the extracted cues are exploited to complete feature extraction and infer more detailed structures. The features derived from robust extraction provide initial contextual information which is often incomplete due to the limited data resolution. LiDAR data give observations in such areas, but the observations are too sparse to generate reliable geometry. Therefore, the information provided by the extracted models or features can help in understanding the scene, while the sparse observation points provide real data useful to verify the hypotheses proposed from prior knowledge. The system strongly relies on the following three aspects:
1. Use of extracted cues about the preliminary models or extracted features. The basic preliminary models used are a coarse DTM or approximation polygon, while some extracted features, such as building outlines or roof patches, will also be used. The extracted cues are reliable to provide contextual information.

2. Use of prior knowledge to restrict the search hypotheses. This confines the desired solution to have a form that is agreeable with prior knowledge. For example, spatial coherence encourages neighbouring points to belong to the same class. Using spatial coherence as a prior, classification will encode interactive potential to alleviate classification error on object boundaries.

3. Use of observation data to verify multiple hypotheses. This confines the desired solution to close the observation data as much as possible. Using geometric fitness, hypotheses can be evaluated by residual error.

The energy minimization approach provides an expressive and elegant framework to naturally encode ‘soft’ problem constraints such that the optimal solution can be discriminated by solving a minima problem. It has the potential to avoid the framework being trapped by hard constraints (Kolmogorov and Zabih, 2002). The general form of the energy function to be minimized can be expressed as

\[
E(f) = E_{data}(f) + \lambda \cdot E_{prior}(f)
\]

A lower cost in the data term indicates higher agreement with the observation data while a lower cost in the prior energy term means a solution is more in accordance with prior knowledge. The energy function is usually the sum of terms corresponding to different soft constraints encoding data, and prior knowledge of the problem. It is clear that smaller values of the energy function indicate better potential solutions. Minimization of the energy function can be justified using Bayesian statistics (Geman and Geman, 1984) from optimization approaches, such as message passing (Wang and Daphne, 2013) and $\alpha$-expansion (Boykov and Jolly, 2001).

Energy minimization has been used in computer vision and photogrammetry to infer information from observation data. For instance, Yang and Förstner (2011) formulated image interpretation as a labelling problem, where labelling likelihood is calculated by a randomized decision forests and piecewise smoothness is taken as a
prior and is encoded by spatial coherence based on Markov Random Fields. Shapovalov et al. (2010) and Lafarge and Mallet (2012) also utilised piecewise smooth priors for scene interpretation within point clouds. Kolmogorov and Zabih (2002) imposed spatial smoothness in a global cost function over a stereo pair of images to determine disparity. Instead of using a smooth prior, Zhou and Neumann (2010) combined quadratic error from the boundary and surface terms to achieve both 2D boundary geometry and 3D surface geometry.

The general strategy is shown in Figure 3.2. In this proposed system, the energy function is deeply involved for the task of “building detection and modelling” to alleviate process errors. The definitions of the energy terms are quite different in the following chapters for various different tasks, such as LiDAR data filtering, building detection, building footprint extraction and plane segmentation. Generally speaking, robust features or preliminary models are first extracted from LiDAR data. Raw observation data as well as prior knowledge are employed to improve and complete the preliminary models.

![Diagram of General strategy for 3D building modelling.](image)

To robustly detect and reconstruct a building model, the hierarchical description of CityGML is used to generate models with different levels of detail. Since the rooftop is almost the only visible part of building in airborne LiDAR data, it is generally not
feasible to determine the geometric structure on façades or inside buildings. Therefore, the reconstructed building model is up to LOD2 complexity.

Figure 3.3 shows the flowchart of the proposed system. The first module of this research proposes a progressive segmentation method for terrain filtering. To overcome the problem of ambiguities in complex terrain, the segmentation is performed on a graphical model, which encodes both point-wise closeness constraints as well as pair-wise smoothness constraints to achieve good segmentation accuracy and to alleviate ambiguity on segment boundaries. The spatial structure is adopted to provide interactive information because the terrain exhibits strong spatial coherence such that neighbouring points are encouraged to have the same class label. A primary focus of the method is on the adoption of a progressive manner to refine the terrain surface, exploiting the previous segmentation result to optimize the filtering result. Given a LiDAR point cloud as input, the method works directly on the raw point cloud without rasterization. The derived output of this module is the terrain model corresponding to a LOD0 product.

Using the extracted line segments from non-terrain points, the second module of this research studies the fundamental problem of building footprint extraction. The proposed method contains three main steps in terms of detection, boundary approximation and boundary regularization. Given non-terrain measurements as input, irrelevant features, such as vegetation, are eliminated from the data based on geometric attributes of line segments as well as interactive information. After that, a novel Vertex-driven Douglas-Peucker (VDP) algorithm is proposed to generate several polygon hypotheses, from which the optimal polygon is selected by minimizing energy in terms of data fitness and polygon conciseness. In other words, the boundary approximation takes an optimal polygon to be a balance between data and prior constraints. Once preliminary polygons are achieved, building boundaries are regularized via a hybrid regularization method in terms of explicit reconstruction of long edges and implicit reconstruction for the gaps between consecutive long edges. This achievement is a set of prismatic building models corresponding to LOD1 complexity.

The last module of this research focuses on detailed rooftop generation. This phase contains two main steps, namely, roof facet segmentation and topological surface reconstruction. In the first step of rooftop reconstruction, a spectral clustering based segmentation is proposed to simultaneously extract multiple planar patches from line
segments. Then in the second step, the building rooftop model is achieved by proper arrangement of extracted segments. A topologically correct model follows the original human design and ingratiates itself with human vision. In this step, the extracted footprint is employed to achieve a good quality model in the form of geometric closeness as well as reasonable topology. The delivered output is a set of polyhedral building models corresponding to a LOD2 product.

Figure 3.3 Input data, processing methods and achievements in the proposed building modelling system.
Chapter 4 LiDAR Data Filtering
4.1 Introduction to Filtering

Given raw LiDAR data as the input, the first module of this research investigates the fundamental task of terrain extraction. In reality, LiDAR systems naturally record the returns from both terrain and non-terrain surfaces, including buildings, bridges, vehicles, vegetation, etc. The entire elevation information, from all measurements, constructs a Digital Surface Model (DSM), whilst elevation information only from terrain measurements forms a DTM. In practice, DTMs are an essential part of city modelling with a complexity level of LOD0. They are widely applied for a wide range of geospatial applications. For instance, the DTM is one of the critical input datasets for flood prediction because it influences the flood direction, flood extent and flow velocity (Abdullah et al., 2009). Since no interpretation is performed during data acquisition, LiDAR terrain points have to be investigated among raw point clouds such that accurate DTMs can be obtained by the interpolation. This task of binary classification is also referred to as LiDAR data filtering.

This chapter will firstly present, in Section 4.2, the proposed progressive segmentation approach based on graph cuts. Section 4.3 reports the results and statistical comparison with three existing methods. Section 4.4 summarises the chapter.

4.2 Progressive Segmentation via Graph Cuts

This section presented a new iterative graph-cuts based approach for the filtering of LiDAR point clouds. Separation of terrain and non-terrain points in airborne LiDAR data is naturally analogous to binary labelling, where the data points are partitioned into two disjoint sets. Such labelling can be achieved through energy minimization by graph cuts (Boykov and Jolly, 2001). Graph cuts are designed to minimize the energy function in a weighted graph where the energy function defines segmentation. Graph has proven to be an effective optimization model which can enforce piecewise smoothness while preserving relevant sharp discontinuities. In particular, iterative graph cuts allow automatic refinement of soft constraints with newly labelled points, resulting in more robust segmentation, which is denoted as Progressive Segmentation via Graph Cuts (PSGC). The general scheme of the proposed approach is presented in Figure 4.1.
The raw LiDAR point clouds undergo blunder detection for outlier removal. The outliers are usually due to the multi-path effect, which generates extreme low height values. The procedure of outlier detection also allows for determination of a set of sparsely distributed terrain points over the scene, which produces a preliminary terrain surface. Meanwhile, a graph is constructed with the LiDAR points as nodes, and the links of the graph are determined by both the point properties in relation to the terrain surface and the spatial structure among points. The terrain and non-terrain points are then differentiated in an energy minimization procedure achieved by iterative graph cuts. In each iteration, the newly identified terrain points are added to the terrain surface, progressively densifying the preliminary model. In turn, the improved terrain model enables refinement of the weights of the links in the graph, facilitating effective identification of the rest of the terrain points in the successive iteration. This process repeats until no more terrain points can be identified or selected, therefore progressively completing the terrain model. As the energy minimization and graph cuts require a preliminary terrain surface model, the algorithm for outlier detection and extraction of sparse terrain points is initially
presented. In the following, the basics of energy minimization and graph cuts are presented. Afterwards, the definition of the energy function and construction for LiDAR point clouds are discussed in detail. The particular focus is on the iterative graph cuts for progressive identification of terrain points.

4.2.1 Outlier Removal and Initial Terrain Surface Generation

A modified local minimum method (Axelsson, 2000; Chen et al., 2012) with a moving grid is adopted for initializing terrain points and subsequently generating a coarse preliminary DTM. However, outliers in LiDAR data may produce errors in the generation of this DTM. While terrain and non-terrain points are the reflectance from landscape, outliers usually occur because of multi-path errors (Sithole and Vosselman, 2004). In particular, negative outliers with extremely low height values to compared terrain surface interfere with the assumption that the lowest points in relatively large areas must belong to the terrain surface and should be removed.

Some algorithms have been designed for LiDAR outlier detection. Outliers can be detected using distribution-based approaches (Meng et al., 2009; Silván-Cárdenas and Wang, 2006), mathematical morphology (Chen et al., 2007; Kobler and Ogrinc, 2007) and the density-based method (Breunig et al., 2000; Sotoodeh, 2007). While the methods differ in algorithmic design, they are based on the observation that outliers usually demonstrate excessively high or low elevation values. These ideas are followed in the current research. A hybrid detection approach is proposed, which essentially combines outlier detection and local terrain point extraction, therefore simultaneously removing outliers and generating a preliminary DTM.

The proposed approach is built upon the assumption that the LiDAR outliers are scattered with lower height values than their surroundings. Due to the complexity of the landscape over large area, outliers are detected locally in a small grid taking into account the point density and height consistency of the LiDAR points. Firstly, the points in the grid are sorted in the order of height values and the low-lying points with low density are located. These points are treated as potential outliers and they undergo further examination. Observing that outliers usually demonstrate significant height difference to their neighbouring points, two measures are calculated: the height difference between the lowest point and the second lowest point, and the
height difference between the lowest point and the average height of its neighbouring points determined from a triangulated irregular network of the points in the working area. The lowest point in the grid is considered as an outlier and will be removed if both the differences exceed a certain value. The lowest points in the remaining point body will be selected to represent terrain within the grid and they are included in the construction of a rough preliminary DTM.

The proposed approach works well, particularly in rural and residential areas. The grid size is selected as 50m by 50m, sufficiently large enough to avoid the interference of large buildings, therefore ensuring that the determined local lowest point belongs to terrain. This size is also appropriate for characterization of landscapes at rough scale. Since the preliminary DTM is only used as an approximation of terrain in the successive segmentation of graph-cut based approach, the requirement for density and detail within this DTM is not high.

### 4.2.2 Energy Minimization and Graph Cuts

Segmentation of LiDAR point clouds for DTM extraction is essentially a separation of terrain and non-terrain points. It can be considered as a binary labelling problem. Let $P$ denote a set of LiDAR points, each point $p \in P$ will be assigned a unique label in the label set $L$ \{“terrain”, “non-terrain”\}. The goal of segmentation is to find a labelling $f$ that assigns each point $p \in P$ a label $f_p \in L$, where $f$ has both piecewise smoothness and low deviation with respect to the observation data.

This labelling problem can be formulated in terms of energy minimization. Energy minimization is an expressive and elegant framework to both alleviate uncertainties in sensor data processes and remove ambiguities in solution selection (Boykov et al., 1999). It allows a clear expression of problem constraints so that the optimal solution can be determined by solving a minima problem. The segmentation form of the energy function is defined as

$$E(f) = E_{data}(f) + \lambda \cdot E_{smooth}(f) \quad (4.1)$$

$E_{smooth}$ measures the extent to which $f$ is not piecewise smooth, while $E_{data}$ measures the disagreement between $f$ and the observation data. The form of $E_{data}$ is typically
\[ E_{\text{data}}(f) = \sum_{p \in P} d(f_p) \]  

where \( d(f_p) \) measures how the deviation label \( f_p \) fits the point \( p \) in the observation data.

Considering the pairwise interaction of data points, \( E_{\text{smooth}} \) is defined as

\[ E_{\text{smooth}}(f) = \sum_{(p,q) \in N} B_{(p,q)} \cdot \delta(f_p, f_q) \]  

where \( B_{(p,q)} \) is interpreted as a pair-wise interaction function. The indicator function \( \delta(\cdot) \) is 1 if \( f_p \neq f_q \) and 0 otherwise so as to measure only the discontinuity along segment boundaries.

The major difficulty with energy minimization lies in the high computational costs involved. Typically, the energy function has many local minima. In general, a local minimum can be arbitrarily far from the optimum. In addition, local minimization techniques are naturally sensitive to initial estimates. Graph cuts have proven to be a useful optimization tool which can enforce piecewise smoothness while preserving relevant sharp discontinuities. The method has been proven to be able to locate global minima for the energy function (Kolmogorov and Zabin, 2004).

The two-label graph cuts method and the extension to multi-label graph cuts by alpha expansion algorithms are described in Boykov and Funka-Lea (2006) and Boykov and Jolly (2001). Let \( G = \langle V, E \rangle \) be a weighted undirected graph consisting of a set of nodes \( V \) and a set of edges \( E \) that connect nodes. The graph normally contains two additional special nodes that are called terminal nodes: source \((S)\) and sink \((T)\). There are two types of edges in the graph: t-link and n-link. t-links connect each node in the graph with terminals while n-links connect pairs of neighbouring non-terminal nodes. An example of a typical graph is illustrated in Figure 4.2.

Graph cuts are a set of edges such that the linked nodes are in disjoint sets while each node has to connect with only one terminal node which corresponds to its label. The minimum cut problem is to find a cut that has the minimum cost among all cuts. This is equivalent to identifying the lowest cost for a discrete labelling \( f \) that gives the optimum segmentation with energy minimization.
4.2.3 Definition of the Energy Function

Successful point cloud segmentation depends on both the formulated energy function and the adopted optimization graph cuts method. The energy function defines the required segmentation and the graph should be constructed to sufficiently represent an unstructured point cloud. The graph cuts algorithm minimizes the energy function in the weighted graph, resulting in an optimized segmentation of the point cloud.

