POINT CLOUD-BASED ANALYSIS AND MODELLING OF URBAN ENVIRONMENTS AND TRANSPORTATION CORRIDORS

by

Yun-Jou Lin

A Dissertation

Submitted to the Faculty of Purdue University

In Partial Fulfillment of the Requirements for the degree of

Doctor of Philosophy

Lyles School of Civil Engineering

West Lafayette, Indiana

December 2018

THE PURDUE UNIVERSITY GRADUATE SCHOOL

STATEMENT OF COMMITTEE APPROVAL

Dr. Ayman Habib, Chair

Lyles School of Civil Engineering

Dr. Melba Crawford

Lyles School of Civil Engineering

Dr. James Bethel

Lyles School of Civil Engineering

Dr. Darcy Bullock

Lyles School of Civil Engineering

Approved by:

Dr. Dulcy Abraham

Head of the Graduate Program

To my beloved parents

ACKNOWLEDGMENTS

First and foremost, I would like to express my sincere gratitude to my advisor Dr. Ayman Habib for all his invaluable guidance, encouragement, and support during my Ph.D. studies. He has inspired me to become an independent researcher and helped me to understand the power of critical reasoning. Without his valuable suggestions on my research and providing excellent study environment, this dissertation would not have been possible. Besides my advisor, I would like to thank Dr. Bullock for his valuable suggestions and opinions on the transportation-related research. I also would like to thank Dr. Förstner for his guidance in building model reconstruction research. Moreover, I would like to express my appreciation to Dr. Mikhail for being my mentor and for encouraging me. Furthermore, my sincere thanks must also go to the rest of my thesis committee: Dr. Melba Crawford and Dr. James Bethel for their insightful comments and encouragement, but also for the hard questions which incented me to widen my research from various perspectives.

My gratitude also goes to my colleagues at the Digital Photogrammetry Research Group, Fangning He, Weifeng Xiong, Radhika Ravi, Magdy Elbahnasawy, Tamer Shamseldin, Megan Miller, Ronald Benziger, Yi-Ting Chen, Yi-Chun Lin, Tian Zhou, Meghdad Hasheminasab, and Guo Zhou. I would also like to appreciate all the help from my friends, Keng-Fan Lin, Zhou Zhang, Ma Chen, and Man Chung Chim.

I deeply thank my dear parents and dear one and only brother for their unconditional trust, timely encouragement, and endless patience.

TABLE OF CONTENTS

ACKNOWLEDGMENTS ii

TABLE OF CONTENTS iii

LIST OF TABLES viii

LIST OF FIGURES x

LIST OF ABBREVIATIONS xvi

ABSTRACT xviii

1. INTRODUCTION 1

1.1 Problem Statement 4

1.1.1 Challenges in Point Cloud Down-sampling 4

1.1.2 Challenges in Segmentation Technique 5

1.1.3 Challenges in Digital Building Model Generation 6

1.1.4 Challenges in Lane Width Estimation 7

1.2 Research Objectives 8

1.3 Dissertation Outline 14

2. LITERATURE REVIEW 15

2.1 Introduction 15

2.2 Point Cloud Generation 15

2.2.1 Image-based Point Cloud 16

2.2.2 Laser-based Point Cloud 17

2.3 Existing Down-sampling Techniques 18

2.4 Existing Segmentation Techniques 19

2.4.1 Spatial-domain Segmentation Techniques 20

2.4.2 Parameter-domain Segmentation Techniques 21

2.4.3 Quality Control Procedures 22

2.5 Urban Environments – DBM Reconstruction 23

2.5.1 Data Sources for DBM Generation 24

2.5.2 Reconstruction Strategies of DBM Generation 25

2.6 Transportation Corridors – Lane Width Estimation 27

2.6.1 Road Surface Extraction 28

2.6.2 Lane Marking Extraction 28

2.6.3 Characteristics of Road Extraction 30

3. FIRST TIER OF POINT CLOUD PROCESSING 33

3.1 Introduction 33

3.2 Data Characterization and LPD Estimation 34

3.3 Adaptive Down-Sampling on Planar Region 36

3.4 Multi-Class Simultaneous Segmentation Procedure 39

3.4.1 Distance-Based Region Growing for the Derivation of Seed Regions 40

3.4.2 PCA-Based Classification and Modeling of Seed Regions 40

3.4.3 Region-growing Starting from Optimally-selected Seed Regions 43

3.4.4 Sequential Segmentation of Non-segmented Points and Rough Regions 44

3.5 QC of Segmentation Outcomes 45

3.6 Experimental Results 55

3.6.1 Data Description 55

3.6.2 Experimental Results of Down-sampling Procedure 59

3.6.2.1 Data Characteristics 60

3.6.2.2 Results & Analysis 62

3.6.3 Experimental Results of Multi-Class Simultaneous Segmentation and QC Procedure 75

3.6.3.1 Multi-Class Simultaneous Segmentation Results 76

3.6.3.2 Results of QC Outcomes 80

3.7 Summary 87

4. DIGITAL BUILDING MODEL GENERATION FOR COMPLEX BUILDINGS 88

4.1 Introduction 88

4.2 Building Primitive Boundary Extraction and Spike Removal 89

4.3 Boundary Regularization Using ERMBR Approach 91

4.3.1 Conceptual Basis of RMBR 91

4.3.2 ERMBR Procedure 97

4.3.2.1 Simultaneous RMBR and LSA Integration 97

1st Level MBR Reconstruction and LSA Integration 97

Sub-level MBR Reconstruction and LSA Integration 101

Global Boundary Refinement 103

4.3.2.2 Generalized RMBR Procedure 105

4.4 Boundary Regularization Using Breakpoint Detection Approach 110

4.4.1 Conceptual Basis of Breakpoint Detection Approach for 1D Profile 110

4.4.2 Breakpoint Detection Approach for 2D Building Boundary 113

4.4.3 Boundary Regularization Using Detected Breakpoints 114

4.5 Integration of the Outcome of the Regularization Strategies 118

4.6 Water-tight DBM Generation 122

4.6.1 Identification of Adjacent Segments from Neighboring Primitives 124

4.6.2 Alignment for Primitive with All Straight-line Segments 125

4.6.3 Segment Merging 126

4.7 Experimental Results 129

4.7.1 Building Identification and Boundary Extraction Results 131

4.7.2 Boundary Regularization Results 133

4.7.3 Water-tight DBM Results 140

4.8 Summary 145

5. LANE WIDTH ESTIMATION 146

5.1 Introduction 146

5.2 System Architecture of the TMMS Used in this Thesis 146

5.3 Conceptual Basis of LiDAR Point Positioning and System Calibration 148

5.4 Lane Width Estimation Strategy 150

5.4.1 Road Surface and Lane Marking Extraction 152

5.4.2 Derivation of the Lane Marking Centerline 154

5.4.3 Lane Width Estimation 158

5.5 Experimental Results 160

5.5.1 Data Description 160

5.5.2 Experimental Results of Lane Width Estimation 164

5.5.2.1 Impact of Mounting Parameters on Lane Width Estimation 166

5.5.2.2 Performance Evaluation of Different Sensors 168

5.5.2.3 Repeatability Analysis of Lane Width Estimation 171

5.5.2.4 Accuracy Analysis of Lane Width Estimation 175

5.6 Summary 177

6. CONCLUSION AND RECOMMENDATION FOR FUTURE RESEARCH 179

6.1 Summary of Contributions 179

6.2 Recommendations for Future Work 182

REFERENCES 184

LIST OF TABLES

Table 3.1. *Noise-level-to-area ratio* for representative sample points in the MTLS, STLS2, and DIM datasets 61

Table 3.2. Down-sampling parameters for the adaptive and point-spacing-based approaches 63

Table 3.3. LPD statistics for the original and down-sampled MTLS datasets (Note that the dimensionality-based PCA classification is used for adaptive down-sampling) 64

Table 3.4. LPD statistics for the original and down-sampled STLS2 datasets (Note that the dimensionality-based PCA classification is used for adaptive down-sampling) 65

Table 3.5. LPD statistics for the original and down-sampled DIM datasets (Note that the dimensionality-based PCA classification is used for adaptive down-sampling) 66

Table 3.6. LPD statistics for the original and down-sampled DIM datasets (Note: the threshold-based PCA classification is used for adaptive down-sampling) 67

Table 3.7. Multi-class simultaneous segmentation execution times for the different datasets..... 70

Table 3.8. Segmentation results after the QC procedure for the different datasets 71

Table 3.9. Time performance of the proposed segmentation.. 77

Table 3.10. QC measures for the different datasets 84

Table 4.1. Pre-defined thresholds used for DBM generation 130

Table 4.2. Statistical evaluation of the deviation between the reference dataset and the integrated models 138

Table 4.3. Statistical evaluation of the deviation between the reference dataset and the generated water-tight building models 144

Table 5.1. Details of test datasets collected in this thesis 162

Table 5.2. Mounting parameters of the MMS used to collect all datasets, as estimated from the multi-sensor system calibration procedure 163

Table 5.3. Pre-defined thresholds for lane width estimation 165

Table 5.4. Data processing time for each step of lane width estimation 166

Table 5.5. Statistics of comparison between lane width estimates using accurate and inaccurate values of mounting parameters 168

Table 5.6. Statistics of comparison between the lane width estimates from VLP16 and HDL32E sensors 169

Table 5.7. Statistics of lane width comparison from interstate highway datasets and U.S. highway datasets 174

LIST OF FIGURES

Figure 1.1. Flowchart of the proposed two-tier data processing framework 13

Figure 2.1. Segments invading/invaded and over-segmentation problems 23

Figure 3.1. An illustration of adaptive down-sampling 38

Figure 3.2. (a) Simulated point cloud, (b) derived point cloud after applying adaptive down-sampling, (c) and (d) the point density distribution corresponding to (a) and (b) 39

Figure 3.3. Representation scheme for 3D planar features; planes that are almost parallel to the $xy-plane$ (a), planes that are almost parallel to the $xz-plane$ (b), and planes that are almost parallel to the $yz-plane$ (c) 41

Figure 3.4. Representation scheme for 3D pole-like features; pole-like features that are almost parallel to the $z-axis$ (a), pole-like features that are almost parallel to the $y-axis$ (b), and pole-like features that are almost parallel to the $x-axis$ (c) 42

Figure 3.5. Classified point cloud: planar, pole-like, and rough regions are shown in blue, green, and red, respectively 46

Figure 3.6. Possible segmentation artifacts: misclassified planar features (a) and (b); misclassified pole-like feature (c); partially misclassified planar features (d), (e), and (f); and partially misclassified pole-like feature (g). Planar and pole-like features are displayed in light blue and light green, respectively, in subfigures $a, b, c, e, f, and\ g$ 48

Figure 3.7. Inner and outer boundary derivation for the identification of intraclass competition for neighboring points 51

Figure 3.8. Slicing and immediate-neighbors concept for the identification of fully/partially misclassified pole-like features (a)/(b)... 53

Figure 3.9. Segmented planar feature (in light blue) and the encompassing MBR (in red (a), segmented linear features (in green) (b), and final segmentation after the identification of partially-misclassified linear features (c) 55

Figure 3.10. Point clouds from the (a) MTLS, (b) ALS, (c) STLS1, (d) STLS2, (e) STLS3, and (f) DIM datasets 58

Figure 3.11. Point density maps of (a) original as well as the (b) adaptively, (c) uniformly, and (d) point-spacing-based down-sampled DIM datasets (Note: the scale bars show the LPD along planar surfaces in $pntsm2$) 68

Figure 3.12. Segmentation-based classification of the (a) original MTLS dataset as well as (b) adaptively, (c) uniformly, and (d) point-spacing-based down-sampled datasets 73

Figure 3.13. Planar and linear/cylindrical segmentation results of the (a) original STLS2 dataset as well as snippets that show zoomed-in areas of the segmentation outcome for the (b – d) original, (e – g) adaptively, (h – j) uniformly, and (k-m) point-spacing-based down-sampled datasets 74

Figure 3.14. Planar and linear/cylindrical feature segmentation results for the (a) original DIM dataset as well as the (b) adaptively, (c) uniformly, and (d) point-spacing-based down-sampled datasets 75

Figure 3.15. Perspective views of the classified point clouds for the ALS (a), STLS1 (b), STLS2 (c), STLS3 (d), and DIM (e) datasets (planar, pole-like, and rough regions are shown in blue, green, and red, respectively) 78

Figure 3.16. Perspective views of the segmented point clouds for the ALS (a), STLS1 (b), STLS2 (c), STLS3 (d), and DIM (e) datasets (different segments are shown in different colors) 79

Figure 3.17. Perspective views of the segmented point clouds after the QC procedure for the ALS–planar (a), DIM–planar (b), STLS1–planar (c), STLS1–pole-like (d), STLS2–planar (e), STLS3–planar (f), and STLS3–pole-like (g) datasets; different segments shown in different colors 83

Figure 3.18**.** Examples of improved segmentation quality by the different QC measures 85

Figure 4.1. Flowchart of digital building model generation procedure 89

Figure 4.2. Illustration of spike removal procedure: (a) a partial boundary with a spike, (b) smaller angles between two adjacent edges, (c) boundary after removing spike A, (d) boundary after removing spike B, and (e) highly irregular boundary 91

Figure 4.3. (a) MBR with minimum area and (b) MBR with maximum overlapping points 93

Figure 4.4. Illustration of the RMBR procedure: (a) initially-extracted boundary, (b) 1st level MBR generation, (c) non-overlapping point identification, (d) projected point generation, (e) 2nd level MBRs generation, (f) the shape of building model after the incorporation of the second-level MBRs, (g)-(i) repeat the procedure of (c) to (f) while considering 3rd level MBR 94

Figure 4.5. Illustration of incorrect building model as a result of inappropriate Th_{pt} value: (a) extracted boundary and (b) shape of building model 96

Figure 4.6. Illustration of previous RMBR strategy leading incorrect shape of building model (a) the shape of a building, (b) 2nd level MBRs (MBRs – b, d, e, f, and g as well as overlapping MBRs – a and c), (c) updated shape of the building model after subtracting the second-levels, and (d) incorrect final shape of building model after adding the 3rd level MBR 96

Figure 4.7. Illustration of simultaneous RMBR and LSA boundary regularization: (a) overlapping point identification (orange points), (b) derived parameters from 1st level MBR adjustment using LSA, (c) normal distance for an overlapping point to a corresponding edge of the MBR, (d) refined 1st level MBR in red, (e) identification of non-overlapping points (blue points), (f) projection of non-overlapping boundary points (red points), (g) 2nd level MBR generation, (h) LiDAR boundary points overlapping with 2nd level MBRs (orange points), (i) refined 2nd level MBRs in red, (j) the shape of building model after the incorporation of the second-level MBRs, and (k)-(m) repeat the procedure of (f) to (j) 99

Figure 4.8. Refinement of sub-level MBR for a shared-corner case 103

Figure 4.9. Refinement of sub-level MBR for a non-shared-corner case.................................. 103

Figure 4.10. Illustration of boundary: (a) before refinement and (b) after refinement.............. 104

Figure 4.11. Detection of overlapping sub-level MBRs... 106

Figure 4.12. Generalized RMBR procedure: (a) original building, (b) 1st level MBR, (c) 2nd level MBRs, (d) subtraction of general 2nd level MBRs, (e) RMBR process of *overlapping MBRs*, (f) subtraction of overlapping 2nd level *oMBR*, and (g) final building model obtained from the subtraction of overlapping models (B1 and B2) from the general model (A)........................... 108

Figure 4.13. Illustration of modeling the surface profile.. 112

Figure 4.14. Breakpoint detection for building boundary (a) 1D (U-direction and V-direction) profiles of building boundary, (b) 2D building boundary, black: LiDAR boundary points, gray: densified and smoothed points, red: breakpoints.. 114

Figure 4.15. Wrongly detected breakpoints.. 115

Figure 4.16. Correct vertex derived from an intersection of neighboring segments 116

Figure 4.17. Wrong vertex derived from the intersection of neighboring segments and the derivation of correct vertices: (a, c) a straight-line segment to a straight-line segment, (b, d) a straight-line segment to an curved segment.. 117

Figure 4.18. Wrong vertex from both *extracted segments* and *breakpoint segment* 118

Figure 4.19. Example of the disadvantage of ERMBR approach, (a) orthophoto, (b) primitive with multi-orientation and curved boundary, (c) primitive with highly irregular boundary points, red dot: vertices of ERMBR boundaries, black dot: LiDAR boundary points, blue line: regularized boundaries 119

Figure 4.20. Example of the disadvantage of BD approach, (a) orthophoto, (b) incorrectly regularized boundary, (c) regularized boundary without right-angle corners, red dot: vertices of BD boundaries, black dot: LiDAR boundary points, grey dot: densified and smoothed points, red circle: breakpoints, blue line: regularized boundaries, green line: LiDAR boundary 120

Figure 4.21. Illustration of integration of the outcome of the BD and ERMBR strategies 122

Figure 4.22. Illustration of integration strategy that can avoid incorrect segments from ERMBR boundaries and derive the boundaries with right-angle corners 122

Figure 4.23. Illustration of different types of edge merging (a) parallel straight-line segment merge, (b) non-parallel straight-line segment merge, and (c) curved segment to straight-line segment merge, and (d) curved segment to curved segment merge... 123

Figure 4.24. Illustration of identification of adjacent segments from neighboring primitives .. 125

Figure 4.25. Illustration of primitive alignment for primitives with all-straight line segments . 126

Figure 4.26. Illustration of (a) before, and (b) after parallel straight-line segment merging...... 127

Figure 4.27. Illustration of (a) before, and (b) after non-parallel straight-line segment merging 128

Figure 4.28. Illustration of (a) before, and (b) after straight-line and curved segment merging 128

Figure 4.29. Illustration of (a) before, and (b) after curved segment merging 128

Figure 4.30. Perspective view of the LiDAR point cloud colored by height............................. 129

Figure 4.31. (a) Segmentation results, (b) planar segments after QC procedure, (c) ground/non-ground classification, (d) building hypothesis, (e) building boundary 132

Figure 4.32. Integrated models derived from the integration of the outcome of ERMBR and BD approaches (a) overview of whole area, (b)-(g) zoom-in areas 134

Figure 4.33. Reference data for validation (blue line: integrated models, red points: manual measurements) 137

Figure 4.34 Incorrect results that were not included in the validation procedure (a) LiDAR boundaries, (b) integrated models... 137

Figure 4.35. Building with a parapet: (a) segmented points (blue) and LiDAR boundary (green), and (b) building model in red.. 139

Figure 4.36. Building with short segments: (a) LiDAR boundary (green), and (b) building model in red 139

Figure 4.37. Comparison between the building primitives before and after applying water-tight building model generation strategy: (a) – (b) parallel straight-line segment merging, (c) – (d) non-parallel segment merging (curved-line to curved-line), and (e) – (h) parallel and non-parallel straight-line segment merging.. 141

Figure 4.38. Produced 3D seamless digital building models from the proposed methodology . 143

Figure 5.1. Terrestrial mobile mapping system used in this thesis... 147

Figure 5.2. Synchronization process and data storage for the mobile mapping system 148

Figure 5.3. Illustration of point positioning of a directly geo-referenced multi-unit LiDAR system 149

Figure 5.4. Calibration test field with hut-shaped targets and highly reflective boards 150

Figure 5.5. Flowchart of the lane width estimation strategy, (a) input LiDAR point cloud (colored by height), (b) extracted road surface, (c) candidate lane marking points, (d) derivation of lane marking centerline, and (e) lane width vs. longitudinal distance plot 151

Figure 5.6. Illustration of the various thresholds for road surface extraction: the height threshold (h_{IMU}) and the lateral distance threshold (d_w) 153

Figure 5.7. Estimation of lane marking centerline procedure (a) candidate lane marking points, (b) region growing-based clustering, (c) cluster partition, (d) outlier removal: RANSAC-based, (e) outlier removal: trajectory-based, (f) lane marking centerline, (g) down-sampled lane marking centerline.. 157

Figure 5.8. Lane width estimation procedure: (a) illustration of opposite-side lane marking centerlines, (b) lane width derivation, (c) interpolation among lane marking centerlines, and (d) inaccuracy in lane width estimation without centerline interpolation 159

Figure 5.9. Non-continuous lane markings: dash line (gray points: lane marking points, blue points: lane marking centerline, green points: interpolated points)... 159

Figure 5.10. Location of test datasets and their trajectory (red): (a) interstate highway, (b) U.S. highway, and (c) main arterial 161

Figure 5.11. Lane marking points and derived lane marking centerline: (a) lane marking points (green: using accurate mounting parameters and orange: using inaccurate mounting parameters);(b) lane marking points (green) and derived centerline (blue) using accurate mounting parameters; and (c) lane marking points (orange) and derived centerline (blue) using inaccurate mounting parameters.. 167

Figure 5.12. Comparison of lane width estimates using accurate and inaccurate values of the mounting parameters for the 2017/02/11 dataset from U.S. Highway 168

Figure 5.13. Comparison of estimated lane width from VLP16 and HDL32E for the 2016/11/30 dataset along the U.S. highway 169

Figure 5.14. Anomalies in the marked area of Figure 5.11: (a) candidate lane marking points (gray) and the derived centerlines (green) from VLP16; (b) candidate lane marking points (gray) and the derived centerlines (red) from HDL32E; and (c) candidate lane marking points from VLP16 (gray), the interpolated centerline from VLP16 (blue), and the derived centerline from HDL32 (red). 170

Figure 5.15. Interpolation of straight portion: (a) candidate lane marking points (gray) and the derived centerlines (green) from VLP16; (b) candidate lane marking points (gray) and the derived centerlines (red) from HDL32E; and (c) candidate lane marking points from VLP16 (gray), the interpolated centerline from VLP16 (blue), and the derived centerline from HDL32 (red)....... 171

Figure 5.16. Comparison of estimated lane width from two datasets for the interstate highway work zone area 172

Figure 5.17. Anomalies in the 2016/11/05 dataset: (a) RGB image and (b) extracted lane marking points (gray) and derived lane marking centerline (red)... 172

Figure 5.18. Image from the 2017/05/02 dataset for the area that had poor lane markings in the 2016/11/05 dataset (Figure 16(a))... 173

Figure 5.19. Comparison of estimated lane width from three datasets (collected on 2016/11/30, 2017/02/11, and 2017/08/17) for the U.S. highway segment 173

Figure 5.20. Road intersection area resulting in the spike in Figure 5.19: (a) the captured image at the intersection, and (b) the extracted lane marking points (pink) and interpolated lane marking centerline (green) 175

Figure 5.21. Extracted lane marking points (gray) and derived lane marking centerline (green), as well as estimated lane width (unit: ft) using the proposed strategy.. 176

Figure 5.22. On-site manual measurements of lane width for the main arterial segment 176

Figure 5.23. Comparison of a derived lane width from the proposed strategy and a manually digitized centerline and a ground truth lane width... 177

LIST OF ABBREVIATIONS

Abbreviation	**Definition**
3D	Three-dimensional
ALS	Airborne Laser Scanners
BP	Breakpoint Detection
DBM	Digital Building Model
DIM	Digital Image Matching
DSM	Digital Surface Model
DTM	Digital Terrain Model
FAST	Feature from Accelerated Segment Test
FHWA	Federal Highway Administration
FOV	Field of View
GNSS/INS	Integrated Global Navigation Satellite Systems and Inertial Navigation Systems
ERMBR	Enhanced Recursive Minimum Bounding Rectangle
LiDAR	Light Detection and Ranging
LPD	Local Point Density
LPS	Local Point Spacing
LSA	Least Squares Adjustment
MBC	Minimum Bounding Circle
MBR	Minimum Bounding Rectangle
MMS	Mobile Mapping System

MTLS	Mobile Terrestrial Laser Scanners
NCC	Normalized Cross-Correlation
NHAP	National Human Activity Pattern Survey
OEEPE	Organisation Europeenne d'Etudes Photgrammetriques Experimentales
PCA	Principal Component Analysis
QC	Quality Control
RANSAC	Random Sample Consensus
RMBR	Recursive Minimum Bounding Rectangle
RMSE	Root Mean Square Error
SGM	Semi-Global Matching
STLS	Stationary Terrestrial Laser Scanners
SVM	Support Vector Machine
TIN	Triangular Irregular Network
TMMS	Terrestrial Mobile Mapping System

ABSTRACT

Author: Lin, Yun-Jou. PhD
Institution: Purdue University
Degree Received: December 2018
Title: Point Cloud-Based Analysis and Modelling of Urban Environments and Transportation Corridors
Major Professor: Ayman Habib

3D point cloud processing has been a critical task due to the increasing demand of a variety of applications such as urban planning and management, as-built mapping of industrial sites, infrastructure monitoring, and road safety inspection. Point clouds are mainly acquired from two sources, laser scanning and optical imaging systems. However, the original point clouds usually do not provide explicit semantic information, and the collected data needs to undergo a sequence of processing steps to derive and extract the required information. Moreover, according to application requirements, the outcomes from the point cloud processing could be different. This dissertation presents two tiers of data processing. The first tier proposes an adaptive data processing framework to deal with multi-source and multi-platform point clouds. The second tier introduces two point clouds processing strategies targeting applications mainly from urban environments and transportation corridors.

For the first tier of data processing, the internal characteristics (e.g., noise level and local point density) of data should be considered first since point clouds might come from a variety of sources/platforms. The acquired point clouds may have a large number of points. Data processing (e.g., segmentation) of such large datasets is time-consuming. Hence, to attain high computational efficiency, this dissertation presents a down-sampling approach while considering the internal characteristics of data and maintaining the nature of the local surface. Moreover, point cloud segmentation is one of the essential steps in the initial data processing chain to derive the semantic information and model point clouds. Therefore, a multi-class simultaneous segmentation procedure is proposed to partition point cloud into planar, linear/cylindrical, and rough features. Since segmentation outcomes could suffer from some artifacts, a series of quality control procedures are introduced to evaluate and improve the quality of the results.

For the second tier of data processing, this dissertation focuses on two applications for high human activity areas, urban environments and transportation corridors. For urban environments, a new framework is introduced to generate digital building models with accurate right-angle, multi-orientation, and curved boundary from building hypotheses which are derived from the proposed segmentation approach. For transportation corridors, this dissertation presents an approach to derive accurate lane width estimates using point clouds acquired from a calibrated mobile mapping system. In summary, this dissertation provides two tiers of data processing. The first tier of data processing, adaptive down-sampling and segmentation, can be utilized for all kinds of point clouds. The second tier of data processing aims at digital building model generation and lane width estimation applications.

1. INTRODUCTION

Three-dimensional (3D) spatial data processing has become a critical task for a continually increasing variety of applications, such as urban planning and management, as-built mapping of industrial sites, change detection, autonomous driving, and road safety inspection. Data collection and processing indeed are the most time-consuming and prominent steps in all the above applications and subsequent processes. In the early stages of development, 3D spatial data were collected through on-site measurement and investigation or were generated manually using stereoscopic plotting instruments. Due to new technological advances, the 3D world can be quickly reconstructed through two primary sources, laser scanning and optical imaging systems, which are capable of rapidly acquiring and deriving dense 3D point clouds. Laser scanners directly acquire high-precision point clouds along object surfaces in an efficient manner. Optical imaging systems, on the other hand, provide spectral information, high spatial resolution, and dense point clouds from the captured imagery through photogrammetric space intersection. More specifically, point clouds are derived from overlapping images after conjugate points in such imagery are identified, which are established through modern dense image matching strategies.

To allow for the derivation of semantic information, image and laser-based point clouds must undergo a sequence of data processing steps to meet the demands of Digital Building Model (DBM) generation, urban planning [1], as-built mapping of industrial sites, cultural heritage documentation [2], and road inventory.