Each point in the LiDAR point cloud is considered as a node in the graph. The graph also contains two terminal nodes $S$ and $T$, representing labels \{terrain\} and \{non-terrain\}, respectively. The edges among the nodes are defined such that each node is connected to its 3-D voronoi neighbours with an edge. All points are also connected to the terminal nodes representing labels, constructing the t-links in the graph. When mapping $E_{data}$ to the graphical model, an equivalent graph with the edge weight of \{p, S\} as the agreement with terrain labelling, and similarly the edge weight of \{p, T\} as the agreement with non-terrain labelling, is established such that any cut on the graph will result in a corresponding labelling (Vu, 2008). These weights correspond to the data term of the energy function and will be summed at each candidate labelling configuration. The weight is evaluated in relation to the height residual ($\Delta z$) between the point and the preliminary terrain surface, which is defined as

![Graphical Model](image-url)
As indicated in Equation (4.4), the weights of a point-linking between terrain and non-terrain complement each other. A lower value of $d(l_p)$ indicates the edge between them will be cut during the segmentation. For instance, a terrain point has a small $\Delta z$, resulting in the high weight edge of {$p$, S} and a low weight edge of {$p$, T}. Such a definition encourages the cut to go through the edge between the point and the non-terrain terminal so that the point will be labelled as “terrain”. In this way, the definition of data energy penalizes the terrain points above the ground, and also penalizes non-terrain points on the ground, therefore, enforcing the desired solution to conform to the terrain. $\sigma_1$ is set to 0.3m to represent an average residual of terrain points above the rough surface. Figure 4.2 illustrates the effect of the defined $E_{data}$ in the energy function. The colour of the points encodes the weight of the edge {$p$, S}, which is also termed closeness potential, changing from green to red as the value decreases. Each LiDAR point has a closeness potential that increases when close to terrain while decreasing when away from the terrain.

Figure 4.3 Weights of points linking with the preliminary terrain surface.

The edges that connect points with each other correspond to n-links in the graph. These edges are also assigned weights representing the smoothness term of the energy function. This weight corresponds to the $E_{smooth}$ in the energy function. The choice of $E_{smooth}$ is also a critical issue. An appropriate $E_{smooth}$ will enforce spatial coherence and encourage LiDAR points with smooth neighbours to be assigned with the same label and preserve discontinuity around sharp height changes.
Since the spatial structure of the point cloud derived from \textit{k-nearest neighbors} (KNN) often results in several disconnected components (Golovinskiy and Funkhouser, 2009), a 3-D voronoi neighbour system is constructed on the raw point cloud to ensure connectivity. Airborne LiDAR with nadir perspective has poor sampling on vertical structures, which leads to longer edges along terrain and object boundaries due to height interruption. The Euclidean distance between a pair of points is adopted to measure the smoothness and the interaction function $B_{(p,q)}$ is determined as

$$B_{(p,q)} = \exp\left(-\frac{D_{pq}}{\sigma_2}\right)$$

(4.5)

where

$$D_{pq} = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2 + (z_i - z_j)^2}$$

(4.6)

$\sigma_2$ is selected to be the average point spacing within the LiDAR data. As observed in Equations (4.5) and (4.6), closely located points generate larger smoothness values, while the smoothness value is small for distant points. This allows for detection of abrupt changes and therefore facilitates extraction of boundaries. An example is given in Figure 4.4, where the colour ranges from green to red representing changing smoothness from high to low. It can be seen that the larger the point distance, the lower the smoothness value. The large distance usually accompanies abrupt height changes, at the places between terrain and tree or between terrain and buildings.

![Figure 4.4 Weights of pair-wise linking.](image)

The Combination of $E_{data}$ and $E_{smooth}$ in the energy function to differentiate terrain and non-terrain points in LiDAR point clouds is achieved through energy minimization by graph cuts.
4.2.4 Segmentation of LiDAR Point Cloud with Iterated Graph Cuts

It is obvious from the last Section that minimizing the energy function in Equation 4.1 is equivalent to finding a cut in the graph that has minimum cost among all possible cuts. This minimum cut (min-cut) problem can be solved efficiently by computing the maximum flow between two terminals (Ford and Fulkerson, 1962). There are many standard polynomial time algorithms for min-cut/max-flow (Cook et al., 1998). A review is given in Boykov and Kolmogorov (2004), where the authors compared different maximum flow methods and concluded that algorithms based on augmented paths perform better. In this research, the max-flow method with improved augmented paths described in Boykov and Jolly (2001) and Boykov and Kolmogorov (2004) has been adopted.

The max-flow method for min-cut is conventionally achieved in a single optimization step. However, such a scheme suffers from high complexities of scene, making it hard to segment large datasets over large areas. In addition, this scheme usually requires significant user interaction for initial labelling (McGuinness and O’Connor, 2010). The quality of segmentation varies and insufficient initial input may result in poor segmentation and even errors (Peng et al., 2009). A number of recent publications have addressed this problem and developed algorithms for multi-scale graph cuts (Nagahashi et al., 2007; Michael Ying Yang and Förstner, 2011).

To alleviate this issue, Rother et al. (2004) proposed an iterated graph cuts approach called GrabCut for image segmentation which alternates between estimation and parameter learning and segments the image iteratively. In each iteration, the newly labelled pixels are used to refine graph parameters and the final segmentation result is achieved when the energy converges. GrabCut significantly improves image segmentation, even in the case where initial labelling is insufficient (Boykov and Funka-Lea, 2006). GrabCut also demonstrates efficiency in machine learning and computer vision (Boykov and Funka-Lea, 2006). Recently, GrabCut was extended to carry out segmentation of RGB-D data (Winn and Jojic, 2005).

The iterative scheme is adapted in this research. A coarse preliminary DTM model is first constructed to estimate data constraints directly from object space. The coarse preliminary DTM is obtained automatically (See Section 4.2.1). In each iteration,
terrain points close to the preliminary DTM are identified and labelled in the graph by a max-flow method. The newly labelled terrain points are employed to densify and refine the preliminary DTM so that a more detailed terrain surface is achieved. The improved DTM helps to refine and update the t-link of the graph, resulting in effective differentiation of terrain and non-terrain points in successive iterations. The complete procedure of progressive segmentation via graph cuts (PSGC) for filtering LiDAR data is given in algorithm 4.1 as follows:

Algorithm 4.1 progressive segmentation via graph cuts.

**Inputs:** Point cloud \( \{P\} \) of whole area, spatial neighbour finding function \( \mathcal{N}(\cdot) \)

**Result:** classified point cloud, detailed DTM

**Initialize** extracted terrain points \( \{E\} \leftarrow \Phi \), labelled terrain points \( \{L\} \leftarrow \Phi \), graph \( G \)

Apply initial terrain point detection to obtain initial terrain points \( \{I\} \)

\( \{E\} = \{I\} \)

**repeat**

**for** \( p \) in \( \{P\} \) **do**

Calculate \( d(l_p) \) on TIN surface of \( \{E\} \) using equation (4.4)

\[ d(l_p) \stackrel{add}{\longrightarrow} G \]

Find the spatial nearest point of \( p_i \) so that \( \{B\} \leftarrow \mathcal{N}(p) \)

**for** \( q \) in \( \{B\} \) **do**

Calculate \( B_{(p,q)} \) using equation (4.5) and (4.6)

\[ \text{if} \ B_{(p,q)} \notin G \]

\[ B_{(p,q)} \stackrel{add}{\longrightarrow} G \]

Perform segmentation on \( G \) to obtain point list \( \{L\} \) in terrain segment

\( \{E\} = \{L\} \)

**until** \( \text{size} \ \{E\} \) unchanged

Obtain DTM from interpolation of points in \( \{E\} \)

Note that only the weights of the data term in the iteration need to be recalculated, whereas the weights of the smoothness term remain the same. With new terrain points added to the DTM, the TIN surface of \( \{E\} \) is changed, resulting in some
relevant points having a higher closeness value. These changes of closeness affect the optimization of graph cuts on the subsequent segmentation.

4.3 Experiments

4.3.1 Test Datasets

Two test areas, both in Victoria, Australia, have been employed to evaluate the performance of the PSGC filter. The test areas have varying terrain complexity, and land cover consisting of one urban site (Unimelb) and one suburban site (Eltham). The Unimelb site is located in the University of Melbourne, near to the city centre. The terrain is generally flat, but with moderate elevation changes in the north part. Unimelb has a mixture of dense high-rise buildings, vegetation and sports fields, which is typical of an urban environment. The suburban site lies in Eltham about 25 kilometres northeast of the Melbourne CBD. The terrain of Eltham is hilly with steep slopes. The site covers diverse objects including a big shopping mall, residential houses, rivers, bridges and vegetation. Table 4.1 summarises the overall characteristics of the two sites.

Table 4.1 Characteristics of two test sites.

<table>
<thead>
<tr>
<th>Name</th>
<th>Location</th>
<th>Area</th>
<th>Terrain Characteristics</th>
</tr>
</thead>
<tbody>
<tr>
<td>Unimelb</td>
<td>Campus of Melbourne University</td>
<td>650m×700m</td>
<td>- Flat terrain</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>- High-rise buildings</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>- Sparse vegetation</td>
</tr>
<tr>
<td>Eltham</td>
<td>Eltham railway station area</td>
<td>1000m×1050m</td>
<td>- Hilly terrain</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>- Various building sizes</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>- Vegetation on steep slopes</td>
</tr>
</tbody>
</table>

The Unimelb dataset was acquired by an Optech Gemini scanner on July, 28th, 2011 and the Eltham dataset by the same scanner on October, 28th, 2008. The average point spacing is 0.47 m for Unimelb and 0.75 m for Eltham. Particularly, data gaps and outliers can be observed in both datasets. The aerial imagery and hillshaded TINs of the DSM for each of the two datasets are shown in Figures
4.3.2 Results

As mentioned in Section 4.2, the proposed filtering algorithm needs to specify the weighting parameter ($\lambda$) for the graph cuts segmentation while other parameters, i.e. the common residual ($\sigma_1$) and average point spacing ($\sigma_2$), can be automatically determined from the properties of the input LiDAR data. The weighting parameter is determined by trial and error from a small part of the reference data, based on a knowledge of landscape characteristics. Generally, flat terrain will have a low $\lambda$ value to encourage the data term, whereas terrain with steep slopes needs a high $\lambda$ value to encourage the smoothness term. This allows detection of terrain points on hilly slopes. The value of $\lambda$ has been assigned as 32 and 128 for *Unimelb* and *Eltham*, respectively. Table 4.2 summarizes the data filtering results.

| Name   | Total points | Outliers | Initial terrain points | Classified terrain points | Classified non-terrain points |
|--------|--------------|----------|------------------------|----------------------------|
| *Unimelb* | 2,575,995    | 912      | 182                    | 1,365,830                  | 1,209,253                     |
| *Eltham*  | 2,577,387    | 1,375    | 420                    | 1,401,801                  | 1,174,211                     |

In *Unimelb*, there are 2,575,995 points in the raw data, of which 912 points are detected as outliers and are eliminated. Figure 4.5(c) shows the distribution of initial terrain seeds generated from a 50×50m window size. A total number of 182 terrain points are detected. As presented in Section 4.2.4, the refinement procedure will
repeat until no more terrain points can be detected. Since the terrain of Unimelb is relatively flat, two iterations are required. In Figure 4.5(d), 1,235,227 terrain points coded in red are detected from the first segmentation, while 130,603 terrain points coded in blue are further detected from the iterative segmentation. This means that 9.56% of terrain points are detected from the refinement. The Figure 4.5(e) shows the classification result, and the shaded TIN of the DTM generated from interpolation of the terrain points is shown in Figure 4.5(f).

Figure 4.5 Filtering result of Unimelb: (a) aerial image, (b) shaded TIN of DSM, (c) initial ground points, (d) iterative segmentation, (e) classification result and (f) shaded TIN of DTM.

In Eltham, the raw data contains 2,577,387 points and 1,375 are detected as outliers. Figure 4.6(c) shows the distribution of the 420 initial terrain points. A visual inspection is carried on the detected points to ensure their validity. Due to the existence of steep slopes, the processes required four iterations for Eltham. In Figure 4.5(d), the 1,059,346 terrain points in red are detected from the first segmentation
and the 342,455 terrain points in blue are detected those in the refinement. This means also that 24.43% of terrain points are further detected from the refinement, as shown in Figure 4.6(d). Figure 4.6(e) shows the classification result. As can be seen, more detected terrain points, mostly in steep areas, result in the more detailed DTM shown in Figure 4.6(f).

![Figure 4.6 Filtering result of Eltham: (a) aerial image, (b) shaded TIN of DSM, (c) initial ground points, (d) iterative segmentation, (e) classification result and (f) shaded TIN of DTM.](image)

A special highlight is given to the steep slopes in Eltham. The initial terrain points from local minima estimation, as shown in Figure 4.7, are sparse in order to avoid the inclusion of large buildings. In the first segmentation algorithm, most of terrain points received closeness penalty from the preliminary DTM. As a result, these points are classified as non-terrain, as shown in the profile in the right side. With the iterative refinement, points on slopes are progressively classified as terrain points due to the recalculation of the closeness penalty from the more detailed DTM.
In this stage, the filtering quality is assessed visually via labelled points. Figure 4.8 shows the point labelling results in both Unimelb and Eltham. As seen in Figure 4.8 (a), most points on the ridge are correctly classified because of the smooth transition in slope. Courtyard area shown in Figure 4.8(b) is also detected correctly. This is due to the fact that the point-wise closeness potential is measured from the preliminary DTM, which avoids the generation of a topological description in the segment-based filter (He, 2010). The result in Figure 4.8(c) shows that the points around steep terrain are preserved. Also, as seen in the result of Figure 4.8(d), PSGC removes the influence of large flat-top buildings. In addition, PSGC also enables elimination of bridge objects, as shown in Figure 4.8(e). This result can be explained by the fact that segmentation is constrained by both closeness and smoothness conditions. In this case, even the bridge surface has a small smoothness constraint, the closeness constraint being still high from the local lowest points in the river bed. The impact is that the points on bridge will have a high closeness penalty and will thus be labelled as non-terrain points. PSGC overcomes a tradition problem with the smoothness segmentation method, namely that regions extend to the bridge surface if only surface smoothness is considered. A filtering deficiency can be found in the result of Figure 4.8(f) where vehicle points in white circle are classified as terrain because these points have a small height residual to the ground, and meanwhile demonstrate strong smoothness to their neighbouring terrain points.
4.3.3 Criteria for Quality Assessment

The evaluation was performed to quantify analysis of filtering errors. Sithole and Vosselman (2004) presented point-level evaluation scheme. Four evaluation metrics, defined as follows, are used for the quantitative evaluation of performance:

- Type I error corresponds to those points belonging to the terrain class that the filter has failed to recognise. Type I error is expressed as:

  \[
  Type\ \text{I}\ \text{error} = \frac{b}{a + b} \times 100\%
  \]  

  (4.7)

  where \(a\) is the number of points which have been correctly classified as terrain, while \(b\) is the number of terrain points in the reference data that have been incorrectly classified as non-terrain points. \(a + b\) represents the total number of terrain points in the reference data.