A variety of techniques (e.g., down-sampling, segmentation, DBM generation, and feature extraction) have been introduced over time for point cloud processing. Down-sampling is important because the derived point clouds from laser scanners and image-based dense-matching techniques usually include a large number of points. However, data processing (e.g., segmentation and 3D modeling) of such huge datasets is time-consuming and may not be necessary. In order to attain high computational efficiency while maintaining the characteristics of the local surface (i.e., the planar, linear/cylindrical, and rough characteristics), an appropriate down-sampling is necessary. Point cloud segmentation based on pre-defined criteria is one of the initial steps in the data processing chain. Furthermore, since point clouds come from a variety of resources (e.g., Airborne Laser Scanners – ALS, Stationary Terrestrial Laser Scanners – STLS, Mobile Terrestrial Laser Scanners – MTLS, and Digital Image Matching – DIM data), the internal characteristics of the data are different and should be considered in the segmentation process to avoid serious artifacts in the segmentation outcomes [3]. A solid segmentation technique should be able to generate different kinds of objects (e.g., planar, pole-like, and rough) from a given point cloud to ensure the validity and reliability of the derived 3D models. Due to the complexity of the real world, a comprehensive segmentation technique may not exist and the segmentation outcomes could have some artifacts. In this situation, a quality control procedure of the segmentation outcome is an essential step to improving and measuring the quality of the results.

Since application environments are diverse, the nature of the desired outputs from point clouds also can differ. For urban scenery, buildings are among the most prominent features. National Human Activity Pattern Survey (NHAPS) indicated that Americans spend 86.9% of their time indoors (e.g., residences, offices, restaurants, and other indoor locations), plus another 5.5% inside a vehicle [4]. In addition, Organisation Europeenne d'Etudes Photgrammetriques Experimentales (OEEPE) conducted a survey on 3D city models to investigate the feasibility of using 3D models [5]. Their results showed the most important objects of interest according to the users were buildings (95%), traffic networks (76%), and vegetation (71%).

Building model reconstruction is critical for describing the 3D world in urban environments. Moreover, it is useful to model buildings in urban areas because rapid urbanization can cause social and environmental problems in city planning and development. Access to accurate DBMs could assist the urban planning process and reduce the cost to city management. A variety of building reconstruction techniques have been proposed in the literature. However, the majority of these techniques have certain limitations/assumptions to their building models (i.e., only allow for right-angle building models or building models bounded by contiguous straight lines) [6]–[8]. In order to reconstruct more realistic building models from point clouds, not only should the above assumptions be considered but more of the elements of buildings as well, such as the curved boundaries. Moreover, the topology between connected building models should be considered.

For transportation corridors, road digital maps including road characteristics (e.g., lane marking, lane width, slope, curvature, clothoid, shoulder width, and shoulder barriers) are useful for driver assistance systems, road safety inspection, traffic accident reduction, and infrastructure monitoring. Road characteristics are more critical in work zones since the lanes in such areas undergo frequent alterations over the lifetime of a project, thus increasing the risk of crashes [9]. The Federal Highway Administration (FHWA) reported that there were an estimated 96,626 crashes in work zones in 2015, 642 of which involved at least one fatality [10]. Efficient work zone monitoring and inspection is one of the critical tasks in decreasing traffic accidents. For example, lane width is an essential aspect of road safety inspection in work zones, and traffic congestion further increases the probability of severe accidents. In the past, collecting geospatial data for building digital road maps was an expensive, time-consuming, and labor-intensive task and exposed the field crew collecting data to dangerous road traffic. When a lane is narrow, it also results in reduced roadway capacity. Using Terrestrial Mobile Mapping Systems (TMMS) equipped with Light Detection and Ranging (LiDAR), geo-referenced point clouds can be collected rapidly in work zone areas without affecting traffic. Then, an accurate lane width can be derived from the acquired point cloud for risk assessment.

This thesis introduces two tiers of point cloud processing. The first tier proposes an adaptive data processing framework (i.e., adaptive down-sampling and multi-class simultaneous segmentation and quality control) to deal with multi-source and multi-platform point clouds. Due to the increasing demand for a variety of applications, point cloud processing has become more critical to meet their requirements. The second tier introduces two point cloud processing strategies targeting applications related to high human activity areas, namely, urban environments and transportation corridors. In the remaining sections of this chapter, the problem statement, research objectives, and structure of the thesis are discussed.

1.1 Problem Statement

Although several researchers have studied and developed various point cloud processing techniques, challenges remain that need to be addressed in order to accomplish more accurate, robust, and reliable outcomes. Furthermore, identifying problems and creating new applications should continue using 3D datasets. Some of the critical problems in point cloud down-sampling, segmentation, digital building model generation, and lane width estimation are summarized below.

1.1.1 Challenges in Point Cloud Down-sampling

The state-of-the-art mapping technologies can rapidly provide dense point clouds. However, processing (e.g., registration, segmentation, and reconstruction) such large datasets continues to be time-consuming and unnecessary. Some of the down-sampling techniques do not consider point density, point distribution along physical surfaces, and point cloud noise. Such techniques may down-sample point clouds based on the distance to the laser scanner (range-based) or the distance between two points (distance-based). Image-matching-based point clouds cannot be down-sampled using the range-based method. Distance-based down-sampling can cause problems in further processing activities (e.g., segmentation). One should expect that the down-sampling for flat ground should behave differently from the down-sampling of a utility pole with power lines since the point cloud for latter includes more information for feature modeling. Therefore, a new adaptive down-sampling approach is needed to thin point clouds while considering the nature of the local surface (i.e., planar, linear/cylindrical, and rough features) and variations in local characteristics (e.g., Local Point Density – LPD and noise level). Moreover, the approach should be suitable for both laser scanning and image-based point clouds.

1.1.2 Challenges in Segmentation Technique

Point cloud segmentation, which is one of the primary tasks in point cloud processing, extracts and partitions objects according to the spatial proximity and similarity of the local attributes. Since point clouds can come from a variety sources/platform (e.g., ALS, STLS, MTLS, and image-matching strategies), an effective segmentation technique should consider the internal characteristics of point clouds. Moreover, the performance of a segmentation technique is sensitive to the noise level within the point cloud in question [11]. In order to overcome these problems, an adaptive segmentation procedure, which considers the internal characteristics of the data (i.e., LPD and noise level) as well as the quality of the seed point/region, is needed to derive better segmentation outcomes. The outcome of a segmentation process usually suffers from some artifacts, such as over-segmentation, under-segmentation [12], and misclassification. Most of the segmentation techniques do not include a Quality Control (QC) procedure to evaluate and improve the segmentation results. A succession of QC procedures is also needed to identify the potential problems and to deal with various kinds of artifacts.

1.1.3 Challenges in Digital Building Model Generation

For urban environments, building model generation is the driving force in cyber/smart city efforts. The strategies to reconstruct a DBM from point clouds can be classified into model-driven and data-driven. The model-driven approach requires a model library first, and the building models then are combinations of those existing models. This approach is preferred when the area in question has many buildings with similar roof types; however, it can be challenging for complex buildings since it is difficult to include every building model in the library. The data-driven approach can reconstruct any kind of building model, but its performance may be sensitive to the noise level within a given point cloud. Moreover, in order to derive reasonable building models, most of the research may utilize some constraints/assumption to regularize the building models. For example, building boundaries may consist of straight lines, right angles, or parallel lines. In this case, however, some of the buildings, such as buildings with curved boundaries or multi-orientation buildings, cannot be described. Furthermore, some techniques do not consider the topology relationship between connected building sub-blocks. Therefore, an innovative DBM reconstruction technique is needed to consider the following items. First, a building model generation strategy should consider both straight-line and curved boundaries. Second, most of the buildings have right-angle corners so the right-angle constraint should be maintained while being capable of dealing with multi-orientation buildings. Finally, the topology between connected building sub-blocks should be considered.

1.1.4 Challenges in Lane Width Estimation

For transportation corridors, improving road safety is a critical task for decreasing accidents. In road safety inspection, lane width evaluation is one of the critical inspection items. In the past, collecting geospatial data for road safety inspection was an expensive, time-consuming, and labor-intensive task. Using TMMS equipped with LiDAR units and cameras, geo-referenced point clouds and images can be collected rapidly and remotely in work zone areas without affecting traffic. The imagery can provide information in color to ease the detection of features of interest along a road surface; but the difficulty of feature extraction from imagery greatly depends on illumination conditions, camera exposure, obstacles, and shadows. Moreover, cameras cannot provide sufficient information in all circumstances since they are pointed in a fixed direction and the overlap area between successive images could be limited by the occlusions arising from traffic congestion in work zones. LiDAR units can scan forward and backward, however, thus mitigating the occlusion problem and also can be operated day and night. Hence, this thesis only focuses on feature identification and extraction from LiDAR point clouds.

The research to date related to road feature identification focused on road extraction, curb identification, and lane marking extraction and identification, while only a few papers discussed lane width estimation. Holgado-Barco et al. [13] discussed lane width estimation, which requires intensity as well as raw measurements (i.e., timestamp and scan angle information) and is only suitable for a 2D laser scanner. Therefore, a new approach which only requires a 3D point cloud along with the intensity information is needed.

1.2 Research Objectives

The main objectives of this thesis can be summarized as follows:

- General Research Objectives

a. Introduce an adaptive down-sampling strategy for managing large datasets obtained/derived from laser scanning and optical camera systems while considering the nature of the local surface (i.e., planar, linear/cylindrical, and rough features), variations in local characteristics (i.e., LPD), and the noise level of the whole dataset.

b. Propose a multi-class simultaneous adaptive segmentation approach to simultaneously derive planar and varying-radii pole-like segments and then group the remaining points as rough segments. This approach should be suitable for multi-platform/multi-source point cloud data.

c. Introduce a succession of QC measures to deal with misclassified points in the proposed segmentation approach.

d. Consider different application environments of point clouds by addressing the needs of the following:

  - For urban environments, a new approach is proposed to regularize building boundary and generate seamless building models using ALS data.
  - For highway corridors, a framework is provided to derive accurate lane width using LiDAR-based TMMS for risk assessment.

These general objectives will be achieved through the following specific objectives:

- Specific Research Objectives

a. Implement an adaptive down-sampling strategy to remove redundant points within planar neighborhoods. The main characteristics of the proposed approach are as follows:

- The adaptive down-sampling strategy considers the planar, linear/cylindrical, and rough feature categories within a point cloud. For object modeling, a high point density in planar regions may not be necessary. Hence, redundant points will be selectively removed in those regions. During point cloud acquisition, incomplete point clouds are usually obtained that partially represent linear/cylindrical and rough surfaces. Therefore, within the proposed procedure, the points along linear/cylindrical and rough local neighborhoods are maintained.
- The procedure considers variations in local characteristics, such as the LPD, of a point cloud and allows for selective removal of points in high-density planar regions.
- The strategy introduces two Principal Component Analysis (PCA) approaches that can identify the nature of local neighborhoods while considering the noise level within the constituents of a point cloud.
- A probability-based test is adopted to adaptively down-sample and remove points using the evaluated LPD within planar local neighborhoods.
- The proposed strategy is suitable for both laser scanning and image-based point clouds.

b. Present a region-growing approach for the segmentation of planar, pole-like, and rough features. The main characteristics of the proposed approach are as follows:

   - Planar and varying-radii pole-like features are simultaneously segmented.
   - ALS, STLS, MTLS, and DIM point clouds can be manipulated by the proposed segmentation procedure.
   - The region-growing process starts from optimally-selected seed regions to reduce the sensitivity of the segmentation outcome to the choice of the seed location.

  - The region-growing process considers variations in the local characteristics of the point cloud (i.e., LPD and noise level).

c. Propose a new QC procedure for improving and evaluating the segmentation outcomes. The main characteristics of the proposed approach are as follows:

  - The QC process considers possible competition among neighboring planar and pole-like features for the same points.
  - The QC procedure considers possible artifacts arising from the sequence of the region growing process.
  - The QC process considers the possibility of having partially or fully misclassified planar and pole-like features.

d. Propose a building model generation strategy which can reconstruct complex building models with several characteristics, including right-angle, multi-orientation, and curved boundaries using point clouds. The main characteristics of the proposed approach are as follows:

  - An Enhanced Recursive Minimum Bounding Rectangle (ERMBR) approach, which includes a simultaneous Recursive Minimum Bounding Rectangle (RMBR) [7] and Least Squares Adjustment (LSA) procedure as well as a generalized RMBR procedure, is proposed to regularize boundaries with right angle and single orientation.
    - o The proposed simultaneous RMBR and LSA procedure is not sensitive to the thresholds. This is achieved by including the following steps: (1) a spike removal method is proposed to eliminate outliers in the extracted boundaries;

    (2) LSA is adopted to avoid identifying artificial details; and (3) the variation of LPD in every building hypothesis is considered.

    - The proposed a generalized RMBR procedure can successfully reconstruct all right-angled buildings accurately, which was not fully accomplished by previous work.

  - A boundary regularization strategy using a breakpoint detection (BD) approach is introduced to regularize boundaries with multi-orientation and curved edges.
  - A strategy for integration of the outcome of ERMBR and BD approaches is proposed to maintain right-angle corners, to reconstruct multi-orientation buildings, and to generate curved boundaries.
  - A strategy for water-tight DBM generation is proposed to consider the topology between adjacent building primitives and to provide a more realistic visualization of the building models.

e. Propose a framework to derive accurate lane width using in-house developed mobile mapping system. The main characteristics of the proposed approach are as follows:

  - A robust approach is proposed to derive lane width from 3D point clouds. The proposed framework is tested using road segments over tens of miles to show its robustness.
  - The proposed framework is not limited to a specific sensor model and does not require raw measurements (e.g., scan angles and time stamps). The required information for the proposed lane width estimation framework is only the trajectory information and the 3D point cloud along with intensity information.
  - Four validation strategies are proposed –

- An analysis of the impact of the mounting parameters' quality on lane width estimation is conducted.
- The proposed strategy is applied to datasets collected using two types of laser scanners to compare the quality of derived lane width estimates.
- To demonstrate the precision of estimated lane width and the reliability of the introduced calibration approach, an analysis of multiple datasets collected over the same test area from different seasons is conducted.
- To show the accuracy of the proposed lane width estimation strategy, the derived lane width estimates are compared to the on-site manual measurements and estimates from manually digitized lane markings.