- Type II error measures those points belonging to the non-terrain class in reference that the filter has labelled as terrain. Type II error is calculated as:

  \[
  Type\ \text{II}\ \text{error} = \frac{c}{c + d} \times 100\%
  \]  

  (4.8)

  where \(c\) is the number of non-terrain points in reference data that have been incorrectly classified as terrain, and \(d\) is the number of points which have been correctly classified as non-terrain points; \(c + d\) indicates the total number of non-terrain points in the reference data.
• Total error is aimed to provide an overall indicator of those points having a different class labelling to the reference data. Total error is calculated as:

\[ \text{Total error} = \frac{(b + c)}{(a + b + c + d)} \times 100\% \]  \hspace{1cm} (4.9)

where \( b + c \) represent the number of labelling differences between the filtered and reference data, while \( a + b + c + d \) is the total number of points in the reference data.

• Kappa coefficient is the measurement of filtering performance, which is free of any bias resulting from chance agreement between the filtering result and the reference data (Richards, 2013). The Kappa coefficient is defined as:

\[ \text{Kappa coefficient} = \frac{(e - f)}{(1 - f)} \times 100\% \]  \hspace{1cm} (4.10)

where \( e \) represents the probability of correct classification with respect to the reference data, while \( f \) is the probability of chance agreement.

4.3.4 Comparison with Existing Approaches

The two datasets were processed using three prominent filtering approaches: ALDPAT, Terrascan, and upward-fusion. All three existing methods, along with PSGC, are automatic and progressive filters. ALDPAT is free software based on progressive morphological filtering (Zhang and Chen, 2003). Terrascan is a well-known commercial software system for LiDAR processing based on progressive TIN densification (Axelsson, 2000). Upward-fusion (Chen et al., 2012) and PSGC are implemented in this study based primarily for DTM refinement.

Figure 4.9-4.13 illustrates the error distribution of the five samples subject to comparison. The aerial imagery is shown for visual reference. The DTM is the point interpolation from the reference terrain points. Error distributions from the four approaches are shown in the figure. The accuracy assessment is summarised in Table 4.3. Results show that PSGC has the best performance on Sample11, Sample21, Sample22 and Sample23 in terms of the lowest total error and the highest kappa coefficient. ALDPAT produces the better result on Sample12 but the differences of both total error and kappa coefficient compared to the developed PSGC method are quite small (total error is 3.02% versus 3.64% and kappa coefficient is 93.76% versus 92.62%).
Figure 4.9 Reference data and filtering errors of Sample1: (a) aerial imagery, (b) generated DTM, (c) – (f) error distribution of PSGC, ALDPAT, terrascan and upward-fusion, respectively (Type I in green and Type II in red).

Figure 4.10 Reference data and filtering errors of Sample2: (a) aerial imagery, (b) generated DTM, (c) – (f) error distribution of PSGC, ALDPAT, terrascan and upward-fusion, respectively (Type I in green and Type II in red).
Figure 4.11 Reference data and filtering errors of Sample21: (a) aerial imagery, (b) generated DTM, (c) – (f) error distribution of PSGC, ALDPAT, terrascan and upward-fusion, respectively (Type I in green and Type II in red).

Figure 4.12 Reference data and filtering errors of Sample22: (a) aerial imagery, (b) generated DTM, (c) – (f) error distribution of PSGC, ALDPAT, terrascan and upward-fusion, respectively (Type I in green and Type II in red).
Figure 4.13 Reference data and filtering errors of *Sample23*: (a) aerial imagery, (b) generated DTM, (c) – (f) error distribution of PSGC, ALDPAT, terrascan and upward-fusion, respectively (Type I in green and Type II in red).

In terms of Type I error, PSGC showed the best performance for all samples, which means most terrain points were correctly measured to eliminate omission error. This is mainly attributable to PSGC’s exploitation of spatial smoothness to enhance labelling consistency. From the Type I error distribution of the three compared methods in *Sample11*, most errors occur in the ridge terrain and the Type I error ranges from 5.67% to 7.96%. In contrast, PSGC accurately represents the terrain structure and the Type I error is as low as 0.72%. Also as shown in the result for *Sample21*, only 2.52% of terrain points fail to be detected by PSGC, while the omission errors of the three other filters are concentrated along the steep terrain. Moreover, the courtyard in *Sample12* and highways in *Sample22* are well preserved. The Type II errors of PSGC are relatively larger than with the other filters. As can be seen by the Type II error distribution in *Sample21*, some low and dense shrub points on steep slopes are misclassified as terrain points. However, manual filtering or post-processing can be applied on the derived DTM to eliminate commission errors as it is easier to rectify commission error than omission error (Sithole and Vosselman, 2004). It is noteworthy that a bridge crossing roads in the middle of *Sample23* is correctly classified as non-ground, while Terrascan and upward-fusion regard them as part of
the terrain. The filtering comparison in Figure 6.14 suggests that the newly proposed PSGC approach has better performance across various types of landscapes.

Table 4.3 Comparison of filtering errors and kappa coefficients of PSGC, ALDPAT, Terrascan, and Upward-fusion for all sample sites.

<table>
<thead>
<tr>
<th>Sample No.</th>
<th>Error type</th>
<th>PSGC (%)</th>
<th>ALDPAT (%)</th>
<th>Terrascan (%)</th>
<th>Upward-fusion (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sample11</td>
<td>I</td>
<td>0.72</td>
<td>5.67</td>
<td>7.96</td>
<td>7.97</td>
</tr>
<tr>
<td></td>
<td>II</td>
<td>6.01</td>
<td>1.33</td>
<td>4.63</td>
<td>4.57</td>
</tr>
<tr>
<td></td>
<td>Total</td>
<td>1.83</td>
<td>1.98</td>
<td>3.52</td>
<td>3.51</td>
</tr>
<tr>
<td></td>
<td>Kappa</td>
<td>95.32</td>
<td>95.07</td>
<td>91.24</td>
<td>91.27</td>
</tr>
<tr>
<td>Sample12</td>
<td>I</td>
<td>0.42</td>
<td>4.28</td>
<td>6.53</td>
<td>16.76</td>
</tr>
<tr>
<td></td>
<td>II</td>
<td>5.90</td>
<td>0.81</td>
<td>3.16</td>
<td>3.36</td>
</tr>
<tr>
<td></td>
<td>Total</td>
<td>3.64</td>
<td>3.02</td>
<td>5.73</td>
<td>13.17</td>
</tr>
<tr>
<td></td>
<td>Kappa</td>
<td>92.62</td>
<td>93.76</td>
<td>88.18</td>
<td>72.20</td>
</tr>
<tr>
<td>Sample21</td>
<td>I</td>
<td>2.52</td>
<td>15.35</td>
<td>14.28</td>
<td>16.48</td>
</tr>
<tr>
<td></td>
<td>II</td>
<td>6.32</td>
<td>1.72</td>
<td>1.44</td>
<td>3.17</td>
</tr>
<tr>
<td></td>
<td>Total</td>
<td>5.44</td>
<td>12.17</td>
<td>11.12</td>
<td>13.87</td>
</tr>
<tr>
<td></td>
<td>Kappa</td>
<td>88.56</td>
<td>72.67</td>
<td>75.12</td>
<td>68.86</td>
</tr>
<tr>
<td>Sample22</td>
<td>I</td>
<td>1.08</td>
<td>9.39</td>
<td>11.39</td>
<td>18.90</td>
</tr>
<tr>
<td></td>
<td>II</td>
<td>5.82</td>
<td>2.61</td>
<td>0.57</td>
<td>2.13</td>
</tr>
<tr>
<td></td>
<td>Total</td>
<td>3.07</td>
<td>5.70</td>
<td>5.99</td>
<td>11.19</td>
</tr>
<tr>
<td></td>
<td>Kappa</td>
<td>93.76</td>
<td>88.56</td>
<td>88.03</td>
<td>77.81</td>
</tr>
<tr>
<td>Sample23</td>
<td>I</td>
<td>3.00</td>
<td>4.81</td>
<td>5.74</td>
<td>20.52</td>
</tr>
<tr>
<td></td>
<td>II</td>
<td>4.30</td>
<td>2.62</td>
<td>5.02</td>
<td>3.89</td>
</tr>
<tr>
<td></td>
<td>Total</td>
<td>3.00</td>
<td>3.11</td>
<td>4.46</td>
<td>11.86</td>
</tr>
<tr>
<td></td>
<td>Kappa</td>
<td>93.80</td>
<td>93.60</td>
<td>90.80</td>
<td>76.29</td>
</tr>
</tbody>
</table>
To evaluate the quality of extracted the DTMs, a total of 22 ground control points from *Eltham* were collected for quantitative assessment. Control points are manually identified and selected randomly. The distribution of control points is shown in Figure 4.15(a), where regions on ridge or near vegetation are challenging for filtering. Figure 4.15(b)-(e) show the errors of PSGC, ALDPAT, Terrascan and Upward-fusion, respectively. Table 4.4 is used to evaluate the quality of DTMs. ALDPAT and Terrascan have difficulty in discontinuous regions, as errors are larger than 0.5m in such areas. As the result, DTMs covering such regions fail to represent sufficient detail. Upward-fusion has the worst performance on this site since both average error (0.54m) and standard deviation (0.65m) are relatively large. This confirms that height difference measurement alone is insufficient to generate a detailed DTM if the terrain is complex. PSGC has good performance in those regions. As indicated in Table 4.4, most of the errors are less than 0.5m and the generated DTM is more accurate in terms of average error (0.25m) and standard deviation (0.27m). One limitation observed is that regions close to bushes are inaccurately extracted because the derived DTM includes many low-level object points.
Figure 4.15 Error distribution of generated DTM s: (a) distribution of ground control points, (b) accuracy assessment from PSGC, (c) accuracy assessment from ALDPAT, (d) accuracy assessment from Terrascan, (e) accuracy assessment from Upward-fusion.
Table 4.4 Error statistics from generated DTM s of PSGC, ALDPAT, Terrascan and upward-fusion.

<table>
<thead>
<tr>
<th></th>
<th>Number of points in the error range of 0-0.2 m</th>
<th>Number of points in the error range of 0.2-0.5 m</th>
<th>Number of points in the error range of 0.5-1 m</th>
<th>Number of points with error &gt;1 m</th>
</tr>
</thead>
<tbody>
<tr>
<td>PSGC</td>
<td>8</td>
<td>13</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>ALDPAT</td>
<td>7</td>
<td>9</td>
<td>4</td>
<td>2</td>
</tr>
<tr>
<td>Terrascan</td>
<td>6</td>
<td>9</td>
<td>5</td>
<td>2</td>
</tr>
<tr>
<td>Upward-fusion</td>
<td>2</td>
<td>5</td>
<td>11</td>
<td>4</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>Mean error</th>
<th>Standard deviation</th>
</tr>
</thead>
<tbody>
<tr>
<td>PSGC</td>
<td>0.25 m</td>
<td>0.27 m</td>
</tr>
<tr>
<td>ALDPAT</td>
<td>0.38 m</td>
<td>0.43 m</td>
</tr>
<tr>
<td>Terrascan</td>
<td>0.39 m</td>
<td>0.41 m</td>
</tr>
<tr>
<td>Upward-fusion</td>
<td>0.54 m</td>
<td>0.65 m</td>
</tr>
</tbody>
</table>

4.4 Chapter Summary

In this chapter, a newly developed progressive segmentation approach for the separation of terrain and non-terrain points has been presented. Closeness constraints are computed based on a preliminary DTM, and smoothness constraints are obtained from spatial structure. With a graph cut algorithm, the optimal solution for filtering terrain points is found, which is in contrast to other methods that largely rely on pre-defined thresholds. The standard graph cuts technique provides an optimal solution to segmentation in one-shot, while PSGC makes use of the intermediate segmentation result to progressively refine the preliminary terrain surface, thus influencing the potential labelling cost for the next segmentation iteration. This can significantly improve filtering accuracy in discontinuous areas.

The experiments reported also illustrate the better performance of PSGC as compared to the other three software packages, ALDPAT, Terrascan and Upward-fusion, in terms of type I error, total error and Kappa coefficient. Type II error is the major issue for PSGC, however the Type II error is only slightly higher than that yielded by the other filters. Moreover, such errors are easily fixed via manual inspection. It was also found that PSGC can effectively utilize both data and
smoothness constraints in the graph-based optimization to better eliminate errors on bridge surface.
Chapter 5 Building

Footprint Extraction
5.1 Introduction to Footprint Extraction

This chapter discusses the detection and extraction of building footprints. A building footprint, which is also referred to as a building outline, is an essential component in geospatial information systems. It not only defines a Region of Interest (ROI), but also reveals valuable information about the general shape of building roofs. Thus, the building footprint provides valuable information for a variety of applications, such as urban planning, virtual city tourism, pollution modelling, and disaster management. In addition, building footprint can be used as prior shape estimates in the modelling of more detailed rooftop structures at a complexity level of LOD2 (Vosselman, 2002).

This chapter first presents, in Section 5.2, the extraction of straight line segments and the detection of buildings with line segments. Section 5.3 describes a vertex-driven Douglas-Peucker method and the energy minimization scheme to determine the optimal polygon determined from initial boundaries. In Section 5.4, hybrid reconstruction in terms of explicit and implicit reconstruction through energy minimization is described in detail. Results of experimental testing with simulated data as well as real data are presented in Section 5.5. These demonstrate the potential of the proposed approach. A discussion on error analysis and the performance of the new method using high-resolution LiDAR data with complex building shapes will also be presented. A chapter summary is offered in Section 5.6.

5.2 Building Detection

5.2.1 Extraction of Straight Line Segments

Building detection is the first step in the footprint extraction. As discussed in Section 3.3, straight line segments have the potential to not only reduce the data volume, but also to preserve reliable geometric information that is useful for building detection. Therefore, the straight line segment is employed as an intermediate object for building detection.

To extract line segments efficiently, the whole LiDAR data is firstly divided into several scan lines. Modern LiDAR systems tag points at the end of each scan line, and these end points can be employed as flags to define intervals, where each scan line can be viewed as a profile of the landscape.
Treating successive points in each scan as a polyline, the well-known Iterative-End-Point-Fit simplification algorithm is adopted to extract straight line segments. This has been described in Nguyen et al. (2007). The simplification algorithm is based on a crude simplification scheme that recursively detect key vertices according to a point-to-edge distance tolerance. The algorithm starts with a single edge by simply connecting the first and last points. Then the edge is iteratively split, based on the furthest perpendicular point-to-line distance until all point-to-edge distances are small. Once all key vertices are detected, the raw polyline can be simplified as a polyline which only has connecting keys. Figure 5.1 demonstrates the extraction process of straight line segments. For illustration purposes, the extraction result is shown by coding raw points with different colours. Input points are first discretised into scan lines, as shown in Figure 5.1(b). After solving the polyline simplification problem, each scan line is divided into a set of segments, as illustrated in Figure 5.1(c). A close-up of a raw scan line and the segmented result are shown in Figure 5.1(d).