Figure 1.1 is a flowchart of the proposed framework for point cloud processing and two chosen application field in urban environments and transportation corridors.

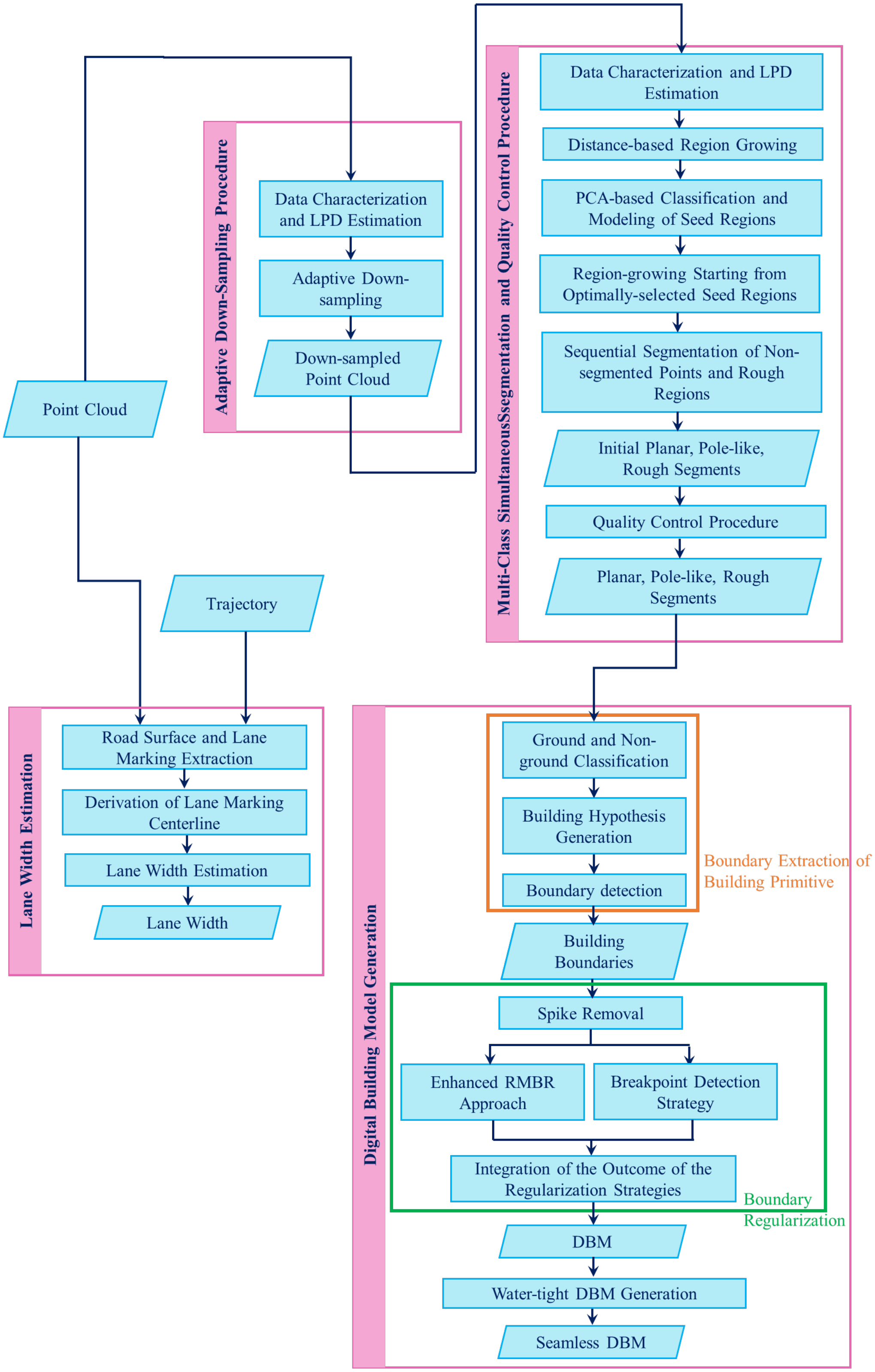

Point Cloud
Adaptive Down-Sampling Procedure
Data Characterization and LPD Estimation
Adaptive Down-sampling
Down-sampled Point Cloud
Multi-Class SimultaneousSsegmentation and Quality Control Procedure
Data Characterization and LPD Estimation
Distance-based Region Growing
PCA-based Classification and Modeling of Seed Regions
Region-growing Starting from Optimally-selected Seed Regions
Sequential Segmentation of Non-segmented Points and Rough Regions
Initial Planar, Pole-like, Rough Segments
Quality Control Procedure
Planar, Pole-like, Rough Segments
Trajectory
Lane Width Estimation
Road Surface and Lane Marking Extraction
Derivation of Lane Marking Centerline
Lane Width Estimation
Lane Width
Digital Building Model Generation
Ground and Non-ground Classification
Building Hypothesis Generation
Boundary detection
Boundary Extraction of Building Primitive
Building Boundaries
Spike Removal
Enhanced RMBR Approach
Breakpoint Detection Strategy
Integration of the Outcome of the Regularization Strategies
Boundary Regularization
DBM
Water-tight DBM Generation
Seamless DBM

Figure 1.1. Flowchart of the proposed two-tier data processing framework

1.3 Dissertation Outline

The remainder of this dissertation presents the proposed framework in more detail.

- Chapter 2 reviews the existing approaches for point cloud generation, down-sampling techniques, segmentation methodology, digital building model reconstruction approaches, and accurate lane width estimation.
- Chapter 3 introduces the framework of the first tier point cloud processing, which includes an adaptive down-sampling strategy and multi-class segmentation approach as well as a quality control process for the segmentation outcomes.
- Chapter 4 introduces the second tier of data processing in urban environments. A strategy is proposed to generate complex building models from airborne LiDAR point cloud data.
- Chapter 5 presents the second tier of data processing in transportation corridors. A framework for deriving accurate lane width, including system development, system calibration, lane width estimation, and validation using four experiments, is introduced.
- Chapter 6 presents the key contributions of this thesis and recommendations for future research work.

2. LITERATURE REVIEW

2.1 Introduction

The primary objective of this thesis was to develop a new framework for the first tier of point cloud processing (i.e., adaptive down-sampling and multi-class simultaneous segmentation) and the second tier of data processing for urban environment and transportation corridors (i.e., digital building model reconstruction and accurate lane width estimation). The point cloud sources can come from optical image and laser scanning systems, which will be reviewed in Section 2.2. The point clouds generated from the current techniques usually include an excessive number of points, data processing (e.g., segmentation and registration) of these huge data is time-consuming and unnecessary. Previous research efforts that focused on data down-sampling will be discussed in Section 2.3. Segmentation is one of the initial and critical steps in the first tier of data processing to derive the semantic meanings from point clouds. A review of existing segmentation techniques and quality control procedures will be addressed in Section 2.4. The second tier of data processing includes two applications related to high human activity areas, urban environments and transportation corridors. For urban environments, the existing pertinent literature for building reconstruction is presented in Section 2.5. For transportation corridors, the state-of-the-art road feature extraction techniques are discussed in Section 2.6.

2.2 Point Cloud Generation

Optical imagery and laser scanners are the two major sources for indirectly or directly deriving point clouds, both of which can meet the demands of the intended 3D modeling applications. The imagery captured by optical sensors can provide spectral information, high spatial resolution, and point clouds through an intersection procedure. On the other hand, laser scanners are capable of directly acquiring high precision point clouds along object surfaces in an efficient manner. Point cloud generation from optical images and laser scanning systems is reviewed in Section 2.2.1 and Section 2.2.2.

2.2.1 Image-based Point Cloud

Electro-Optical sensors onboard space-borne, airborne, and terrestrial platforms are capable of acquiring imagery with high resolution that could be used for point cloud generation. Identification of conjugate points in overlapping images is a key prerequisite for image-based point cloud generation. The photogrammetric community has used area-based and feature-based matching techniques [14]. Area-based image matching is performed by comparing the gray values within a defined template in one image to those within a larger search window in an overlapping image to identify the location that exhibits the highest similarity. Pratt [15] proposed the Normalized Cross-Correlation (NCC) measure, which compensates for local brightness and contrast variations between the gray values within the template and search windows. Feature-based matching, on the other hand, compares the attributes of extracted features (e.g., points, lines, and regions) from overlapping images. Feature from Accelerated Segment Test (FAST) [16] and the Harris descriptor can be used to detect corners as point features. The Scale Invariant Feature Transform (SIFT) detector and descriptor can be used to identify and provide the attributes for key image points [17]. The SIFT descriptor then can be used to identify conjugate point features in overlapping images. Alternatively, Canny edge detection and linking can be used to derive linear features from imagery [18]. Then, generalized Hough Transform can be used to identify conjugate points along the detected edges [19]. The 3D coordinates of conjugate points and linear features, which can be used for a wide range of applications such as building model generation [20], can be derived through a photogrammetric space intersection. However, both area-based and feature-based image matching techniques are incapable of providing a detailed object description, which are needed for 3D object modeling (i.e., they are mainly used for automated recovery of image orientation). Compared to area-based and feature-based matching techniques, recently developed dense-matching algorithms can provide precise point clouds with high-density through a global matching constraint [21]. Hirschmuller [22] proposed Semi-Global Matching (SGM), which performs pixel-wise matching using mutual information. Haala [23] showed that pixel-wise dense matching and the current software tools are capable of generating high definition landscape Digital Surface Model (DSM) from airborne imagery.

2.2.2 Laser-based Point Cloud

In contrast to imaging sensors, laser scanners can directly derive dense point clouds. Depending on the used platform, a laser scanner can be categorized into ALS, STLS, and MTLS. ALS was developed in the early 1990s, and is mainly used for collecting surface data over large areas. The level of detail in ALS data depends on the flying height, pulse rate, scanning rate, aircraft speed, FOV, among other parameters. The point density for ALS data usually ranges from 1 to 40 pts/m^2 [24] which is suitable for DSM and Digital Terrain Model (DTM) generation as well as rough building model generation [7], [25]. Due to the nature of the data acquisition mechanism, ALS systems cannot provide the necessary details for extracting objects that do not belong to building rooftops and terrain such as building facades, light poles, trees, and fences. As a result of their proximity to the objects of interest, STLS and MTLS systems can acquire dense point clouds for the extraction and modeling of building facades, fences, trees, and light-poles. STLS systems were introduced almost 15 years ago with the improvement in modern terrestrial direct geo-referencing systems. MTLS systems were developed in the late 1980s and early 1990s

[26]–[28]. In the early stage, MTLS systems usually carried a set of optical cameras and a geo-referencing system (integrated Global Navigation Satellite Systems and Inertial Navigation Systems – GNSS/INS), which is capable of providing an accurate position and orientation of the platform at high frequency [29]. With the emergence of laser scanners, modern Mobile mapping Systems (MMSs) usually carry both laser scanners and optical cameras as well as a geo-referencing system. MMSs equipped with LiDAR units are capable of directly acquiring high precision point clouds from object surfaces in an efficient manner.

2.3 Existing Down-sampling Techniques

Derived point clouds from ALS, STLS, and MTLS systems as well as image-based dense-matching techniques usually include an excessive number of points. Processing (e.g., registration, segmentation, and reconstruction) such huge datasets is time-consuming and unnecessary [30]–[32]. For example, representing a planar surface only requires a few points to define a reliable surface geometrically and spatially. Thus, points along high-density planar surfaces could be redundant. The point density can either be reduced during the data collection or processing stages. During the data collection stage, density reduction can be achieved by decreasing the scanning resolution (i.e., lowering pulse repetition frequency and/or scan rate). However, this approach will reduce the point density throughout the entire area (i.e., areas farther from the scanning system will suffer from significant loss of detail). Alternatively, the collected point cloud can be thinned through data processing. Software tools are available for point cloud thinning, commonly known as down-sampling (e.g., "CloudCompare" [33], which includes down-sampling functions such as random down-sampling and point-spacing down-sampling). For random down-sampling, the software randomly removes points to achieve the designed number of points. Although it is a quick way to down-sample a dataset, it may lose the most information from the point clouds. For the point-spacing down-sampling method, it considers the minimum distance between two points. When the distance between two points is less than the designed threshold, the point would be removed. However, this approach only considers the distance between two points without considering the nature of the local surface, which could be changed after down-sampling. Cabo et al. [34] reduced the size of a point cloud through regular voxelization of 3D space and then processed the reduced data to extract pole-like features. Another approach is to use octrees to remove the redundant points based on the cubic size [35]. Mandow et al. [36] proposed a range-independent down-sampling method to uniformly thin the STLS data according to their spherical scanning characteristics. A given point cloud is down-sampled based on the desired angular resolution instead of the distance to the scanner. Puttonen et al. [37] proposed two methods for range-based down-sampling of point clouds based on the point-to-scanner distance.

None of the above techniques consider the point distribution along the physical surfaces and the characteristics of the local neighborhoods encompassing such points. In addition, the range-independent and range-based down-sampling strategies are only suitable for STLS and MTLS datasets (i.e., they cannot be applied to image-based point clouds). Therefore, an optimal down-sampling strategy is needed that can consider the physical surface characteristics (i.e., planar, linear/cylindrical, and rough surface), internal characteristics (e.g., local point density) and the 3D distribution of the point clouds rather than merely considering the point-to-point or point-to-scanner distances.

2.4 Existing Segmentation Techniques

Point-cloud-based object modeling usually starts with a segmentation process to categorize the data into subgroups that share similar characteristics. Segmentation approaches can generally be classified as being either spatial or parameter domain. The spatial-domain segmentation and parameter-domain segmentation are reviewed in Section 2.4.1 and Section 2.4.2, respectively. Because there is variation in the internal characteristics and complexity of scenes, segmentation outcomes may suffer from some artifacts. Therefore, a quality control procedure (Section 2.4.3) is needed to evaluate and improve the segmentation outcomes.

2.4.1 Spatial-domain Segmentation Techniques

For the spatial-domain approach (e.g., region-growing based segmentation), the point cloud is segmented into subgroups according to the spatial proximity and similarity of the local attributes of its constituents. More specifically, starting from the seed points/regions, the region-growing process augments neighboring points using a pre-defined similarity measure. The spatial proximity and local attribute determinations depend on whether the point cloud is represented as a raster, Triangular Irregular Network (TIN), or un-structured set. Rottensteiner and Briese [38] interpolated non-organized point clouds to generate a DSM, which they used to detect building regions through a height and region-growing analysis of a DSM-based binary image. The region-growing process is terminated whenever the RMSE of a plane-fitting process exceeds a pre-set threshold. Forlani et al. [39] used a region-growing process to segment raster elevation data, where the height gradient between neighboring cells was used as the stopping criterion. For TIN-based point clouds, the spatial neighborhood among the generated triangles and the similarity of the respective surface normals have been used for the segmentation process [40]. For non-organized point clouds, data structuring approaches (e.g., Kd-trees or Octree data structures) were used to identify local neighborhoods and derive the respective attributes [41], [42]. Yang and Dong [43] classified point clouds using Support Vector Machine (SVM) into planar, linear, and spherical local neighborhoods. Then, region growing was implemented by checking the similarity of the derived attributes (e.g., principal direction, normal vector, and intensity). Region-growing segmentation approaches are usually preferred due to their computational efficiency. However, their performance is quite sensitive to the noise level within the point cloud in question as well as the selected seed-points/regions [11], [31], [44].

2.4.2 Parameter-domain Segmentation Techniques

For the parameter-domain approach, a feature vector is first defined for the individual points using their local neighborhoods. Then, the feature vectors are incorporated in an attribute space/accumulator array where peak-detection techniques are used to identify clusters (i.e., points sharing similar feature vectors). Filin and Pfeifer [45] used a slope-adaptive neighborhood to derive the local surface normal for the individual points. Then, they defined a feature vector that encompassed the position of the point and the normal vector to the tangent plane at that point and used a mode-seeking algorithm to identify clusters in the resulting attribute space [46]. Biosca and Lerma [47] utilized three attributes to define a feature vector: 1) the normal distance to the fitted plane through a local neighborhood from a defined origin, 2) the normal vector to the fitted plane, and 3) the normal distance between the point in question and the fitted plane. Then an unsupervised fuzzy clustering approach was implemented to identify peaks in the attribute space. Lari and Habib [31] introduced an approach where the individual points were classified as either belonging to planar or linear/cylindrical local neighborhoods using PCA. The attributes of the classified features then were stored in different accumulator arrays where the peaks were identified without the need for tessellating the array to detect the planar and pole-like features. Parameter-domain segmentation techniques do not depend on seed points. However, the identification of peaks in the constructed attribute space is a time-consuming process, the complexity of which depends on the dimensionality of the involved feature vector [11]. The spatially-disconnected segments that share the same attributes will be erroneously grouped together as well. In general, the existing spatial-domain and parameter-domain segmentation techniques do not perform simultaneous segmentation of planar, pole-like, and rough regions in a given point cloud.

2.4.3 Quality Control Procedures

The outcome of a segmentation process usually suffers from some artifacts [12]. The traditional approach for QC of the segmentation results is based on having reference data, which is manually generated, and deriving correctness and completeness measures [48], [49]. The correctness measure evaluates the percentage of correctly-segmented constituents of regions in a given class relative to the total size of that class in the segmentation outcome. The completeness measure, on the other hand, represents the percentage of correctly-segmented constituents of regions in a given class relative to the total size of that class in the reference data. The reliance on reference data to evaluate the correctness and completeness measures is a major disadvantage of such QC measures. Past research addressed the possibility of deriving QC measures that are not based on reference data. More specifically, Belton [50], Nurunnabi et al. [51], and Lari and Habib [3] developed QC measures that make hypotheses regarding possible segmentation problems, propose procedures for detecting instances of such problems, and develop mitigation approaches to fix them without the need for reference data. Over-segmentation (where a single planar/pole-like feature is segmented into more than one region) and under-segmentation (where multiple planar/pole-like features are segmented as one region) are key segmentation problems that were considered in prior research. More specifically, the problems associated with planar and pole-like feature segmentation were independently addressed.

However, segmentation problems arising from possible competition among neighboring planar and pole-like features have not been addressed in the past to the best of the author's knowledge. For example, one should expect that segmented regions at an earlier stage might invade segmented regions at a later stage, as shown in Figure 2.1.

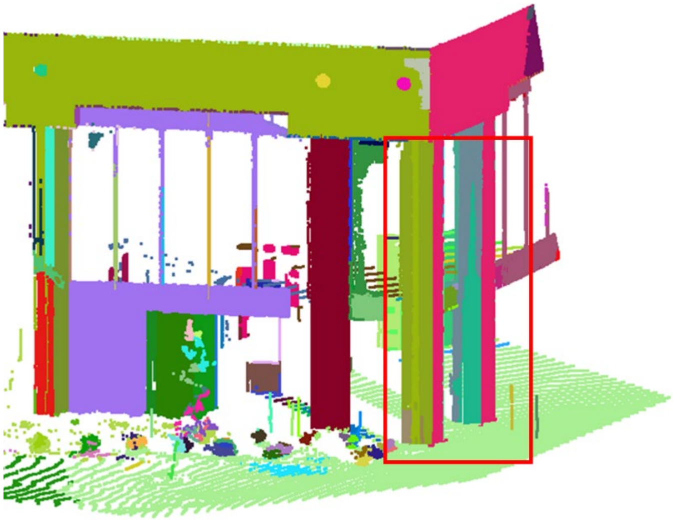


Figure 2.1. Segments invading/invaded and over-segmentation problems

2.5 Urban Environments – DBM Reconstruction

In general, DBM generation can be achieved through two main steps: building detection and building reconstruction [25]. Building detection aims at distinguishing building objects from other objects in the involved dataset for the extraction of building hypotheses or building areas. The second step, building reconstruction, derives 3D building models whose accuracy, level of automation, and complexity depends on the level of details and geometry of the building (i.e., right-angle, multi-orientation, and curved boundary buildings). A great deal of research was conducted to generate building models, which will be addressed in terms of the input data sources and modeling strategies are presented in Section 2.5.1 and Section 2.5.2, respectively.

2.5.1 Data Sources for DBM Generation

The sources available for DBM generation include remote sensing data (i.e., optical imagery and LiDAR data) as well as 2D GIS data (e.g., digital maps and cadastral data). However, remote sensing data are preferred over 2D GIS data as the latter could be out of date, lack roof structure lines, and/or have low vertical accuracy. The existing approaches to building modeling from remote sensing data can be mainly classified into three groups: optical imagery, LiDAR data, and integration of LiDAR data and optical imagery.

For optical imagery, the general method is comprised of image matching and reconstruction. Traditional photogrammetric approaches utilize either area-based [15] or feature-based [17] image matching techniques to identify conjugate points in overlapping images. The 3D coordinates of conjugate features are derived through space intersection, and the building models are reconstructed based on 3D features [52]. However, traditional image matching techniques are not capable of generating dense point clouds and can only support a low level of automation for building model generation. Recently developed dense image matching techniques can generate point clouds with a high level of detail [22], [23]. However, the quality of these dense point clouds depends on the texture of the visible surfaces [53] and the extent of the occluded areas.

In the case of LiDAR, the data from airborne systems is the most suitable for reconstructing the building models according to the shape of the rooftops because of the nature of its data acquisition mechanism. More specifically, ALS systems can directly provide the necessary details that belong to building rooftops and terrain. Compared with imagery, LiDAR data can provide better vertical accuracy as well as rapidly derive 3D surfaces to facilitate automation of building reconstruction [54]. Also, LiDAR data are not influenced much by sun shadow and relief displacement [55]. Due to the errors in attitude estimation in the case of LiDAR data, however, the horizontal accuracy is worse than that of optical imagery. Therefore, an integration of LiDAR data and imagery is developed to combine the accurate height information from LiDAR data with the spectral information and the high planimetric resolution from optical imagery [25], [56], [57].

Awrangjeb et al. [56] proposed a method for 3D roof extraction through the integration of LiDAR data and multispectral ortho-imagery. Cheng et al. [58] integrated airborne LiDAR data and optical multi-view images to segment roof points and automatically reconstruct 3D building models. It is worth noting that fusing more data sources can help ease occlusion difficulties and improve the accuracy of building detection and reconstruction. However, in many cases, auxiliary data sources may not be available. Moreover, the integration of different data sources requires high quality data registration. Inconsistencies between different data sources will decrease the accuracy of DBM generation. In this study, the proposed strategy deals with only LiDAR data. So, the discussions henceforth mainly focus on LiDAR data.

2.5.2 Reconstruction Strategies of DBM Generation

The three mainly used strategies for building reconstruction from LiDAR data are the model-driven, data-driven, and hybrid approaches. The model-driven approaches are top-down processes that start by establishing a model library [6], [59], [60] and then search the most appropriate model for a given building. A building roof model can be composed of either one planar surface or a combination of differently oriented planar surfaces. Each of these surfaces is described using an initial set of parameters, which is then updated using the information derived from existing data. The model-driven methods are preferred when the area in question has many buildings with similar roof types as it would enhance the computational performance and the accuracy of the outcome models. However, representation of complex building roofs can be challenging because it is difficult to include every building model in the model library. Moreover, details in the reconstructed models might be compromised because it is necessary to fit the actual data to the pre-defined building models. Henn et al. [61] proposed a model-driven reconstruction of buildings from sparse LiDAR point clouds. First, the buildings were decomposed into rectangles and the parameters of roof planes were estimated using RAndom SAmple Consensus (RANSAC) approach. Then, Support Vector Machine (SVM) was adopted to identify the most probable roof model (i.e., flat, shed, gabled, hipped, and pyramidal roof).

The data-driven approaches, on the other hand, are bottom-up processes as they do not assume pre-defined models. Therefore, the shape of the generated models is not constrained and can theoretically be arbitrary. However, the data-driven approaches could have some limitations and assumptions, such as straight line, right angle, perpendicular lines, and parallel lines [8], [62], [63]. Moreover, the model's performance may be sensitive to the noise level of a given point cloud. Yan et al. [8] proposed a building reconstruction approach to generate seamless building models using straight-line segments. First, planar roof patches from ALS data were extracted, and the roof patches were projected to 2D plane to derive topology between roof patches. Then, Douglas-Peucker algorithm was utilized to derive simplified regularized boundaries which consisted of straight-line segments. The simplified boundaries were adjusted by applying 2D snake algorithm with parallel and deviation constraints. Shon et al. [63] reconstructed buildings using rectilinear lines whiling considering topology between the roof patches. First, the roof patches were extracted and the rectilinear lines were extracted. The authors proposed a binary space partitioning tree to derive the topology between adjacent linear features for reconstructing seamless building models.

For the third strategy, a mixture of model-driven and data-driven approach (i.e., hybrid approaches) has been adopted in many DBM generation studies in order to exploit the advantages of both approaches and overcome their shortcomings. Sohn and Dowman [57] applied a data-driven method to LiDAR data and IKONOS imagery in order to extract building line cues and then employed a model-driven method to deduce the missed line cues. Most of the model-based approaches, as well as the model-based components of hybrid approaches, adopt a hypothesis-and-test process [64], [65].

For building reconstruction using LiDAR data, most of previous work [59-65] mainly focuses on the building with straight line. Moreover, some of existing approaches may not consider the topology relationship (i.e., step lines and ridge) between the elements of building models [7], [61], [66], [67].

2.6 Transportation Corridors – Lane Width Estimation

The road surface, road markings, and characteristics of roads (e.g., slope, curvature, superelevation, azimuth, and lane width) are crucial factors for road safety inspection, building road digital maps, and autonomous vehicle navigation. Gargoum et al. [68] provided a review of available approaches for the extraction of road features using LiDAR data. They categorized the features into on-road information (road surface, lane markings, and road edge), road side information (traffic signs), road side objects (lamp posts, trees, and utility poles), and geometric information and assessment (road cross-section information, vertical alignment information, pavement condition assessment and monitoring, sight distance assessment, and vertical clearance assessment). Guan et al. [69] reviewed the use of mobile LiDAR for road information inventory. They discussed three classes of road features: 1) road pavements, 2) road surface structures (road markings, manhole covers, and curvilinear pavement cracks), and 3) pole-like objects. Road features extraction (road surface, road markings, and characteristics of roads) are discussed in Section 2.6.1, Section 2.6.2, and Section 2.6.3, respectively.

2.6.1 Road Surface Extraction

For the road surface feature extraction, Han et al. [70] detected road boundaries and obstacles by extracting line segments from the raw point clouds in the polar coordinates. Then, the extracted line segments were classified into road and obstacle segments by analyzing the change in the roll and pitch angles of each segment. Kang et al. [71] used Hough transform to detect curb positions. Then, two Kalman filters were applied to track the curb using a prediction model. At the end, a probabilistic decision-making algorithm was applied to accurately estimate the roadside curb position. Zhang [72] collected 3D point clouds with an LMS-200 SICK sensor mounted on top of a vehicle to detect the road surface and the road surface edges by analyzing the change in elevation. First, the candidate road segments were extracted from the elevation data using a local-extreme-signal detector to identify the road surface edge. Then, the candidate road segment was given a weight based on the standard deviation of the elevation and was classified using linear SVM to decide whether the candidate was a road segment or not. Finally, the points representing the curb were detected by checking whether or not the line segments representing their projections onto the ground plane were perpendicular to the road surface. Kumar et al. [73] utilized a Riegl VQ-250 laser scanner and an IXSEA LandINS GNSS/INS onboard an MMS to collect point clouds and extract road surfaces using a combination of two modified versions of the parametric active contour or snake model. The snake model was initialized based on the navigation information obtained from the GNSS/INS.