![Figure 5.1](image-url)

**Figure 5.1** A demonstration of the line extraction algorithm: (a) input LiDAR point cloud, (b) the separated scan lines, (c) extracted line segments are rendered in different colour and (d) close-up of a raw scan line and segmentation result.
5.2.2 Line-based Building Detection

Note that the extracted line segments are from all landscape surfaces, mainly consisting of terrain, vegetation and buildings. To deal with the building detection problem, this research extends a previously developed rule-based classifier (He et al., 2013) by adopting an energy-based scheme using the characteristics from line segments and pair-wise interaction. This work first excludes the line segments near to the terrain surface because buildings should have certain height above the ground. After that, the detection approach discriminates the characteristics of segments on vegetation surfaces. It is reasonable to assume that a line segment lying on a vegetation surface ($f_x = vegetation$) is more likely to have a short length because the vegetation crown often exhibits large height variance. On the other hand, a line segment from a rooftop ($f_x = building$) is often of greater length due to height consistency. This approach checks the sizes of all non-terrain segments, where ‘size’ expresses the number of points belonging to a segment, and assigns only segments with small size to vegetation, while classify the remaining segments as non-vegetation. This approach formulates a data characteristic function as $E_{data}(f_{x_i})$ for labelling $f \in \{vegetation, building\}$ and assigns a labelling penalty ($0 < \alpha < 1$) for disobediences as:

\[
E_{data}(f_{x_i}) = \begin{cases} 
\alpha & \text{if } f_{x_i} = building \text{ and } size(x_i) < \sigma \\
\alpha & \text{if } f_{x_i} = vegetation \text{ and } size(x_i) > \sigma \\
0 & \text{otherwise}
\end{cases} \quad (5.1)
\]

where $f_{x_i}$ is the label of line segment $x_i$.

Ambiguities can be found for small components of the building, such as dormers and chimneys. Since these components have relatively small segment size, they are likely to classified as vegetation. However, these objects exhibit strong spatial coherence to encourage the object to have the same class label with its surroundings. Therefore the prior energy term ($E_{prior}$) over the Markov Random Field (MRF) is conducted as a smoothness characteristic function, where the penalty is assigned to a pair of neighbouring line segments if their labelling is different. The pair-wise interaction is formulated as:

\[
E_{prior}(f_{x_i}, f_{x_j}) = \begin{cases} 
1 & \text{if } f_{x_i} \neq f_{x_j} \\
0 & \text{if } f_{x_i} = f_{x_j}
\end{cases} \quad (5.2)
\]
Integration of the data term and the prior term into the global labelling energy $E$ lead to:

$$E(f) = \sum_{x_i \in X} E_{\text{data}}(f_{x_i}) + \lambda \sum_{(x_i, x_j) \in \mathcal{N}} E_{\text{prior}}(f_{x_i}, f_{x_j})$$  \hspace{1cm} (5.3)$$

where $X$ is the collection of non-terrain line segments and $\mathcal{N}$ is the set of neighbouring line segment pairs. Adopted here in the method proposed by Zhang and Faugeras (2004) to define the line segment neighbourhood. Particularly, line segments falling in a cylindrical space with radius $r$ and its axis coinciding with the direction of the query line segment are defined as neighbours of the segment.

With the energy minimization problem being solved using the well-known graph-cut method (Boykov and Jolly, 2001), the class label of each point is the same as that of the corresponding line segment. The improvement compared with conventional point-based detection approach can be observed near the building edges in an example shown in Figure 5.2 as points on building edges are aggregated in advance by line-based detection.

![Figure 5.2 Results for building detection: result from point-based classification (left), and result from line-based classification (right).](image)

### 5.2.3 Initial Building Boundary Determination

To further represent individual buildings, a distance-based clustering algorithm is applied to building points. The TIN graph is first created among building points to represent the data structure, where long edges are eliminated to cut off connections of different patches. Consequently, connected-component labelling is performed on
the undirected graph to isolate building patches. Small patches (<6m²) are recognized as detection error and removed from consideration.

While points near to the outline of buildings are reserved to derive footprint shape, inside points are considered as redundant observations and should be eliminated. The 2D α-shape algorithm (Bernardini and Bajaj, 1997) is employed to delineate initial boundaries from grouped points, where the value of α is often defined as 1.5 times the average point spacing. Figure 5.3 shows the grouping result of building patches and obtained initial boundaries.

![Figure 5.3 Extracted building patches (left) and initial boundaries (right).](image)

### 5.3 Building Footprint Extraction

#### 5.3.1 Boundary Simplification

The obtained initial boundaries usually exhibit a zigzag pattern along the derived outlines, as shown in the close-up view of Figure 5.4(a). To provide a meaningful and compact description of the boundary, polygon simplification is critical to preserve relevant edges. Rather than using the original Douglas-Peucker (DP) algorithm (Douglas and Peucker, 1973) for polygon simplification (Jwa et al., 2008; Kim and Shan, 2011b; Weidner and Förstner, 1995), a Vertex-driven Douglas-Peucker (VDP) algorithm is proposed to simplify a polygon from its initial boundary. The main difference between the two algorithms is that VDP focuses on polygonal complexity while the original DP considers data deviation. In VDP, a number of key points, denoted as \( n \), is required to generate a polygonal hypothesis, and the optimal value of \( n \) is determined through energy minimization in the sequential optimal polygon selection process. This is in contrast to the original DP algorithm which
Chapter 5 Building Footprint Extraction

relies on a pre-defined point-to-edge distance which is difficult to optimize. Obviously, when the number of the key points is defined as two, the polygonal approximation will appear as a straight line segment which links the farthest data points to minimize data distortion. With $n$ increasing, the polygon expands by iteratively adding the point with the highest point-to-edge distance to the current form polygon. Figure 5.4(b)-(e) shows the shapes of various polygonal hypotheses with different $n$. It is clear that the lower the degree of simplification, the lower the amount of data fitting error, but the higher the complexity of the polygon.

Figure 5.4 Polygonal hypotheses with different numbers of vertices.
5.3.2 Optimal Polygon Selection

In order to select the optimal polygon among the different hypotheses, the energy function is adopted to globally present the combination of the overall cost of data and the model. According to the general form of the energy function in Equation 3.1, the data energy depends on the capability of describing data $D$ using a polygon $P$. Thus, the data energy is defined as the residual error of each point towards the hypothesis polygon. Since a building footprint can be represented as a simple polygon, the complexity of the polygon can be employed as a model constraint. The complexity of a polygon model can be estimated from LiDAR points along the building boundary. As the vertices of the polygon are a subset of the boundary points, a sequence of coordinate $(X, Y)$ values encodes the polygonal complexity. To simplify computation, a 2D bounding box enclosing the polygon is employed to indicate coordinate range.

In this formulation, the global energy function, $GE(D, P)$, is defined as a combination of $E(D|P)$, which encodes the energy of data $D$ over the polygon $P$, and the energy of $P$ itself. The global energy function $GE(D, P)$ can be expressed as

$$GE(D, P) = \frac{\alpha}{2n^2} + \lambda_1(n\log_2 A)$$

(5.4)

where $\Omega$ is the sum of the squared residuals, $n$ is the number of polygon vertices (defined in VDP) and $A$ is the area of the data bounding box, which represents the coordinate range. $\lambda_1$ is introduced to balance data and prior terms, and its value is usually set to 1, which is applicable in most cases.

The best polygon representing the building footprint can be found when the overall energy is minimized. This can be done by iteratively calculating and comparing the total energy with $n$ in the range of $[3, n_D]$, where $n_D$ is the number of data points. Taking the boundary in Figure 5.4 as an example, the data energy term $(E(D|P))$, the prior energy term $(E(P))$ and the total energy term $GE(D, P)$ of each $n$ are shown in Figure 5.5. $E(D|P)$ drops swiftly when $n$ increases from 4 to 8 and closes to 0 after $n = 9$ while $E(P)$ grows linearly with increasing $n$. From the $GE(D, P)$, the polygon with $n = 8$ has the minimum total energy and thus is the corresponding polygon of the best simplification of the building footprint.
Figure 5.5 Energies for different vertex numbers.

5.4 Hybrid Reconstruction

5.4.1 Explicit Reconstruction

It can be observed that the preliminary boundary is quite irregular because key points are the subset of raw boundary points. Therefore, a regularization step is often necessary to enhance the geometric shape of building footprint through imposition of parallelism and perpendicularity constraints, for example.

A local adjustment is first applied on each edge to approximate the real direction. Let $K = \{k_1, k_2, ..., k_n\}$ be the key points of the preliminary polygon. The adjusted line $l_{k_1,k_2}$ is determined by a RANSAC line fitting method using the LiDAR points between $k_1$ and $k_2$. An example is shown in Figure 5.4(a). Note that angular difference before and after adjustment may represent a dramatic difference for some edges. Therefore, edges with large direction difference from the preliminary polygon, or with small length, are eliminated. Thereafter, this approach explores the direction relationship among line segments. If the angular difference between the potential line and the longest line is close to 0° or 90°, then the regularity relationship is built as parallelism or perpendicularity.
If direction alignment is directly applied to the longest line, the quality of the dominant direction is mainly dependent upon the local fitting of the longest line. Therefore, global adjustment is applied to find the precise dominant direction as well as the optimal parameters for each edge. The data fitting error ($DFE$) is used to measure the deviation of points from the fitted line segments, and it is expressed as the accumulation of squared Euclidean distances between each point and the corresponding segment line (Fisher, 2004):

$$ DFE = \sum_{l \in L} \sum_{p_j \in l_i} w_{p_j} d^2(p_j, l_i) $$

(5.5)

Here, $l_i$ is one of the remaining line segments $L$ and $p_j$ is an observation on $l_i$. $d^2(p_j, l_i)$ represents the squared distance of a point $p_j$ to its corresponding line, which has an associated measurement uncertainty weight $w_{p_j}$ (Kanatani, 2008). $w_{p_j}$ is set to 1 if the line segment has the direction relationship with the longest line segment, and to zero otherwise.

In order to minimize $DFE$ and meanwhile maintain direction relationships, an error minimization optimization approach is employed. As distinct from the energy function defined in Equation 5.1, where the energy is an accumulation of soft constraints from data and from prior knowledge, the direction relations are considered as equal hard constraints. The objective error function with equal constraints is defined as

$$ \min(DFE), \ c \in C_\theta $$

(5.6)

where $c$ is the direction constraint. $C_\theta$ includes parallelism and perpendicularity constraints. The well-known non-linear convex point approach (Fisher, 2004) is applied to solve the optimization problem. The regularized result of the local fitting of building edges in Figure 5.4(a) is shown in Figure 5.4(b). It can be seen that the line segments in Figure 5.4(b) better represent the shape of the building footprint than those in Figure 5.4(a).
5.4.2 Implicit Reconstruction

As shown in the previous section, short line segments are eliminated in the explicit reconstruction. The fitted short line segments usually contain only a few LiDAR points, therefore they are not reliable to represent a building edge, particularly when the density of the LiDAR points is not high. The uneven distribution of points due to an irregular scanning pattern makes the situation even worse. Short segments usually link neighbouring long segments. They are referred to as connectors in the following discussion and they will again be reconstructed using energy minimization. However, since robust features cannot be reliably extracted from the limited observations, the connectors need to be reconstructed implicitly and a different definition of energy is necessary. Three types of connector hypotheses are categorized as follows (see also in Figure 5.7):

a. No additional line: two adjusted line segments are directly connected by a line extension to bridge the gap (Figure 5.7a).

b. One additional line: use one line to intersect two fixed line segments. The additional line is defined by one point and a floating direction (Figure 5.7b).

c. Two additional lines: use two rays to intersect two fixed line segments. The two rays are defined at one point with two floating directions (Figure 5.7c)
The optimal connector model is selected by energy minimization. Unlike the global energy for polygon selection (Section 5.3.2), model selection is performed locally on each gap and the data observations are the points between the two adjacent lines. The data energy is defined as the residuals describing the deviation of boundary points from the model. A low residual implies more agreement with the hypothesis model. To describe the prior energy, both shape complexity and smoothness are encoded. Thus, the model energy is extended by adding two more smoothness constraints: (1) length of additional line (favouring short length), and (2) angle transition (preferring right angle). The total energy can be further expressed as

\[
LE(D, P) = \frac{a}{2ln^2} + \lambda_2(Nlog_2A + S'log_2S + \sum t_{\angle\theta}log_2N)
\]  \hspace{1cm} (5.7)

where \(A\) is the area of the bounding box from local observation data; \(S\) the extended length of the two line segments; \(N\) the number of new added vertices; \(S'\) the length of the connector model; \(t_{\angle\theta}\) the angle penalty, where \(t_{\angle\theta} = 0\) if \(\theta = 90^\circ\) and \(t_{\angle\theta} = 1\) otherwise; and \(\lambda_2\) the weight coefficient to trade-off between the data term and model term.

The result of the connection procedure onto the discretised line segments of Figure 5.6(b) is shown in Figure 5.8. It can be seen that the gaps are closed, and the obtained polygon well represents the building footprint.
Chapter 5 Building Footprint Extraction

5.5 Experiments

This section reports on a performance evaluation of the developed algorithms for building footprint extraction using energy minimization-based hybrid reconstruction. The criteria for quantitative assessment of performance are first introduced. Simulated data with various noise levels is initially used to illustrate the robustness of the developed approach as compared to the original DP method for boundary simplification and explicit reconstruction for building footprint extraction. Following this, the results from a real dataset over a large area are presented. The performance is quantitatively evaluated using manually extracted precise reference data.