2.6.2 Lane Marking Extraction

Transportation agencies apply highly reflective glass beads to lane markings to improve their visibility at night. Therefore, well-maintained retro-reflective lane markings along the extracted road surface will be manifested as high-intensity points in LiDAR point clouds. Although the material used for road markings can provide high-intensity return, the reflected laser pulse intensities fluctuate strongly according to the incident angles and range between the laser beam firing point and its footprint. Kumar et al. [74] used a set of range-dependent thresholds to extract lane markings. First, the road surface was extracted and divided into blocks along the driving direction based on lateral distance from the navigation data representing the trajectory; and different thresholds then were applied to different blocks for road marking extraction. Finally, the morphological operations and generic knowledge of the dimensions of the road markings were applied to complete the shapes of the extracted road markings and remove the noise.

Yu et al. [75] proposed a multi-segment threshold to mitigate the effects of the intensity variation of point clouds acquired from Riegl VMX-450. First, the road surface was partitioned into blocks along the trajectory direction and each block was partitioned into a set of segments according to the lateral distance from the trajectory. The road markings in each segment were detected using Otsu's thresholding algorithm. The incorrectly classified points were recognized and removed by calculating their local point density. Finally, the road marking points were classified into specific categories using the following steps: (1) Euclidean distance grouping, (2) voxel-based normalized cut segmentation strategy for dividing a group including several types of markings, and (3) marking classification using trajectory information, deep learning models, and PCA.

Guan et al. [76] collected point clouds from a Riegl VMX-450 MMS system and extracted lane markings using multiple thresholds. First, a curb-based strategy that relied on navigation data was applied to extract the points belonging to the road surface by detecting the elevation changes along the road profile. Next, the points representing the road surface were interpolated into a geo-referenced intensity image. Then, road markings were segmented using multiple thresholds that correspond to different ranges as determined by point density. Finally, a morphological operation was applied to complete the shape of the road marking and remove the noise.

Cheng et al. [77] proposed a road marking extraction and classification strategy. The tested point clouds were acquired from a RIEGL VMX-450 system. First, the non-ground points were removed and the road surface points were rasterized into a 2D intensity image. The road markings were extracted using Otsu's thresholding method. The extracted road markings were partitioned into segments using region growing strategy. Finally, the segments were classified using a decision tree with the help of the derived geometric attributes. Yang et al. [78] proposed a binary kernel descriptor to extract the road information from mobile laser scanning point clouds. Binary kernel descriptor encodes the shape and intensity information to detect curbs and road markings.

2.6.3 Characteristics of Road Extraction

The characteristics of a road include slope, curvature, superelevation, azimuth, and lane width. Cai and Rasdorf [79] modeled road centerlines and predicted their length using LiDAR and planimetric road centerline data. The LiDAR data (with a point density of approximately 0.031 pts/m^2) were obtained from the North Carolina Floodplain Mapping Program and the planimetric road centerline data were obtained from the GIS unit of North Carolina Department of Transportation (NCDOT). A snapping and interpolation approach was used to obtain the 3D points along the road centerlines. Holgado-Barco et al. [80] used PCA to analyze point clouds captured from an Optech LYNX mobile mapper for deriving road parameters (slope, vertical curves, and superelevations) from a segmented cross-sectional road surface extracted with the help of the trajectory and setting the scan-angle threshold. The slope and superelevation were computed using

PCA for each road segment. After deriving the slope, the characteristics of vertical curves were estimated.

Holgado-Barco et al. [81] extracted lane markings and derived the geometric design of a road (i.e., curvature and azimuth). First, the lane marking points were extracted by setting the intensity and scan-angle thresholds and the points were segmented every 0.5 m using their time-stamps. The centroid of each segment was regarded as a point along the lane marking centerline. Next, the lane marking centerline was used to derive its azimuth for checking the horizontal alignment and curvature of the road. Wang et al. [82] estimated road slope and superelevation using point clouds from an Optech Lynx SG1 system. First, the road surface was segmented and partitioned into blocks along the driving direction using navigation data. Then, the points from a road surface partition were used to derive the slope and superelevation.

Holgado-Barco et al. [13] proposed a strategy to obtain road cross-section information (number of lanes, width of the roadway, width of the shoulder and lanes, and superelevation) using a point cloud obtained from an Optech Lynx MMS with a 2D LiDAR. First, the road surface was extracted by removing all the points whose heights were out of the range of the roadway height. The height threshold was computed based on the set of points that corresponded to the horizontal projection of the vehicle using a scan angle threshold specific to the location of the sensors on the MMS. Next, a line fitting was applied to the segmented roadway points from each scan line for deriving the reference line along the cross-section of the road. When the distance of a point within a scan line to the reference line was larger than a pre-defined threshold, the point was regarded as a non-road surface point and was removed. Then, the lane markings were detected from each scan line by identifying the local maximum in the intensity response. To ensure a better separation between consecutive scan lines, the data were collected at a high speed to ensure a separation of approximately 11 cm. To remove false positives, the distance between each two consecutive points detected as lane markings within a scan line was determined to check if it was more than a pre-defined distance threshold. A distance-based clustering technique was applied to the hypothesized lane marking points; and based on the acquisition time of the lane marking points within a scan line, the clusters were categorized into three classes: 1) right lane marking, 2) left lane marking, and 3) ordered (first, second, etc.) center lane markings. Then, a spline was fitted to each class of road marking and the lane width was extracted based on the classified lane marking points and the derived road axis.

For road information extraction from point clouds, most of the previous work [70-78] mainly focused on deriving the road surface and road markings. Some literature [79-82] proposed road characteristics extraction strategies (e.g., slope, curvature, superelevation, and azimuth) but only Holgado-Barco et al. [13] provided a strategy for extracting road cross-section information which includes lane width estimation – one of the key factors in road safety inspection in work zone areas. However, their approach is only suitable for 2D laser scanners and cannot be extended to other types of laser scanners as their strategy is dependent on extracting each scan line separately, which can be achieved for 2D but not for 3D laser scanners. Moreover, their strategy requires availability of the raw measurements of the captured points, such as the timestamp and scan angle. These shortcomings are addressed in the proposed strategy in this thesis because it does not require any information regarding individual scan lines, timestamps, or scan angles. Instead, only the trajectory information and the 3D point cloud with intensity information are required in order to derive accurate lane width estimates.

3. FIRST TIER OF POINT CLOUD PROCESSING

3.1 Introduction

LiDAR-based and image-matching-based point clouds usually have an excessive number of points that do not provide semantic information. Point clouds therefore must undergo a sequence of procedures to generate useful information for a variety of applications, such as mapping and modeling. Point clouds may come from different sources/platforms, and the variance of their internal characteristics (LPD and noise level) can profoundly affect the outcomes of procedures. Therefore, the characteristics of point clouds (Section 3.2) should be the first consideration to ensure the performance of the data processing procedures and the quality of the derived information. Processing a huge number of datasets can be time-consuming and unnecessary; at the same time, an inappropriate down-sampling also would compromise the data processing results. Therefore, an adaptive down-sampling procedure is introduced in Section 3.3 to consider the variation of the internal characteristics and the physical surface characteristics. Point cloud segmentation is one of the initial steps in the data processing chain to partition points into segments and to derive semantic meanings (planar, linear/cylindrical, and rough features). A multi-class simultaneous segmentation of planar, pole-like, and rough features therefore is introduced in Section 3.4. Since segmentation outcomes might exhibit artifacts as mentioned in Chapter 2, a new quality control procedure to improve the segment results is introduced in Section 3.5. Finally, the experimental results of this thesis are presented in Section 3.6.

3.2 Data Characterization and LPD Estimation

In this section, two PCA-based approaches for the characterization of local neighborhoods together with a procedure for LPD estimation are introduced. For laser scanners, the point density depends on the utilized sensor and/or platform as well as the sensor-to-object distance. For image-based point clouds, on the other hand, the point density depends on the texture of the visible surfaces [53] and the extent of the occluded areas. Therefore, a unique LPD must be estimated for every point within the dataset in question. In order to facilitate and identify the nearest n-neighbors for a given point, which is needed for LPD estimation, a kd-tree data structure is applied for organizing the point cloud and increasing the speed of the query process [83].

Precise LPD estimation should be based on the physical surface characteristics of the local neighborhood centered at the query point (i.e., whether the query point is encompassed within a planar, linear/cylindrical, or rough region) [3]. To describe the nature of the local neighborhood for a given point, the PCA is used to decide whether the point belongs to a planar, linear/cylindrical, or rough feature. The dimensionality [84] and threshold-based PCA approaches are used for establishing the point classification. First, the n-nearest neighbors (P_n) of a query point are searched for using the kd-tree data structure. A covariance matrix (*Cov*) is then derived using the 3D coordinates of the points defining (P_n) and their centroid ($\bar{P}$) as given by Equations 3.1 – 3.2. Then, an eigenvalue decomposition of the covariance matrix (Equation 3.3) is used to evaluate the nature of the local neighborhood defined by P_n. The eigenvalues (λ_1, λ_2, and λ_3) of the covariance matrix (*Cov*) are positive and can be ordered such that $\lambda_1 \geq \lambda_2 \geq \lambda_3 > 0$. For the dimensionality-based approach, the feature classification of the neighborhood can be decided according the largest value of the defined dimensionality measures a_{1D}, a_{2D}, and a_{3D} in Equations 3.4 – 3.6 (i.e., linear/cylindrical, planar, or rough neighborhoods will be reflected by situations where a_{1D}, a_{2D}, or a_{3D}, respectively, is the largest one). Alternatively, a threshold-based PCA approach can be used to classify the local neighborhoods. In this regard, normalizing the ordered eigenvalues ($\lambda_{1n} \geq \lambda_{2n} \geq \lambda_{3n} > 0$ – where $\lambda_{in} = \lambda_i/(\lambda_1 + \lambda_2 + \lambda_3)$) is the first priority. A linear/cylindrical local neighborhood has one of the normalized eigenvalues significantly larger than the other two, which can be mathematically described by Equation 3.7 with the help of an additional threshold. A planar local neighborhood, on the other hand, is identified when two of the normalized eigenvalues are significantly larger than the third one. This situation can be mathematically described by Equations 3.8 and 3.9, which involve two thresholds. A rough local neighborhood will be characterized by three normalized eigenvalues that are of similar magnitude (i.e., the local neighborhood neither belongs to a planar feature nor a linear/cylindrical region). The LPD for a planar feature can be established using Equation 3.10. For a linear/cylindrical feature, a LSA model-fitting procedure is used to derive the feature parameters (point along the axis, axis orientation, and radius *R*). When radius *R* is small, the point is identified as belonging to this linear feature and the respective LPD is derived using Equation 3.11. Alternatively, the point is identified as part of a cylindrical feature, and the LPD is derived according to Equation 3.12. For a rough local neighborhood, the LPD is derived using Equation 3.13. Equations 3.10 - 3.13 also provide the corresponding Local Point Spacing (LPS).

$$Cov_{3\times3} = \frac{1}{n+1}\sum_{i=1}^{n+1}(P_i - \bar{P})(P_i - \bar{P})^T \tag{3.1}$$

$$\bar{P} = \frac{1}{n+1}\sum_{i=1}^{n+1} P_i \tag{3.2}$$

$$Cov_{3\times3} = [\overrightarrow{e_1} \quad \overrightarrow{e_2} \quad \overrightarrow{e_3}]\begin{bmatrix}\lambda_1 & 0 & 0\\ 0 & \lambda_2 & 0\\ 0 & 0 & \lambda_3\end{bmatrix}\begin{bmatrix}\overrightarrow{e_1}^T\\ \overrightarrow{e_2}^T\\ \overrightarrow{e_3}^T\end{bmatrix} \tag{3.3}$$

$$a_{1D} = \frac{\sqrt{\lambda_1}-\sqrt{\lambda_2}}{\sqrt{\lambda_1}} \tag{3.4}$$

$$a_{2D} = \frac{\sqrt{\lambda_2} - \sqrt{\lambda_3}}{\sqrt{\lambda_1}} \tag{3.5}$$

$$a_{3D} = \frac{\sqrt{\lambda_3}}{\sqrt{\lambda_1}} \tag{3.6}$$

$$\lambda_{1n} > Threashold_1 \tag{3.7}$$

$$\frac{\lambda_{2n}}{\lambda_{1n}} > Threashold_2 \tag{3.8}$$

$$\lambda_{3n} < Threashold_3 \tag{3.9}$$

$$LPD_{planar}\left(\frac{pts}{m^2}\right) = \frac{n+1}{\pi r_n^2} \quad LPS_{planar}(m) = \frac{1}{\sqrt{LPD_{planar}}} \tag{3.10}$$

$$LPD_{linear}\left(\frac{pts}{m^2}\right) = \frac{n+1}{2r_n} \quad LPS_{linear}(m) = \frac{1}{LPD_{linear}} \tag{3.11}$$

$$LPD_{cylindrical}\left(\frac{pts}{m^2}\right) = \frac{n+1}{4\pi R r_n} \quad LPS_{cylindrical}(m) = \frac{1}{\sqrt{LPD_{cylindrical}}} \tag{3.12}$$

$$LPD_{rough}\left(\frac{pts}{m^3}\right) = \frac{n+1}{\frac{4}{3}\pi r_n^3} \quad LPS_{rough}(m) = \frac{1}{\sqrt[3]{LPD_{rough}}} \tag{3.13}$$

Where,

r_n is the distance between the point in question and its nth-farthest neighbor, and R is the radius of the cylindrical feature.

3.3 Adaptive Down-Sampling on Planar Region

Two of the commonly used point cloud thinning approaches are uniform and point-spacing-based down-sampling. Uniform down-sampling removes points according to the order with which the points are inserted in the input data file and a user-defined number of points. Point-spacing-based down-sampling, on the other hand, is based on a pre-specified minimum distance between neighboring points. However, these methods do not consider the local characteristics of the 3D surface encompassing the points in question. Therefore, it is expected that these down-sampling strategies could compromise further data processing activities such as segmentation and object reconstruction. The proposed adaptive down-sampling strategy considers the physical characteristics of local neighborhoods and selectively removes points within planar neighborhoods whose LPDs are higher than a user-defined LPD. The proposed strategy is based on the fact that for planar neighborhoods, high-density is not crucial for the identification and modeling of such features. On the other hand, linear/cylindrical and rough neighborhoods are partially represented by point clouds (i.e., neither laser scanning nor imaging systems can capture/derive points that completely cover such neighborhoods). Therefore, points within linear/cylindrical and rough neighborhoods are maintained. The hypothesis of the proposed down-sampling strategy, which is denoted here forward as adaptive down-sampling, is that the method will reduce the data processing time while maintaining reliable processing, which is mainly the segmentation of planar, linear/cylindrical, and rough neighborhoods.

When a point belongs to a planar feature, the adaptive down-sampling introduced in Al-Durgham (2014) is applied to achieve a desired point density through the probability-based test in Equation 3.14.

$$\delta = {}^{LPD_d}/_{LPD_i} \begin{cases} > t, \ maintain \\ else, \ \ ignore \end{cases} \tag{3.14}$$

Where,

LPD_d is the desired local point density in pts/m^2,

LPD_i is the local point density at the i$^{\text{th}}$ point local neighborhood in pts/m^2, and

t is a random number that is picked from a uniform distribution in the range [0, 1].

According to Equation 3.14, when the local point density (LPD_i) of a point is below the desired point density (LPD_d), the point will be maintained in the down-sampled dataset. This is due to the fact that when LPD_i is smaller than LPD_d, the test value (δ) always will be larger than

t since the random number (t) is generated from a uniform distribution in the range [0, 1]. Alternatively, when the LPD is larger than the desired point density, as shown in Figure 3.1, the point will be removed with the probability of (1 - δ). In other words, the point will be maintained with the probability of δ (e.g., if LPD_d is 30 pts/m^2 and a given location LPD_i is 100 pts/m$^{2)}$. Then, δ is 0.3 and the probability of the generated random number (t) falling between 0 and 0.3 is 30%, meaning that 30% of points will be maintained in the area having 100 pts/m^2 point density.

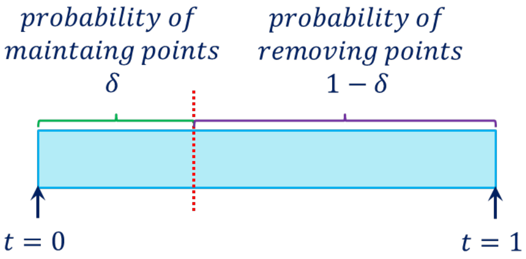


Figure 3.1. An illustration of adaptive down-sampling

An illustration of the adaptive down-sampling strategy on a simulated circular point cloud with 3.0m radius is shown in Figure 3.2. The distribution of the point density that corresponds to the original simulated point clouds (Figure 3.2 (a)) is shown in Figure 3.2 (c), which is derived from Equation 3.10. The distribution of the point density that corresponds to the point cloud after applying the adaptive down-sampling (Figure 3.2 (b)) is shown in Figure 3.2 (d). For the original point cloud, the maximum point density is about 2000 pts/m^2. In this example, the desired point density is set as 400 pts/m^2; therefore, within the maximum point density area, the probability of selecting random values (t) less than or equal to 1/5 is 1/5. This means that 4/5 of the points in the central high-density neighborhood will be removed assuming that there are enough samples drawn from such neighborhood. Therefore, after applying the adaptive down-sampling, the high point density area will be reduced to the desired point density to achieve a uniform LPD distribution, which can be seen in Figure 3.2 (d).

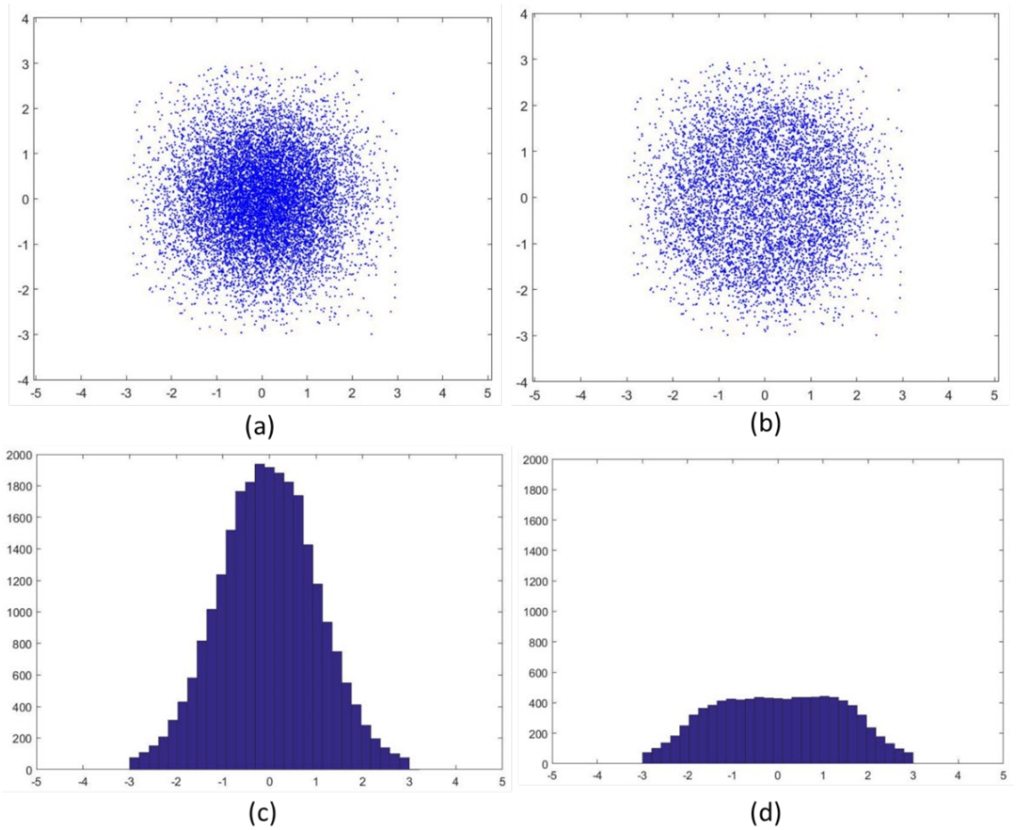


Figure 3.2. (a) Simulated point cloud, (b) derived point cloud after applying adaptive down-sampling, (c) and (d) the point density distribution corresponding to (a) and (b)

3.4 Multi-Class Simultaneous Segmentation Procedure

Before starting segmentation, data characterization and LPD estimation (Section 3.2) is applied first. Then, the multi-class simultaneous segmentation proceeds according to the following steps: 1) Section 3.4.1: distance-based region growing starting from randomly-selected seed points to define seed regions with pre-defined size; 2) Section 3.4.2: PCA-based classification and feature modeling of generated seed regions; 3) Section 3.4.3): sequential region-growing according to the quality of fit between the neighboring points and the fitted-model through the constituents of the seed regions; and 4) Section 3.4.4: PCA-based classification, model-fitting, and region growing of non-segmented points and distance-based region growing for the segmentation of rough points.

3.4.1 Distance-Based Region Growing for the Derivation of Seed Regions

This step starts by forming a set of seed points that are randomly distributed within the point cloud in question. Rather than directly defining seed regions, which are centered at the randomly-established seed points, the seed regions are defined through distance-based region growing. More specifically, starting from a user-defined percentage of randomly-selected seed points, distance-based region growing is applied, for which the only criterion is the spatial closeness of the points to the seed point in question as determined by the LPS. The distance-based region growing process continues until a pre-specified region size is attained. This approach for seed-region definition ensures that the seed region is large enough, while avoiding the risk of having the seed region comprised of points from two or more different classes. Therefore, when dealing with different features that are spatially close to each other, it is important to confirm that the seed regions belong to the individual objects as long as the spatial separation between those features is larger than the LPS. Having larger seed regions that belong to individual objects will lead to better identification of the respective models associated with those neighborhoods, which in turn will increase the reliability of the segmentation procedure.

3.4.2 PCA-Based Classification and Modeling of Seed Regions

After the seed regions are defined, PCA is used to identify whether they belong to planar, pole-like, or rough neighborhoods. More specifically, the relationships among the normalized Eigen values of the dispersion matrix of the points within a seed region relative to its centroid are used to identify the planar seed regions (i.e., where two of the normalized Eigen values are significantly larger than the third one), pole-like seed regions (i.e., where one of the normalized Eigen values is significantly larger the other two), and rough seed regions (i.e., where the three normalized Eigen values are of similar magnitude). For planar and pole-like seed regions, a LSA model-fitting procedure is used to derive the plane/pole-like parameters together with the quality of fit between the points within the seed region and the defined model as represented by the respective a-posteriori variance factor, which will be used as an indication of the local noise level within the seed region. For a planar seed region, the LSA estimates the three plane parameters – a, b, and c – using either Equations 3.15, 3.16, or 3.17. The choice of the appropriate plane equation depends on the orientation of the Eigen vector corresponding to the smallest Eigen value (i.e., the one defining the normal to the plane) – as shown in Figure 3.3. For a pole-like feature, the LSA estimates its radius together with four parameters that define the coordinates of a point along the axis and the axis orientation – p, q, a, and b – using either Equations 3.18, 3.19, or 3.20. The choice of the appropriate equation depends on the orientation of the Eigen vector corresponding to the largest Eigen value (i.e., the one defining the axis orientation of the pole-like feature) as shown in Figure 3.4. One should note that the variable ***t*** in Equations 3.18 – 3.20 depends on the distance between the projection of any point onto the axis of the pole-like feature and the utilized point along the axis: (p,q,0) for the axis defined by Equation 3.18, (p,0,q) for the axis defined by Equation 3.19, or (0,p,q) for the axis defined by Equation 3.20 (refer to Figure 3.4).

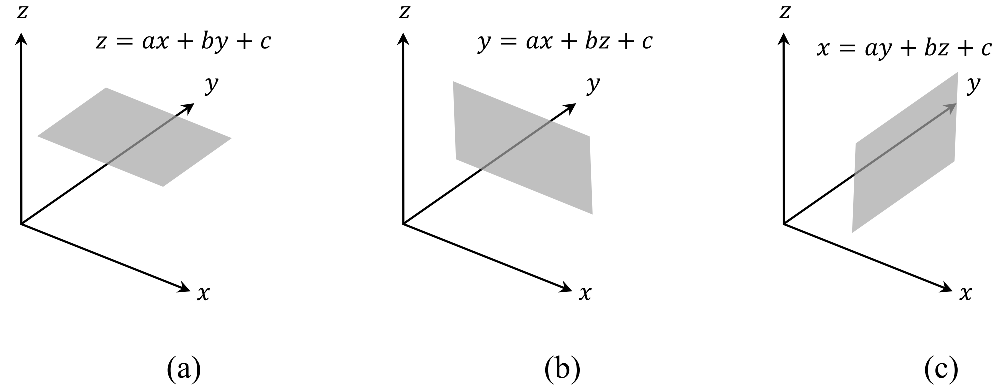


Figure 3.3. Representation scheme for 3D planar features; planes that are almost parallel to the $xy - plane$ (a), planes that are almost parallel to the $xz - plane$ (b), and planes that are almost parallel to the $yz - plane$ (c)

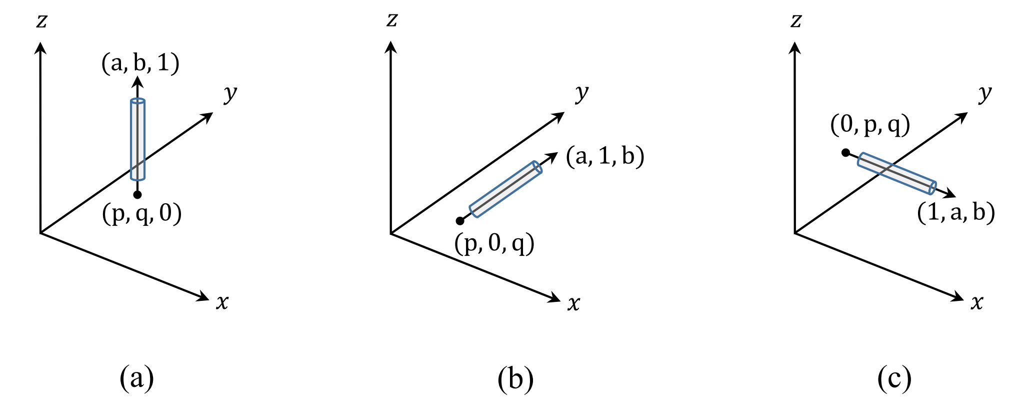


Figure 3.4. Representation scheme for 3D pole-like features; pole-like features that are almost parallel to the $z - axis$ (a), pole-like features that are almost parallel to the $y - axis$ (b), and pole-like features that are almost parallel to the $x - axis$ (c)

$$z = ax + by + c \tag{3.15}$$

$$y = ax + bz + c \tag{3.16}$$

$$x = ay + bz + c \tag{3.17}$$

$$\begin{aligned} x &= p + t\,a \\ y &= q + t\,b \\ z &= t \end{aligned} \tag{3.18}$$

$$\begin{aligned} x &= p + t\,a \\ y &= t \\ z &= q + t\,b \end{aligned} \tag{3.19}$$

$$
\begin{aligned}
x &= t \\
y &= p + t\,a \\
z &= q + t\,b
\end{aligned}
\tag{3.20}
$$

3.4.3 Region-growing Starting from Optimally-selected Seed Regions

In this thesis, the seed regions representing the planar and pole-like features are sorted according to an ascending order for the evaluated a-posteriori variance factor in the previous step. One should note that this a-posteriori variance factor is an indication of the normal distances between the points within the seed region and the best-fitted model (i.e., an indication of the compatibility of the physical surface and the underlying mathematical model as well as the noise level in the dataset). Starting with the seed region that has the minimum a-posteriori variance factor, a region-growing process begins while considering the spatial proximity as defined by the LPS and the normal distance to the defined model through the seed region as the similarity criteria. Throughout the region-growing process, the model parameters and the a-posteriori variance factor are sequentially updated. For a given seed region, the growing process proceeds until no further points can be added. The sequential region growing according to the established quality of fit (the a-posteriori variance factor) ensures that the seed regions showing better fit to the planar or pole-like feature model are considered first. Thus, rather than starting the region growing from randomly established seed points, the growing starts from the locations that exhibit good fit with the pre-defined models for planar and pole-like features.