5.5.1 Criteria for Quality Assessment

Five evaluation metrics, defined as follows, are used for the quantitative evaluation of performance:

- **Coverage Error (CE)** evaluates how completely the approach is able to represent the ground-truth polygon. Coverage percentage is expressed as

  \[
  CE = \frac{\text{total area of difference}}{\text{total area of reference polygon}} \times 100\% \tag{5.8}
  \]

  where the difference analysis between two polygons is based on the Vatti clipping algorithm (Vatti, 1992), implemented in the General Polygon Clipper (GPC) software library (http://www.cs.man.ac.uk/~toby/alan/software)

Figure 5.8 Watertight building footprint.
• **Root Mean Square Error** (RMSE) predicts the deviation of the footprint to the data. Let \( n_D \) be the number of boundary points and \( \Omega \) be the sum of squared residuals accumulated from all boundary points, then the RMSE is defined as

\[
RMSE = \sqrt{\frac{1}{n_D} \Omega}
\]  
(5.9)

• **Direction Difference** (DD) measures the difference of the principal orientation between the extracted polygon and the ground-truth polygon. Given the dominant direction of an estimated polygon \( \hat{d} \) and the corresponding ground-truth dominant direction \( d \), DD is calculated as

\[
DD = \cos(\hat{d}^T \cdot d)
\]  
(5.10)

• **Model Complexity Difference** (MCD) aims to reflect the complexity difference between the derived footprint and the ground-truth. Let \( n \) and \( n_r \) be the number of vertices of the derived polygon and reference polygon, respectively, then MCD is expressed as

\[
MCD = \sqrt{\left(\frac{n-n_r}{n_r}\right)^2}
\]  
(5.11)

• **Vertex Difference** (VD) evaluates the likelihood between the derived footprint and the ground-truth. Let \( r \) be the residual distance of a vertex of the derived polygon to the nearest vertex of the reference polygon, then VD is defined as

\[
VD = \sqrt{\frac{1}{n} \sum_{i}^{n} r_i^2}
\]  
(5.12)

### 5.5.2 Performance on Simulated Data

The simulated data subjected to testing is designed to evaluate the developed VDP approach for polygon simplification and extraction of relevant edges from noise-influenced boundary points. Gaussian noise with five noise levels (\( \sigma = 0.05, 0.1, 0.15, 0.2, 0.25m \)) was added to raw boundary points. Both VDP and conventional DP algorithms were applied. Furthermore, performance was evaluated for explicit reconstruction with both the original DP method (Kim and Shan, 2011b) and the hybrid reconstruction with newly developed VDP presented here were evaluated.
Figure 5.9 Comparison of performance under various Gaussian noise levels.
Figure 5.9 illustrates the results of DP and VDP for polygonal approximation (blue lines). The results of DP and VDP are quite similar when the noise level is low. However, DP presents an over-fitting of the preliminary boundary when the noise level is high. This is because DP only relies on the local point-to-edge distance to locate the key points. As seen in the DP result with $\sigma = 0.25$, noisy vertices were not eliminated in the left part of the figure. With the preliminary polygon, explicit reconstruction generates an irregular footprint since the rule based on angle difference does not function well on noisy boundaries. On the other hand, VDP considers the global data deviation as well as model complexity to control the key point selection. A key point is added to the preliminary polygon only when the alleviation of data distortion is larger than the cost of introduction of a new vertex. Consequently, the hybrid reconstruction based on VDP achieves a regular polygon shape.

The metric evaluation in Table 5.1 indicates that the hybrid method has a low direction difference as the dominant direction is measured by global optimization rather than via local adjustment. MCD for both methods increased at a higher level of noise. Explicit reconstruction exhibits an overestimation result, while hybrid reconstruction shows an underestimation. However, the model complexity of the hybrid method is closer to the ground truth. In addition, much smaller values of the vertex difference achieved from VDP approximation and hybrid reconstruction suggest better performance than the explicit method.

Table 5.1 Statistical assessment under various Gaussian noise levels.

<table>
<thead>
<tr>
<th>$\sigma$ [m]</th>
<th>CE [%]</th>
<th>RMSE [m]</th>
<th>DD [rad]</th>
<th>MCD</th>
<th>VD [m]</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.05</td>
<td>DP</td>
<td>1</td>
<td>0.25</td>
<td>0.04</td>
<td>0.18</td>
</tr>
<tr>
<td></td>
<td>VDP</td>
<td>2</td>
<td>0.24</td>
<td>0.02</td>
<td>0</td>
</tr>
<tr>
<td>0.10</td>
<td>DP</td>
<td>3</td>
<td>0.31</td>
<td>0.16</td>
<td>0.18</td>
</tr>
<tr>
<td></td>
<td>VDP</td>
<td>4</td>
<td>0.25</td>
<td>0.02</td>
<td>0</td>
</tr>
<tr>
<td>0.15</td>
<td>DP</td>
<td>8</td>
<td>0.39</td>
<td>0.12</td>
<td>0.09</td>
</tr>
<tr>
<td></td>
<td>VDP</td>
<td>9</td>
<td>0.38</td>
<td>0.03</td>
<td>0.09</td>
</tr>
<tr>
<td>0.20</td>
<td>DP</td>
<td>12</td>
<td>0.54</td>
<td>0.21</td>
<td>0.45</td>
</tr>
<tr>
<td></td>
<td>VDP</td>
<td>11</td>
<td>0.44</td>
<td>0.07</td>
<td>0.18</td>
</tr>
<tr>
<td>0.25</td>
<td>DP</td>
<td>16</td>
<td>0.59</td>
<td>0.32</td>
<td>0.73</td>
</tr>
<tr>
<td></td>
<td>VDP</td>
<td>14</td>
<td>0.53</td>
<td>0.12</td>
<td>0.27</td>
</tr>
</tbody>
</table>
5.5.3 Performance on Real Data

Experiments have been conducted over the suburban area of Eltham, Victoria, Australia. This site, with well vegetated rolling terrain, is located northeast of the Melbourne CBD. The LiDAR data was collected by an Optech Gemini scanner in 2012 with an average point spacing of ~0.55m. In addition, aerial imagery was available for this site and this was employed as an independent data source in the evaluation.

The developed approach was applied to the whole dataset and the results of a portion containing 17 buildings, whose reference data were available, are presented in Figure 5.10. The detected building regions shown in Figure 5.10(a) are indicated by black lines overlaid on the hillshaded DSM. The small patches on dense tree crowns, highlighted by red squares, are falsely detected building regions. These are subsequently removed by an area-based filtering algorithm. All 17 building regions were detected. The results of preliminary extracted polygons and the final derived footprints obtained using the new developed approach are represented in blue and orange, respectively, in Figure 5.10(b). The reference footprints were generated by manual delineation with the assistance of aerial photography. These are shown by the green polygons in Figure 5.10(c).

Table 5.2 summarised the assessment result for the reconstruction of the 17 buildings. As indicated in the table, the coverage error is relatively high when rooftops are partially occluded by trees (e.g. b11, b16 and b17). Consequently, the vertex difference of these buildings is also relatively high. Another factor contributing to the vertex difference is the model complexity difference. The MCD is caused by either overestimation (e.g. b5 and b9) or underestimation (e.g. b7 and b12). The reconstructed footprints fit the data quite well as the RMSE is small, in the range of 0.18-0.77m. The largest RMSE is from b7 because a small protrusion in the right side is not reconstructed. For direction difference, all reconstruction results show a small deviation to the corresponding reference footprints. The small standard deviations of all evaluation criteria indicate that the proposed approach has a reliable and consistent performance.
Figure 5.10 Building footprint extraction applied to 17 buildings in the Eltham dataset: (a) detected building regions, (b) extracted footprints and (c) ground truth.
Table 5.2 Statistical assessment of building footprint extraction.

<table>
<thead>
<tr>
<th>Building#</th>
<th>CE [%]</th>
<th>RMSE [m]</th>
<th>DD [rad]</th>
<th>MCD</th>
<th>VD [m]</th>
</tr>
</thead>
<tbody>
<tr>
<td>b1</td>
<td>0.13</td>
<td>0.21</td>
<td>0.02</td>
<td>0</td>
<td>0.29</td>
</tr>
<tr>
<td>b2</td>
<td>0.01</td>
<td>0.18</td>
<td>0.01</td>
<td>0</td>
<td>0.48</td>
</tr>
<tr>
<td>b3</td>
<td>0.14</td>
<td>0.53</td>
<td>0</td>
<td>0</td>
<td>0.43</td>
</tr>
<tr>
<td>b4</td>
<td>0.03</td>
<td>0.53</td>
<td>0.03</td>
<td>0</td>
<td>0.38</td>
</tr>
<tr>
<td>b5</td>
<td>0.18</td>
<td>0.63</td>
<td>0.02</td>
<td>0.22</td>
<td>1.94</td>
</tr>
<tr>
<td>b6</td>
<td>0.1</td>
<td>0.49</td>
<td>0.02</td>
<td>0</td>
<td>0.57</td>
</tr>
<tr>
<td>b7</td>
<td>0.1</td>
<td>0.77</td>
<td>0</td>
<td>0.29</td>
<td>0.77</td>
</tr>
<tr>
<td>b8</td>
<td>0.14</td>
<td>0.48</td>
<td>0.03</td>
<td>0.09</td>
<td>0.54</td>
</tr>
<tr>
<td>b9</td>
<td>0.18</td>
<td>0.33</td>
<td>0.03</td>
<td>0.14</td>
<td>3.69</td>
</tr>
<tr>
<td>b10</td>
<td>0.04</td>
<td>0.38</td>
<td>0.01</td>
<td>0.27</td>
<td>0.79</td>
</tr>
<tr>
<td>b11</td>
<td>0.24</td>
<td>0.47</td>
<td>0.06</td>
<td>0</td>
<td>1.16</td>
</tr>
<tr>
<td>b12</td>
<td>0.11</td>
<td>0.21</td>
<td>0.05</td>
<td>0</td>
<td>1.13</td>
</tr>
<tr>
<td>b13</td>
<td>0</td>
<td>0.35</td>
<td>0.07</td>
<td>0</td>
<td>0.54</td>
</tr>
<tr>
<td>b14</td>
<td>0.05</td>
<td>0.56</td>
<td>0.09</td>
<td>0.22</td>
<td>0.54</td>
</tr>
<tr>
<td>b15</td>
<td>0.01</td>
<td>0.3</td>
<td>0.02</td>
<td>0</td>
<td>0.31</td>
</tr>
<tr>
<td>b16</td>
<td>0.24</td>
<td>0.38</td>
<td>0.03</td>
<td>0</td>
<td>0.51</td>
</tr>
<tr>
<td>b17</td>
<td>0.22</td>
<td>0.42</td>
<td>0.01</td>
<td>0.29</td>
<td>0.56</td>
</tr>
<tr>
<td>Average</td>
<td>0.11</td>
<td>0.42</td>
<td>0.03</td>
<td>0.09</td>
<td>0.86</td>
</tr>
<tr>
<td>Standard deviation</td>
<td>0.08</td>
<td>0.15</td>
<td>0.02</td>
<td>0.12</td>
<td>0.81</td>
</tr>
</tbody>
</table>

The details of extracted boundaries are shown in Figure 5.11. The regularized building footprints fit to the LiDAR data and reflect the detailed shapes. Using aerial imagery as a reference, it is noteworthy that most of the features are extracted correctly. Small building parts, as indicated in Circle 1 in the figure, are well detected and reconstructed, presenting sufficient model detail. The regularized footprint may miss important features due to the limited resolution of the LiDAR data. As shown in Circle 2, some corner features are not detected. The facts can be present in the results, these being due to obstructions from vegetation. An example is indicated in Circle 3 of Figure 5.11.
Figure 5.11 Experiments and quality assessments on several building footprints; DSM on the left, aerial image on the right.

5.5.4 Discussion

Several factors have an impact on the results of building footprint reconstruction. The proposed, newly developed approach comprises a series of processing steps and each has an impact on the subsequent processes, and thus also on the final results. Furthermore, the point density within LiDAR point clouds, as well as the sensor scanning pattern directly influence the quality of the results. An example is given in Figure 5.12, which highlights differences between the boundary of a detected building area, the reconstructed footprint and the ground-truth of building b9 (c.f. Figure 5.10). The error is caused by misclassification of a low-level building component at the lower left corner as terrain and this error propagates to the boundary of the building outlines, resulting in an incorrect footprint.
A further challenge is the selection of a proper weight parameter $\lambda_2$ in Equation 5.7. In the current implementation, the weight is determined manually, by trial and error from the reference data, and then a constant $\lambda_2 = 0.5$ is used for other buildings. However, a constant value is not always effective and this often leads to an incorrect generation of short line segments in the implicit reconstruction. As shown by the red circle in Figure 5.13, data term plays a key role in such an area due to the irregular scanning pattern. Therefore, an adaptive $\lambda_2$ needs to be determined based on the local conditions.
Nevertheless, these errors can be largely avoided through an increase in the point density of the LiDAR data. An example is given in Figure 5.14 over Mansfiled, Victoria, Australia. The point spacing is around 0.23m and the point density around 4 times that of the data shown in Figure 5.10. It is clear that the building footprints are correctly reconstructed, despite their complex structure and varying size. Even substructures with extremely short boundary segments are accurately reconstructed and modelled.

Figure 5.14 Footprint reconstruction of complex building shapes with high density LiDAR data.
5.6 Chapter Summary

In this chapter, several approaches for the extraction of building footprints from non-terrain points have been presented. With extracted line segments, the segments on building regions are first recognized. Preliminary polygons are then computed from the boundary points. Unlike utilisation of a pre-defined threshold for polygonal simplification, a *Vertex-driven Douglas-Peucker* (VDP) method has been proposed to improve performance. To enforce shape regularity, different forms of energy minimization are formulated among various hypotheses, and to bridge gaps between consecutive line segments through optimal connectors.

This newly developed approach has been experimentally tested and evaluated with both simulated and real data over large urban test areas. Quantitative assessment of the resulting building footprints against accurate reference data, using various quality criteria, has shown that the developed approach displays a high level of robustness and reliability. The experiments conducted have also shown that even better performance can be achieved with LiDAR data of a higher point density (say >5 pts/m²).
Chapter 6 Building

Rooftop Modelling
6.1 Introduction to Roof Modelling

The extracting and reconstruction of roof shapes is essential for the generation of rooftop geometries within a complexity level of LOD2. The detailed rooftop models enrich geometric description and contribute to better understanding of the urban environment. Typical applications lie in a diverse array of purposes including solar collection and architecture. In the previous chapter, only LiDAR data on building outlines was employed to extract building footprints. To reconstruct a more detailed rooftop model, LiDAR data inside of the footprint is further processed to reflect individual roof facet. This process is also referred to as segmentation. Due to the widespread existence of planar structures in building rooftops, extracting planar patches from LiDAR data is an essential step. Thereafter, boundary determination for the planar patches provides internal shape information to represent the polyhedral building model.

All the segmentation methods reviewed in Section 2.3.2 employ local features only, while the proposed approach developed this research aims to find an optimal partition solution according to global similarities determined through spectral clustering. The principle of spectral clustering and its extension of k-way partitioning are first introduced in Section 6.2. The proposed planar segmentation approach using spectral clustering is then presented, in Section 6.3. Section 6.4 describes a reconstruction method involving both topological constraints and global regularities, which is initialled following roof segmentation. Section 6.5 discusses the results of experimental test and evaluates the performance of the new method. A chapter summary is provided in Section 6.6.

6.2 Spectral Clustering

6.2.1 Principle of Spectral Clustering

Spectral clustering was first suggested by Donath and Hoffman (1973) to construct graph partitions using the weighted adjacency matrix and its eigenvectors. This has been successfully applied for a range of computer vision applications, such as image motion (Yan and Pollefeys, 2006) and mesh smoothness segmentation (Liu and Zhang, 2004). Spectral clustering is based on the concept of spectral graph theory (Chung, 1997). The main idea is to use matrix theory and principal component
analysis to investigate the main properties of the weighted matrix and the Laplacian matrix of the graph, which give rise to a more powerful data representation that preserves angle distortion in feature space. Unlike many traditional clustering algorithms based on convex spherical sample space, spectral clustering views the data clustering as a graph partitioning problem without make any assumptions about the form of the data clusters. As a result, spectral clustering often outperforms traditional clustering algorithms, such as $k$-means clustering (von Luxburg, 2007).

Let a data point be a vertex of a graph. The set of all points in their feature space can then be represented as a weighted undirected graph $G = (V, E, W)$. Here $V = \{v_1, \ldots, v_N\}$ is the set of vertices and $E$ is the set of edges. The non-negative weight ($w_{ij} > 0$) on each edge is a function of the similarity between vertices $v_i$ and $v_j$. When constructing the graph the goal is to model the neighbourhood relationships among the data points. As introduced by von Luxburg (2007), there are three popular constructions to transform a set of data points with pairwise similarities into a graph.

In the $\epsilon$-neighbourhood graph, all points whose pairwise distances are smaller than $\epsilon$ are connected. In the $k$-nearest neighbour graph, two vertices $v_i$ and $v_j$ is connected when $v_j$ is among the $k$-nearest neighbours of $v_i$. The fully connected graph simply connects all points with positive similarity with each other.

In the case of image segmentation (Shi and Malik, 2000), the weight $w_{ij}$ on each graph edge is defined by considering both the optical properties and spatial location:

$$w_{ij} = \exp\left(-\frac{\|F_i - F_j\|^2}{\sigma_i^2}\right) \times \begin{cases} \exp\left(-\frac{\|L_i - L_j\|^2}{\sigma_L^2}\right) & \text{if } \|L_i - L_j\|_2 < R \\ 0 & \text{otherwise} \end{cases}$$

(6.1)

where $F_i$ is a feature vector from pixel intensity, colour or texture information, and $L_i$ is the spatial location of the pixel. In addition, $R$ controls the neighbour relationship on the pixel to be considered in the graph $G$.

For data given in the form of a weighted adjacency graph, the data clustering problem can be transformed to a graph partition problem. To be more specific, a partition of the graph is sought such that the edges between different groups have a rather low weight, which means that points in different clusters are dissimilar from each other. Meanwhile, the edges within a group have high weight, which also means that points within the same cluster are similar to each other.
Taking binary partitioning of a graph as an example, $A$ and $\bar{A}$ are two disjoint subsets of graph $G$, subject to $A \cap \bar{A} = \emptyset$ and $A \cup \bar{A} = V$. From a graph partitioning point of view, the simplest way to partition into two sub-graphs is to solve the minimum cut problem (Wu and Leahy, 1993). However, in practice the minimum cut often tends to generate unbalanced solutions, i.e., a rather small sub-graph is cut away from the rest of the graph. Hence, more advanced constraints are introduced to track this weakness. The Normalized cut (Ncut) is one of the most widely used object functions. It explicitly requests that the two sets $A$ and $\bar{A}$ have reasonably large size by adding a normalized term, which can be expressed as

$$Ncut(A, \bar{A}) = \frac{cut(A, \bar{A})}{assoc(A, V)} + \frac{cut(A, \bar{A})}{assoc(\bar{A}, V)}$$

(6.2)

where $cut(A, \bar{A}) = \sum_{i \in A, j \in A} w_{ij}$ is the degree of dissimilarity on the cut. $assoc(A, V) = \sum_{i \in A, j \in V} w_{ij}$ represents the total connection from nodes in $A$ to all nodes in the graph. In other words, the Ncut criterion seeks a partition that maximizes the association (normalized term) inside each group for a better balance.

Unlike the graph cuts approach presented in Section 4.2, spectral clustering separates data into several groups according to their similarities. The objective is to find a partition of the graph so that the edges within a group have high similarity while the edges between different groups have very low weight. This means that the graph only contains $n$-links for measuring their similarities. One more challenge is that the number of groups is often unknown while the graph cuts approach in Section 4.2 assumes points are affiliated with foreground or background. Therefore, the graph cuts approach is not suited to planar patch extraction. A prevalent strategy to solve this problem in computer vision is relaxation and rounding. First, relaxation gives an alternative description of the clustering task. By keeping a small number of eigenvectors with small eigenvalues, relaxation leads to a lower-bound data representation for the objective function of the graph partitioning problem. Founding maps the continuous eigenvectors to discrete solutions, as close as possible to the new data representation. The first stage is usually related to principal component analysis and spectral embedding whilst the second stage essentially involves an auxiliary clustering problem based on the spectral embedding. Shi and Malik (2000) proved that the minimization of the two-way Ncut problem can be relaxed to the
Rayleigh quotient. To solve the relaxation problem, it is necessary to solve the generalized eigenvalue system:

\[ Lz = \lambda z, \quad L = D^{-\frac{1}{2}}(D - W)D^{-\frac{1}{2}} \]  \hspace{1cm} (6.3)

where \( L \) is the Normalized Laplacian matrix, which is symmetric positive semi-definite. The degree matrix \( D = \sum_{j=1}^{N} w_{ij} \) is a diagonal matrix and its elements are the degrees of the nodes of \( G \). Let \( \lambda_i \) and \( z_i \) be the \( i \)-th eigenvalue and eigenvector of the matrix \( L \), respectively. According to the Raleigh-Ritz theorem, \( z_1 = D^{-1/2}1 \) is an eigenvector of Equation 6.3 with an eigenvalue of 0, and \( z_2 = \arg\min_z z^T L_{nn} z \) is the second eigenvector to the relaxed. Hence, the splitting at value 0 on \( z_2 \) is equivalent to the solution of a minimized cut.

### 6.2.2 \( K \)-way Partitioning of Spectral Clustering

This subsection considers a more general case in which there exist multiple subgraphs in one dataset. \( K \)-way partitioning can be divided into two major groups: recursive two-way partitioning and direct multi-way partitioning (Ng et al., 2002).

Recursive two-way partitioning belongs to the hierarchical clustering method, which iteratively employs the eigenvector with the second smallest eigenvalue to bipartition the sub-graph by finding a splitting edge such that the cut is minimized. Subsequently, a decision is made whether the current partition should be subdivided by checking the stability of the cut. The number of groups segmented by this method can be controlled directly by the maximum allowed \( k \).

Direct multi-way partitioning uses spectral embedding for simultaneous \( k \)-way clustering. Suppose that a symmetric weight matrix \( W \in \mathbb{R}^{n \times n} \) encodes the pairwise similarity, where \( w_{ij} \in [0,1] \). In order to obtain a symmetric normalized weight matrix, the normalized graph Laplacian is calculated as

\[ L_{sym} = D^{-1/2}W D^{-1/2} \]  \hspace{1cm} (6.4)

such that \( L_{ij} = L_{ji} = w_{ij} / \sqrt{D_{ii}D_{jj}} \). Let \( z_1, z_2, ..., z_n \) be the orthonormal eigenvectors of \( L \) corresponding to eigenvalues \( \lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_n \). Let \( U \in \mathbb{R}^{n \times k} \) be formed by the first largest \( k \) eigenvectors \( z_1, z_2, ..., z_k \) of \( L_{sym} \). A well-known property of such
linear algebra is that the $\mathbb{R}^{n \times n}$ matrix $L(k) = UU^T$ is the best rank-$k$ approximation to $L$; equivalently, the most energy-preserving projection to rank $k$. The original data points are now represented in a $k$-dimensional feature space, where data point $i$ is viewed as the $i$-th row of $U$.

Data points can be viewed as the rows of $U$, as mapping original data points into a $k$-dimensional feature space. The angle between two embedding data points is given as

$$\theta_{ij} = \arccos \frac{u_i^T u_j}{\|u_i\| \|u_j\|} \tag{6.5}$$

By normalizing each row of $U$, data points can be angle-preserving projected onto the surface of a $k$-dimensional hyper-sphere. So that

$$\theta_{ij} = \arccos(\bar{u}_i^T \bar{u}_j) \tag{6.6}$$

According to the Polarization Theorem (Brand and Huang, 2003), as the matrix $L$ is projected to successively smaller dominant eigenvectors, the sum of squared angle-cosines $\sum_{i,j} (\cos \theta_{ij})^2$ is strictly increasing. In other words, this implies that the angle will migrate away from $\pi/2$ towards $(0, \pi)$. Equivalently, the angles in the new representation shrink or grow further. Moreover, a corollary of the theorem ensures that small angles are least distorted when suppressing the eigenvectors of smallest magnitude. This implies that points of high similarity will move toward each other when $k$ decreases, while points of low similarity will move further apart. Therefore, clustering in the $k$-dimensional embedding space would be easier than clustering the original set of data points. Clustering of the embedded $\bar{u}_1, \bar{u}_2, ..., \bar{u}_n$ can be achieved by the standard $k$-means clustering simply by measuring Euclidean distances in the hyper-sphere. Alpert and Yao (1995) investigated the performance on the involved number of eigenvectors and showed that direct $k$-way partitioning considerably outperforms recursive spectral bipartitioning in terms of accuracy.

### 6.3 Roof Plane Segmentation

This section presents a novel approach for simultaneously partitioning line segment groups into planar patches based on the spectral clustering principle. Generally, spectral clustering algorithms first construct an affinity matrix by measuring pair-
wise similarities and then use the eigenvectors of the affinity matrix to enhance clustering of data points in the feature space. However, line segments by themselves fail to derive coplanar similarities directly. Therefore, the proposed algorithm begins with retrieving the embedded subspace of each line segment. The weighted adjacency matrix reflecting the subspace similarities is then constructed. The adjacency matrix is projected into a low-dimensional feature space for a powerful data representation. In the new representation, a density-based clustering is performed on data points. Each cluster corresponds to a planar roof segment so that parameters of each plane model can be easily determined. The designed spectral clustering flow chart is illustrated in Figure 6.1. A description is provided in the following in subsections.

Figure 6.1 Flow chart of spectral clustering.

6.3.1 Local Best-fit Plane Retrieval

To perform spectral clustering, coplanar similarity between a pair of line segments initially should be estimated. One observation is that a line segment and its nearest neighbours often belong to the same subspace because of spatial coherence. Let $X$ be extracted line segments. Hence, the subspace $\hat{S}_i$ for a line segment $x_i \in X$ can be obtained by the best-fit plane of line segments in its local neighbourhood. Then, if two line segments $x_i$ and $x_j$ lie in the same subspace $S_k$, their locally estimated subspaces $\hat{S}_i$ and $\hat{S}_j$ should be similar.

With the neighbourhood relationship (Zhang and Faugeras, 1992), the locally best-fit plane is then determined for each line segment. Commencing with a line segment
Chapter 6 Building Rooftop Modelling

\( x_i \in X \) and its neighbours \( X_e \subset X \), a set of candidate planes \( \{P_i\}_{i=1}^c \) formed by two end-points from \( x_i \) and the mid-point from \( x_j \), \( x_j \in X_e \) are generated. The line segment \( x_j \) is verified as an inlier of \( P_i \) if the residual from the mid-point of \( x_i \) to \( P_i \) is smaller than a threshold and \( n_i \cdot d_j \) is close to zero, where \( n_i \) is the normal of \( P_i \) and \( d_j \) is the direction of \( x_j \). The \( P_i \) with the most number of inliers is determined as the best-fit plane of the line segment \( x_i \).

### 6.3.2 Graph Laplacian Construction

The coplanar similarity of each pair of subspaces is measured using both angular difference and geodesic distance between the two planes.

The angular difference is formed as

\[
\text{DiffAng}_{ij} = \sin^2(\theta_{ij}) \tag{6.7}
\]

where \( \theta \) is the angle between the normal vectors of the two subspaces.

And the geodesic distance is defined as

\[
\text{DistGeo}_{ij} = \text{dist}^2(m_i, \hat{s}_j) \tag{6.8}
\]

where \( \text{dist}(m_i, \hat{s}_j) \) is the Euclidean distance from the mid-point \( m_i \) of \( x_i \) to subspace \( \hat{s}_j \).

Then the weighted \( \mathbb{R}^{n \times n} \) adjacency matrix is generated via an exponential kernel as

\[
w_{ij} = \exp(-\text{DiffAng}_{ij}/2\sigma_1^2 - \text{DistGeo}_{ij}/2\sigma_2^2) \tag{6.9}
\]

where \( \sigma_1 \) and \( \sigma_2 \) are the scaling parameters of the Gaussian Kernel and \( w_{ij} \in [0,1] \). Clearly, when \( w_{ij} \) is close to 1, it means both angular difference and geodesic distance are small, then \( x_i \) and \( x_j \) are expected in the same cluster.

To make the adjacency matrix unidirectional, the \( \mathbb{R}^{n \times n} \) affinity matrix is expressed as \( A_{ij} = (w_{ij} + w_{ji})/2 \), such that \( 0 \leq A_{ij} = A_{ji} \leq 1 \). To normalise the affinity matrix, a diagonal matrix \( D \) is defined as the degree of \( A \), in which \( D_{ii} \) is the sum of \( A \)'s \( i \)-th row. Afterwards, the normalised affinity matrix is used to generate a symmetric graph Laplacian:

\[
L_{\text{sym}} = D^{-1/2}AD^{-1/2} \tag{6.10}
\]
where \( L_{ij} = A_{ij} / \sqrt{D_{ii}D_{jj}} \). Line segments are now represented as data points in a \( \mathbb{R}^n \) feature space by each row vector.

### 6.3.3 Spectral Subspace Clustering

Once the graph matrix representation for data points is found, identification of the common underlying subspaces is the next step. The graph Laplacian is projected to the embedded space for clustering. The decomposition in lower dimensional space is derived by choosing the \( k \) largest eigenvectors from the eigenvalue decomposition of \( L_{sym} \). The determination of \( k \) values is from the eigengap heuristic (von Luxburg, 2007), where the criterion is to choose the number \( k \) such that all eigenvalues \( \lambda_1, ..., \lambda_k \) are relatively large, but \( \lambda_{k+1} \) is close to zero. The collected eigenvectors form the new matrix \( U \in \mathbb{R}^{n \times k} \). Let \( u_i \) in the \( i \)-th row of \( U \), the point with vector \( u_i \), represent the \( i \)-th straight line segment in eigenspace. To project a data matrix into a unit sphere, \( u_i \) is normalized to be a unit vector \( \hat{u}_i \), such that \( \|\hat{u}_i\| = 1 \).

Those data points are subsequently grouped in a unit sphere via mean-shift clustering (Comaniciu and Meer, 2002) so as to detect planar structures. The merit of mean-shift clustering is that mean-shift is a non-parametric algorithm based on density measurement, which iteratively computes the mean shift vector by translating density estimation windows until convergence. Mean-shift clustering processes without any prior knowledge of cluster quantities. Additionally, it can handle arbitrarily shaped clusters which make it more reliable for extraction applications. The plane model parameters of each cluster are measured through robust estimation algorithm such as RANSAC.

To clearly illustrate the described segmentation algorithm, the clustering procedure on a simple roof structure is demonstrated through Figure 6.2. The rooftop consists of 3,589 LiDAR points (Figure 6.2(a)). The points can be represented by 131 line segments obtained from LiDAR scan line segmentation, as shown in Figure 6.2(b). The line segments are further processed using the proposed algorithm. The local best-fit planes are first retrieved, and the weighted adjacency matrix is formed containing \( 131 \times 131 \) similarity measurements. Only weights over 0.7 are accepted while others are assigned as zero. In Figure 6.2(c), the accepted coefficients are
shown in grey while the others are in black (the illustration follows grouping orders). An approximately block-diagonal structure is obtained which can be easily discovered by the principle eigenvectors. Compared to the ideal form shown in Figure 6.2(d), the accuracy of the derived adjacency matrix is up to 81%. Spectral clustering is then applied on the constructed graph Laplacian, resulting in six cluster groups corresponding to six planar roof structures shown in Figure 6.2(e).

Figure 6.2 Illustration of spectral clustering process: (a) raw point cloud, (b) extracted line segments, (c) constructed adjacency matrix, (d) ideal adjacency matrix and (e) line-based clustering result.
6.4 Building Rooftop Reconstruction

As a consequence of clustering, multiple planar patches are extracted. The generation of a valid rooftop model requires identification of the boundary of each patch because planar patches have infinite spatial extents and they are partially present in building rooftops (Reisner-Kollmann et al., 2013). Therefore, the estimation of boundary vertices is a critical step since the two successive vertices naturally form a boundary edge. The boundary vertices correspond to either internal vertices to represent roof ridges or corner vertices to indicate step edges of a roof facet. In this work, internal vertices refer to the intersection points derived from adjacent roof planes, whereas corner vertices are defined as the vertices from the building footprint. Therefore, the constraints for boundary vertices are twofold: adjacent roof planes and extracted building footprint. The roof reconstruction process begins with the extraction of ridge lines and internal vertices from plane intersection through topological analysis. In the following step, the integration of roof ridges and footprint edges is applied to determine the corner vertices, which make the boundary complete.

6.4.1 Extraction of Roof Ridge and Internal Vertices

The initial boundary of each planar patch, derived from the data-driven approach, represents only the approximate shape, as shown in Figure 6.3. As can be seen, the initial boundary is composed of redundant vertices such that the boundary is too irregular to represent the general characteristics of the roof shape. In addition, the boundary vertices form the subset of raw measurements, resulting in some important vertices potentially being missed. In reality, a reasonable roof boundary is constrained by adjacent planes. For instance, roof ridges and internal vertices from plane intersection indicate part of the roof boundary (Sampath and Shan, 2010). Therefore, plane adjacent relations among roof planes play a key role in boundary estimation.

![Figure 6.3 Initial boundary of each planar patch.](image)
A pair of plane patches is defined as an adjacency when their regions are close to each other. Initially, for any two plane patches, the intersection line can be readily estimated. Then all boundary points from the two segments close to the line of intersection are marked as common points shared by the two planes. Note that some false intersections, such as corner-connected roof facets, also have common points. Hence, all common points are projected onto the intersection line to determine the roof ridge terminals. Roof ridges with short length are rejected.

Although terminals of generated ridges represent the approximate location of internal vertices, ridges often exhibit small gaps in between, as seen in Figure 6.4. In order to fill gaps, *Roof Topology Graph* (RTG) introduced by Verma et al. (2006) is employed to determine the junction points. In particular, RTG is an undirected graph to describe the adjacency relationships among planar patches. As shown in Figure 6.4(b), each planar patch is represented as a vertex in RTG and two adjacent patches are linked by an edge.

In graph theory, a basic cycle is a shortest closed path with three attributes:

- The source and target is the same vertex
- Each two consecutive vertices in the path are adjacent
- There are no repetitions of vertices in the path

In the context of RTG, a basic cycle indicates an internal vertex on the ridge (Huang and Brenner, 2011; Perera and Maas, 2014). For instance, the basic cycle from patch $P_1$, $P_2$ and $P_3$ indicates a junction point so that ridge $r_{1-2}$, $r_{1-3}$, and $r_{2-3}$ can be snapped together. When treating all edges equivalently, the basic cycle detection algorithm is proposed as follow:

1) Select a vertex from the graph as source. The vertex should be used first.

2) Select one of its adjacent vertices as the target and disconnect the edge in between. The edge should be used first.

3) Find the shortest path from source to target using the well-known depth-first search and denote the path as a closed cycle.

4) Reconnect the edge.

5) Repeat step 2 to 4 until all edges of the vertex are processed.
6) Repeat step 1 to 5 until all vertices are processed.

7) Remove the duplicate cycles.

In this way, RTG is decomposed into several basic cycles. For instance, the rooftop model shown in Figure 6.4 has four basic cycles in RTG and hence has four junction points, as shown in Figure 6.4(c). If the cycle contains three planar patches, then the location of junction points are directly measured from the intersection of the corresponding plane primitives. If more than three planar patches are found in a cycle, the least squares approach is applied. After detecting all conjunct points, roof ridges are snapped, as shown in Figure 6.4(d).

![Figure 6.4 Determination of roof ridge and inner vertices: (a) planar patches and intersection ridge, (b) roof topology graph, (c) four closed cycles and (d) corresponding internal vertices and snapped result.](image)

6.4.2 Completing the Boundary using the Footprint

To complete the roof shape, the extracted building footprints (Section 5.4) are employed. In fact, ridge lines and internal vertices only extract some part of the facet
boundary. Other boundary parts, such as step edges, are not detected through intersection. This is due to the lack of samples from façade structures. The building footprint describing the building shape is used to indicate the location of vertical planes, which helps to determine the corner vertices. In order to consistently merge the building footprint with building internal structures, a combination is proposed with the following regularization rules:

- **Rule 1**: Intersection angle of the step edge and horizontal ridge are conditionally regularized. Let \( \text{direction}_i \) and \( \text{direction}_j \) indicate the line directions of the step edge and horizontal ridge, then:

  \[
  \text{if } \exists |\text{direction}_i - \text{direction}_j| < 10^\circ, \text{ then } \text{direction}_i = \text{direction}_j, \text{ and } \\
  \text{if } \exists |\text{direction}_i - \text{direction}_j| \in 90^\circ \pm 10^\circ, \text{ then } |\text{direction}_i - \text{direction}_j| = 90^\circ.
  \]

- **Rule 2**: Location of internal vertices and footprint vertices are harmonized if they are close to each other. Let \( \text{Vertex}_i \) and \( \text{Vertex}_j \) be the vertices from the internal structure and footprint, then:

  \[
  \text{if } \exists |\text{Vertex}_i - \text{Vertex}_j| < 0.7m, \text{ then } \text{Vertex}_i = \text{Vertex}_j.
  \]

- **Rule 3**: Location of internal vertices is projected to the footprint if they are close to each other. Let \( \text{Vertex}_i \) and \( \text{Vertex}_{pi} \) be the vertices from the internal structure and the projection of \( \text{Vertex}_i \) on the footprint, then:

  \[
  \text{if } \exists |\text{Vertex}_i - \text{Vertex}_{pi}| < 0.7m, \text{ then } \text{Vertex}_i = \text{Vertex}_{pi}.
  \]

Rule 1 helps to solve direction regularization issue between internal and external structures. The constraint is from prior knowledge that the step edges are parallel or perpendicular to the horizontal ridge. After exploring direction constraints among horizontal ridges and step edges, a global least-squares method is used for solving the adjustment of line parameters. Weights for step edges and horizontal ridges are different, depending on the number of observations to them. For step edges, the weight is the count of all boundary points, while the weight for horizontal ridges
is the count of points from the two intersecting planes. Figure 6.5 illustrates the result before and after application of Rule 1.

Figure 6.5 Direction regularization: (a) before adjustment and (b) after adjustment.

When internal vertices are very close to corner vertices, the position of the internal vertex is relocated to the closest corner because the corner vertices are considered to be more accurate. Figure 6.6 shows the result before and after application of Rule 2.

Figure 6.6 Snapping of vertices: (a) before adjustment and (b) after adjustment.

For some roof types, such as a gable roof, internal vertices fail to find a neighbouring corner vertex, as shown in Figure 6.7. A vertical plane is defined by using the step edge and a horizontal normal. The actual locations of the internal vertices are relocated to the intersection of the three corresponding planes, as shown in Figure 6.7 by the result of regularization via Rule 3.
Figure 6.7 Consistency of vertices consistency: (a) before adjustment and (b) after adjustment.

The thresholds for the direction/distance difference should be linked to the resolution of the data. Generally, the value of the distance threshold is defined as 2 times the average point spacing, while the threshold of direction difference is empirically selected ($\pm 10^\circ$).

Note that corner vertices of some segments are not well defined from footprint vertices. For instance, the building roof in Figure 6.8 has three dormer structures. While internal vertices are determined by intersection analysis, corner vertices fail to be detected because substructures are completely inside the building footprint, as shown in Figure 6.8(a). To track this problem, segments within the building footprint need to be further processed. After identifying segments inside of the footprint, 3D bounding boxes, with the boundary edges either orthogonal or parallel to the normal of the corresponding planes, are established for those segments, as shown by the two large dormer planes. If the segment is a horizontal segment, its edges of the bounding box are either orthogonal or parallel to the dominant orientation when projected on the X-O-Y plane, as shown by the middle dormer plane in Figure 6.8(b). After generating 3D bounding boxes, the step edges of roof planes are extruded to the adjacent planes, as shown in Figure 6.8(c).

Figure 6.8 Modelling dormer structures: (a) combination of footprint and internal structures, (b) Determination of 3D bounding box for substructure and (c) building model with dormers.
6.5 Experiments

6.5.1 Test Datasets

The proposed approach has been evaluated on a benchmark dataset covering Vaihingen, Germany, which is made available by the International Society for Photogrammetry and Remote Sensing (ISPRS) Commission WG III/4. This publicly available dataset consists of aerial imagery and airborne LiDAR data to problem of urban object detection and 3D building reconstruction (Rottensteiner et al., 2014). The images were acquired using an Intergraph/ZI DMC aerial camera with 65% forward overlap and 60% side lap. The LiDAR data was collected using a Leica ALS50 system with 4-7 pts/m² data resolution. Three test sites were selected for generating reference data. Area 1 is characterized as a dense development site consisting of historic buildings with complex shapes. Area 2 contains a few high-rise residential buildings with mixed flat and sloped roof types. Area 3 is a purely residential area with detached buildings. A summary of the three areas is tabulated in Table 6.1. The ground truth datasets are detailed 3D models (LOD2) of the building roofs derived by manual stereo plotting. The proposed approach works on LiDAR data only, while images are used for visual inspection proposes. All three test sites have been used in the evaluation.

Table 6.1 Summary of the test sites.

<table>
<thead>
<tr>
<th>Site #</th>
<th>Size (m)</th>
<th>Number of buildings #</th>
<th>Building Characteristics</th>
</tr>
</thead>
<tbody>
<tr>
<td>Area 1</td>
<td>$125 \times 200$</td>
<td>37</td>
<td>Complex building shapes</td>
</tr>
<tr>
<td>Area 2</td>
<td>$170 \times 190$</td>
<td>14</td>
<td>High-rise buildings</td>
</tr>
<tr>
<td>Area 3</td>
<td>$150 \times 220$</td>
<td>56</td>
<td>Detached houses</td>
</tr>
</tbody>
</table>

6.5.2 Results

The performance and limitations of the proposed algorithm were initially evaluated by visual inspection. The segmentation results over three datasets are shown in the first column of Figure 6.9. The plane structures are represented as 2D polygons
visualised together with the orthoimage. A total of 187, 54 and 139 plane patches were extracted from each area. The reconstruction results are shown in the second column of the figure. Some 35, 17 and 61 building models were successfully reconstructed, respectively. The corresponding orthoimages are shown in the last column for visual inspection.

Figure 6.9 Experimental results from ISPRS test data. The first column shows the extracted plane structures overlaid on the true orthoimage. The second column illustrates the reconstruction results and the last column shows the true orthoimages.
Visual inspection illustrates that the proposed plane segmentation and building reconstruction algorithms work well in all three test areas. For example, Figure 6.10 shows some successful results with sloped roofs, which validate the proposed method. The results of planar segmentation and the internal feature extraction are shown in the first column. The second column represents the reconstructed building models and the last column shows the reference roof planes overlaid on the imagery.

Figure 6.10 Highlighted building models. The first column shows clustering results. The second column illustrates the reconstruction results and the last column shows referenced roof patches on the true orthoimages.

Although the building models generally fit well, some errors remain in the extraction process. Clearly, some roof parts are eliminated due to a lack of data, as
indicated in Circle 1 in Figure 6.11. In addition, a few flat and dense vegetation areas are incorrectly detected as roofs, as exemplified by Circle 2. It is observed that small roof structures attached to the ground are barely detected. For instance, the building in Circle 3 is missing, resulting in incomplete extraction. It is also observed that small planar structures with less than four line segments are barely detected, as shown in Circle 4. The reason is that the plane has too few coefficients in the affinity matrix to be distinguished. The quantitative analysis is provided in following sections.

Figure 6.11 Extraction errors: (a) absence of data; (b) low vegetation is detected as a building; (c) low building is detected as terrain; (d) small planar patches fail to be extracted.

6.5.3 Criteria for Quality Assessment

A further evaluation was performed to provide a quantitative assessment of rooftop models in terms of segmentation quality and geometric errors. Rutzinger et al. (2009) presented an evaluation scheme that uses overlap analysis at object level. Six
Chapter 6 Building Rooftop Modelling

evaluation metrics, defined as follows, were used for the quantitative evaluation of performance:

- **Completeness** ($Comp_{obj}, Comp_{10}$) is also termed as producer’s accuracy to evaluate how completely the approach is able to represent the roof planes. At object level, the True Positive (TP) is the number of derived roof planes that have an overlap of over 50% with roof planes in the reference data, while a False Negative (FN) is the number of roof planes in the reference data that have an overlap of less than 50% with derived roof planes. Completeness at object level is calculated as

$$Comp_{obj} = \frac{TP}{TP+FN}$$  \hspace{1cm} (6.17)

$Comp_{10}$ is the completeness when only considering roof planes larger than 10 m$^2$.

- **Correctness** ($Corr_{obj}, Corr_{10}$) is also referred to as user’s accuracy. At object level, a False Positive (FP) is the number of derived roof planes that have an overlap of less than 50% with roof planes in the reference data. Correctness at object level is defined as

$$Corr_{obj} = \frac{TP}{TP+FP}$$  \hspace{1cm} (6.18)

$Corr_{10}$ is the correctness when only considering roof planes larger than 10 m$^2$.

- **Quality** ($Q_{obj}, Q_{10}$) provides a compound performance metric that balances completeness and correctness (Rottensteiner et al., 2014). This is expressed as

$$Q_{obj} = \frac{TP}{TP+FP+FN}$$  \hspace{1cm} (6.19)

$Q_{10}$ is the quality when only considering roof planes larger than 10 m$^2$.

- **Segmentation relationships** ($N_{1:M}/N_{N:1}/N_{N:M}$) reflect the differences in the topologies of the extracted roof planes and the reference data. $N_{1:M}$ is the number of one-to-many relationships between reference roof planes and extracted planes, which is also known as over-segmentation. $N_{N:1}$ is the number of many-to-one relations which also indicates under-segmentation. $N_{N:M}$ is the number of many-to-many relations which mix over- and under-segmentation.
• Planimetric error \((RMSXY)\) expresses the root mean square value of the planimetric distance of the reference roof plane boundary points to their nearest neighbours on the corresponding extracted roof plane boundaries. Let \(n_b\) be the number of boundary points and \(\Omega_1\) be the squared planimetric difference accumulated from all boundary points, then the \(RMSXY\) is defined as

\[
RMSXY = \sqrt{\frac{1}{n_b} \Omega_1}
\]  
6.20

• Height error \((RMSZ)\) measures the root mean square value of the vertical distance between the reference roof plane and the derived roof plane based on a grid process. Let \(n_g\) be the number of grid points on the roof planes and \(\Omega_2\) be the sum of the squared vertical differences, accumulated from all grid points, then the \(RMSZ\) is defined as

\[
RMSZ = \sqrt{\frac{1}{n_g} \Omega_2}
\]  
6.21

6.5.3 Evaluation

The reconstruction results were sent to the ISPRS benchmark test evaluation group, and were evaluated independently. 187, 54 and 139 roof planes were extracted from the three test areas. The evaluation results are tabulated in Table 6.2.

Table 6.2 Statistical evaluation of the building reconstruction results.

<table>
<thead>
<tr>
<th></th>
<th>Area 1</th>
<th>Area 2</th>
<th>Area 3</th>
</tr>
</thead>
<tbody>
<tr>
<td>(Comp_{obj} / Comp_{10} \ [%])</td>
<td>88.2 / 95.2</td>
<td>71.0 / 91.7</td>
<td>82.6 / 96.0</td>
</tr>
<tr>
<td>(Corr_{obj} / Corr_{10} \ [%])</td>
<td>99.5 / 99.2</td>
<td>98.1 / 100.0</td>
<td>97.8 / 100.0</td>
</tr>
<tr>
<td>(Q_{obj} / Q_{10} \ [%])</td>
<td>87.8 / 94.5</td>
<td>70.1 / 91.7</td>
<td>81.1 / 96.0</td>
</tr>
<tr>
<td>(N_{1:M}/N_{N:1}/N_{N:M})</td>
<td>3 / 40 / 2</td>
<td>2 / 2 / 0</td>
<td>2 / 44 / 1</td>
</tr>
<tr>
<td>(RMSXY \ [m])</td>
<td>0.70</td>
<td>0.94</td>
<td>0.71</td>
</tr>
<tr>
<td>(RMSZ \ [m])</td>
<td>0.23</td>
<td>0.32</td>
<td>0.16</td>
</tr>
</tbody>
</table>
The proposed algorithm achieved over 80% completeness (Comp) for Area 1 and Area 3. Completeness was lower in Area 2, at 71%. The performance is better for roofs with area larger than 10m². The completeness for all areas increases up to above 90%, with over 95% in Areas 1 and 3. The higher completeness in Areas 1 and 3 indicates that the algorithm performs better in residential areas with detached buildings. The evaluation shows the correctness (Corr) in all test areas is above 97%. For larger sized roofs, the correctness for Areas 2 and 3 is 100%, indicating reliable results have been achieved by the new developed algorithm. The quality for Areas 1 and 3 is above 80% while Area 2 is 70.1% because the completeness of this area is low. On the other hand, the quality of all testing areas is above 91% for larger sized roofs.

Figure 6.12 illustrates the segmentation errors in terms of topological differences between reference and detected planes. Over-segmentation (1:N) is low for all test areas (maximum 3), while under-segmentation (M:1) is the major concern in Areas 1 and 3, with 40 and 44 cases, respectively. The low under-segmentation in Area 2 is mainly because some roof planes of small size are not detected (low completeness). The mixed error (M:N) is also low in all three test areas.

Figure 6.12 Topological differences: reference planes versus extracted planes in the three test areas.
Chapter 6 Building Rooftop Modelling

The RMSXY errors are above 0.7m for all three testing area, which is less than 1.5 point spacing. Higher quality measurement is indicated by a lower RMSXY value. Areas 1 and 3 have better quality than Area 2 since they exhibit a smaller RMSXY value (0.7m instead of 0.94m). The RMSZ values, which express consistency with the DSM are all smaller than 0.32m. The higher completeness rate in Areas 1 and 3 results in smaller RMSZ errors than in Area 2. Overall, the proposed method achieves very good rooftop reconstruction results.

6.6 Chapter Summary

In this chapter, planar patch extraction and building modelling have been presented. Unlike use of a planar model for data fitting, the task of plane extraction is transformed into spectral subspace clustering by exploring the similarities between the line segments. The line segments, rather than redundant raw LiDAR points, are employed as the basic elements which convey more geometric information for successive processing while significantly reducing the data volume in the clustering analysis. Line segments are projected to spectral space through a graph Laplacian transformation and spectral decomposition, which enhances the correlation of high-similarity objects while penalizing dissimilar objects, thus allowing more efficient plane detection via a mean-shift clustering algorithm in feature space. Plane boundaries are then extracted from adjacent planes. Furthermore, a rule-based regularization is proposed to complete the plane boundary with help from the building footprint.

The new developed method has been experimentally evaluated on the ISPRS benchmark LiDAR data over three test sites, and the independent evaluation by the ISPRS benchmark test group confirmed that an average 80% completeness, with over 98% correctness, was achieved. Some minor errors are observed on dense vegetation with flat tops, which are incorrectly detected as roofs. In addition, buildings with small size are not detected due to insufficient straight line segments on their roofs. This is indicated in the evaluation where the completeness increases to 92% for large sized buildings. It is also noted that the correctness is very high in all three test areas. The close to 100% correctness on large roofs demonstrates the robustness and high reliability of the new developed method.
Chapter 7 Conclusion and Future Work
7.1 Conclusion

A new automated 3D building modelling system using airborne LiDAR has been introduced in this thesis. The proposed system exploits spatial and geometric information of LiDAR data, and consists of three processing modules: point cloud filtering, building footprint extraction and building rooftop reconstruction. Details of the new system have been presented in the separate chapters describing each module. An analysis of input LiDAR datasets and a more general overview of the whole system are reviewed in this section.

The input data source for the proposed system is airborne LiDAR data with a nadir perspective. Rather than using spectral properties, the strategy of feature extraction relies heavily on height context. In the terrain filtering step, the sample point is employed as the basic primitive for extraction. In contrast, both building footprint and rooftop modelling steps employ straight line segments, from the decomposition of scan lines, for building detection and reconstruction. The main advantage is to reduce computational load, which enables large-scale processing.

All developed functions have enabled the system to successfully perform building extraction from LiDAR data. The results and assessments show that the developed system fulfils the objectives stated in Section 1.3. The developed system can thus be utilised as a practical tool for automatic building extraction from airborne LiDAR data. Compared to existing approaches (reviewed in Chapter 2), the new system has a number of advantages.

This research project was one of the first to utilize soft constraints to deal with ambiguities in sensor data and uncertainties in the modelling process. Soft constraints provide a clear expression of the problem so that an optimal solution can be determined by solving a minima problem. The proposed system exploits both data constraints and prior constraints in various forms to better formulate solutions to the extraction and reconstruction problem. Traditional approaches define data constraints based on an interactive manner, e.g. researches by Golovinskiy and Funkhouser (2009). Such methods require cues from the user such that the optimization can be carried out. The proposed system, on the other hand, is able to automatically extract preliminary models from data, thus significantly reducing the need for human input.
In addition, the preliminary model can be improved from the intermediate results to obtain a more complete model.

The use of prior knowledge of the scene is one of the key components in the developed system to achieve reliable building reconstruction. The proposed system attempts to find a balanced solution between the integrity of the geometric fit and prior knowledge about the topological regularity, such as parallelism and perpendicularity in footprint and direction alignment in roof ridges. The topological regularity much more preserves design considerations and satisfies human visual assessment. The combination of observation data and prior knowledge is one of the key components in the designed system to achieve reliable reconstruction. This is a major advantage of the system over other reported approaches, which rarely depend on rule-based extraction. It is possible to reliably infer the preliminary model if prior knowledge is employed. In addition, the combination of information from the data and prior terms can be used to compensate incomplete results. For instance, small footprint edges are confirmed when the observation points in that areas are insufficient. In conclusion, the use of data and prior terms is one of the important keys to reducing the complexity of LiDAR data processing, to compensating for limited data resolution, and to increasing reliability for building extraction.

Several criteria have been defined to evaluate the terrain models, footprint models and rooftop models. For terrain evaluation, the adjustment of the preliminary model proved to be useful. More than 24% of terrain points were further identified in hilly areas using the adjustment of the preliminary model. Low values of Type I error and total error indicate that the results are robust. For footprint evaluation, the implicit reconstruction is important for linking two lines where the samples are limited. All the extracted models are closed polygons with regularized shape. As compared to the explicit reconstruction approach, the developed method offers a more accurate preliminary model and it generates more reasonable details. For rooftop evaluation, the utilization of line segments can drastically reduce the size of the adjacency matrix, which can make the clustering more efficient. More than 70% completeness and 95% correctness were achieved. The geometrical accuracy is about 0.7m for planimetry and 0.3 for height. The benchmark results showed that the developed system has excellent performance in terms of completeness, correctness and data fitness. The results can also be viewed at the website of the ISPRS [http://www2.isprs.org/](http://www2.isprs.org/)
Nonetheless, over-segmentation is still an issue because the developed approach employs line segments for clustering, which reduces the data redundancy on small planes.

The research work of this PhD thesis still leaves some minor issues unresolved. The performance of the optimization shows that the weighting parameter is critical. However, the parameter is determined empirically in this work. It is very difficult to find the optimal value to balance data and prior terms, which need to give consideration of the whole landscape. A post-processing step or manual inspection for error correction is necessary. In addition, the rooftop reconstruction was only performed on LiDAR data of medium resolution. Further tests for various point densities are necessary to report the robustness and efficiency of the developed approach.

7.2 Contributions

In this thesis, the proposed system has addressed problems of ambiguities in sensor data and uncertainties in point cloud processing in modelling urban scene. Towards this, the thesis makes the following key contributions:

- A probabilistic graphical model is proposed for LiDAR point cloud filtering, which integrates geometric location and spatial coherence in the form of t-links and n-links. Based on the graphical structure, accurate extraction is derived from an energy function that can be optimized by graph cuts. Instead of the standard graph cuts technique of providing the extraction in one-shot, an iterative-learning optimization scheme is proposed to refine the result for the subsequent extraction. The problem of missing cues in the preliminary DTM is thus alleviated.

- Utilisation of a combined energy function encoding the deviation of points from a certain polygon, along with the complexity hypothesis model for determination of the optimal approximation polygon for building outlines, where the model complexity is controlled by the number of vertices obtained from the vertex-driven DP algorithm. An explicit reconstruction incorporates geometric knowledge of buildings (parallelism and perpendicularity) in a global
optimization to improve robustness and accuracy; and an implicit reconstruction fills the edge gaps through a connector structure to ensure completeness and topological correctness of the derived polygon.

- The task of planar feature extraction is transformed into spectral subspace clustering by exploring the similarities among line segments. The derived line segments are projected to spectral space through graph Laplacian transformation and spectral decomposition. This enhances the correlation of high-similarity points while penalizing dissimilar points, which allows more efficient plane detection via a mean-shift clustering algorithm in feature space. After deriving the planar pitches, the roof vertices are recovered by adjacency analysis of patches through a robust circle detection algorithm.

### 7.3 Future Work

Although the experiments demonstrated promising results, there are still a number of features that could be improved by either the employment of new data sources or the development of additional novel methods in each individual step. Possible future work lies within three areas.

- **Modelling from image information**

  Airborne LiDAR data has significant advantages in providing surface information for the building modelling task, as demonstrated in this thesis. However, scattered point clouds rarely capture sufficient edge features. On one hand, further work may alleviate this problem by the integration of line features from ortho-imagery since edge detection in high resolution imagery is more reliable. On the other hand, with the rapid advances in image matching techniques for highly overlapping imagery, image matching could alternatively provide qualified positional and radiometric information from one sensor. This presents a new challenge of how to employ spectral properties to assist in or verify edge detection.

- **Modelling with vertical structures**
This research has been restricted to considerations for nadir viewing LiDAR datasets. As a result, achievements describe detailed terrain surface and building roofs, while vertical façades are represented as the extrusion of the building footprint. Some key vertical structures, such as window and vegetation, fail to be represented. Auxiliary information from oblique aerial or terrestrial LiDAR/imagery datasets needs to be integrated. Further research may require solving the consistency issue between skyward and vertical structures from the two data sources.

- **Modelling of more complex building shapes**

In this study, the developed system assumes that building footprints consist of sets of piecewise linear segments and building rooftops consist of sets of planar structures. While the assumptions are true for the majority of buildings, those with ‘special’ structural characteristics, such as domed and curved structures, are not uncommon. Improvement of the current approach and/or development of new methods are warranted to accommodate these cases. Transfer learning is recognized as the new framework to address the determination of weight parameter from training datasets.
Bibliography


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


Verma, V., Kumar, R., Hsu, S., 2006. 3D building detection and modeling from aerial LiDAR data. Computer Vision and Pattern Recognition 2, 2213–2220.


List of Publications

The publications listed below are based on the work presented in this thesis.


Yuxiang He was born in Zhengzhou, China on 7th November, 1984. From 2004 to 2008, he studied Remote Sensing and Photogrammetry at Zhengzhou Institute of Surveying and Mapping, and he received his Bachelor-level degree in 2008. After graduation, he enrolled for a Master of Science by research at the Earth Observation Science Department, ITC, University of Twente, The Netherlands, with a specialization in Geoinformatics. He received the M.Sc degree in 2010 with a thesis titled *Digital Terrain Models Extraction from Huge LiDAR Data*. After that, he joined the Remote Sensing Engineering Centre, Chinese Academy of Surveying and Mapping where he worked on LiDAR applications.

In May 2011, he commenced his PhD research in the Dept. of Infrastructure Engineering at the University of Melbourne, Australia, with scholarship support from the Cooperative Research Centre for Spatial Information (CRCSI). He was also supported through a Melbourne International Fee Remission Scholarship (MIFRS) and Melbourne Abroad Travelling Scholarship (MATS). The research conducted has been within the theme of CRCSI Project 2.02, and has focussed upon automated 3D feature extraction and urban modelling from airborne LiDAR data. The outcomes of the PhD research are presented in this thesis.