3.4.4 Sequential Segmentation of Non-segmented Points and Rough Regions

Depending on the user-defined percentage of seed points, one should expect that some points might not be segmented or considered since they are not within the immediate vicinity of seed points that belong to the same class or they happen to be in the neighborhood of rough seed regions. To consider such situations, a sequential region-growing process is conducted by going through the points within the kd-tree data structure, starting from its root and identifying the points that have not been segmented yet. When a non-segmented point within the kd-tree data structure is encountered, the following region-growing procedure is implemented:

1. Starting from a non-segmented point, distance-based region growing proceeds according the established LPS until a pre-defined seed-region size is achieved.
2. For the established seed region, PCA is used to decide whether the seed region represents a planar, pole-like, or rough neighborhood. If the seed region is deemed as being part of a planar or pole-like feature, the parameters of the respective model are estimated through a LSA procedure.
3. A region-growing process is carried out using the LPS and quality of fit with the established model in the previous step as the similarity measures. Throughout the region-growing process, the model parameters and the respective a-posteriori variance factor are sequentially updated.
4. Steps 1 through 3 are repeated until all the non-segmented nodes within the kd-tree data structure are considered.

The last step of the segmentation process, is grouping neighboring points that belong to rough regions, which is carried out in the following steps:

1. For the seed regions, which have been classified as being part of rough neighborhoods during the first or the second stages of the segmentation procedure, distance-based region-growing is conducted of the non-segmented points.

2. Then, the kd-tree is inspected, starting from its root node, to identify the non-segmented nodes.

3. Finally, the non-segmented nodes are utilized as seed points for distance-based segmentation of the rough regions.

At this stage, the constituents of a point cloud have been classified and segmented into planar, pole-like, and rough segments. For the planar and pole-like features, the respective model parameters and the a-posteriori variance factor, which describes the average normal distance between the constituents of a region and the best-fit model, also have been established.

3.5 QC of Segmentation Outcomes

In spite of the facts that 1) the proposed region-growing segmentation strategy has been designed to optimally-select seed regions that exhibit the best quality of fit to the LSA-based planar/pole-like models and 2) the region growing is based on the established LPS for the individual points, one cannot guarantee that the segmentation outcome will be perfect (i.e., the segmentation outcome might still exhibit artifacts). For example, the segmented regions from an earlier stage might invade the segmented regions at a later stage. Also, due to the location of the randomly-established seed points and the nature of the objects within the point cloud, there might be instances where the seed regions are wrongly classified. For example, a portion of a planar feature is wrongly classified as a pole-like feature (Figure 3.5-a) or a set of contiguous pole-like features are identified as a planar segment (Figure 3.5-b).

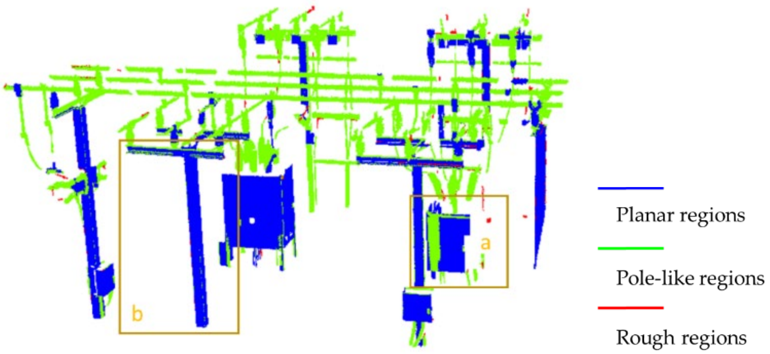


Figure 3.5. Classified point cloud: planar, pole-like, and rough regions are shown in blue, green, and red, respectively

As mentioned in Section 2.4, prior research efforts dealt with the detection and mitigation of over-segmentation and under-segmentation problems, but the potential artifacts that might arise when simultaneously segmenting planar, pole-like, and rough regions were not considered. The proposed QC framework proceeds according to the following three stages: 1) developing a list of hypothesized artifacts/problems that might take place during the segmentation process; 2) developing procedures for the detection of instances of such artifacts/problems without the need for having reference data; and 3) developing approaches to mitigate such problems whenever detected. The hypothesized problems that might take place within a multi-class simultaneous segmentation of planar and pole-like features are described below. Figure 3.6 (a–g) is a schematic illustration of these hypothesized problems; in sub-figures $a,\ b,\ c,\ e,\ f, and\ g$ classified planar regions are displayed in light blue while classified pole-like features are displayed in light green.

1. <u>Misclassified planar features:</u> Depending on the LPD/LPS and pre-set size for the seed regions, a pole-like feature might be wrongly classified as a planar region. This situation might be manifested in one of the following scenarios:

   a. Single pole-like feature wrongly classified as a planar region (Figure 3.6–a)

b. Multiple contiguous pole-like features classified as a single planar region (Figure 3.6–b)

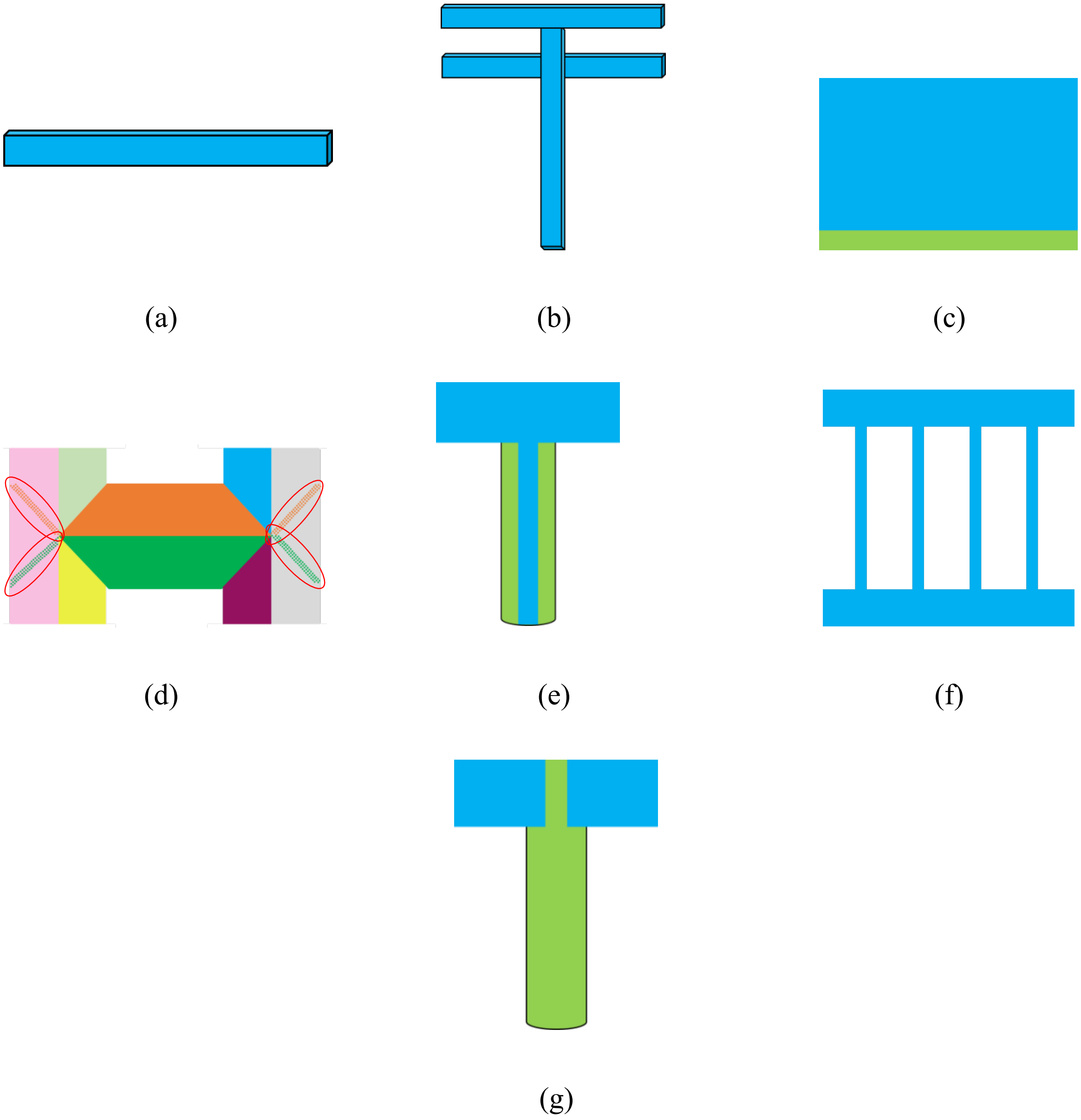
2. Misclassified linear features: Depending on the location of the randomly-established seed points, a portion of a planar region might be classified as a single pole-like feature (Figure 3.6–c).

3. Partially misclassified planar and pole-like features: Depending on the order of the region growing process, the segmented planar/pole-like features at the earlier stage of the segmentation process might invade neighboring planar/pole-like features. This situation might be manifested in one of the following scenarios:

   a. Earlier-segmented planar regions invade neighboring planar features (Figure 3.6–d),

   b. Earlier-segmented planar regions fully/partially invade neighboring pole-like features (Figure 3.6–e&f), where planar region partially invades a neighboring pole-like feature), and

   c. Earlier-segmented pole-like features invade neighboring planar features (Figure 3.6–g).



(a) (b) (c)

(d) (e) (f)

(g)

Figure 3.6. Possible segmentation artifacts: misclassified planar features (a) and (b); misclassified pole-like feature (c); partially misclassified planar features (d), (e), and (f); and partially misclassified pole-like feature (g). Planar and pole-like features are displayed in light blue and light green, respectively, in subfigures $a, b, c, e, f, and\ g$

The above problems can be categorized as follows: 1) interclass competition for neighboring points; 2) intraclass competition for neighboring points; and 3) fully/partially misclassified planar and pole-like features. The following procedure is proposed to detect and mitigate these segmentation problems.

1. Initial mitigation of interclass competition for neighboring points: A key problem in region-growing segmentation is that the derived regions at an early stage might invade the neighboring features of the same or different class, which are derived at a later stage. In this QC category, this procedure considers potential invasion among features that belong to different classes. Specifically, for segmented features in a given class (planar or pole-like features), the features in the other classes (including rough regions) are considered as potential candidates that could be incorporated into the constituent regions of the former class. For example, the constituents of the pole-like features and rough regions are considered as potential candidates that could be incorporated into the planar features. In this case, if a planar feature has potential candidates that are spatially close as indicated by the established LPS and the normal distance between those potential candidates and the LSA-based model through that planar feature is within the respective a-posteriori variance factor, those potential candidates are incorporated into the planar feature in question. The same procedure is applied for pole-like features while considering planar and rough regions as potential candidates. In this regard, the respective QC measure – $QC_{interclass\ competition}$ – is evaluated according to Equation 3.21, where $n_{incorporated}$ represents the number of incorporated points from other classes and $n_{potential\ candidates}$ represents the number of potential candidates for this class. A lower percentage indicates lower instances of points that have been incorporated from other classes.

$$QC_{interclass\ competition} = {}^{n_{incorporated}}\!/_{n_{potential\ candidates}} \tag{3.21}$$

2. Mitigation of intraclass competition for neighboring points: This problem takes place whenever a feature, which has been derived at the earlier stage of the region growing, invades other features from the same class segmented at a later stage. One can argue that intraclass competition for pole-like features is quite limited, mainly due to the narrow spread of pole-like features across its axis). Therefore, for this QC measure, only intraclass completion for planar features is considered as can be seen in Figure 3.6–d, where the middle planar regions invade the left and right planar features (invading portions are highlighted by red ellipses). Detection and mitigation of this problem starts by deriving the inner and outer boundaries of the segmented planar regions (Figure 3.7 illustrates an example of inner and outer boundaries for a given segment). The inner and outer boundaries can be derived using the minimum convex hull and inter-point-maximum-angle procedures presented by Sampath and Shan [86] as well as Lari and Habib [31], respectively. Then, for each of the planar regions, a review is conducted to determine whether some of their constituents are located within the boundaries of neighboring regions and that the normal distances between the constituents and the fitted model through the neighboring regions are within their respective a-posteriori variance factor. In such a case, the individual points that satisfy these conditions are transformed from the invading planar feature to the invaded one. For this QC category, the respective measure is determined according to Equation 3.22, where $n_{invading}$ represents the number of invading planar points that have been transformed from the invading to the invaded segments and $n_{plane\ total}$ represents the total number of originally-segmented planar points. In this case, a lower percentage indicates lower instances of such problem.

$$QC_{intraclass\ competition} = {}^{n_{invading}}/_{n_{plane\ total}} \tag{3.22}$$

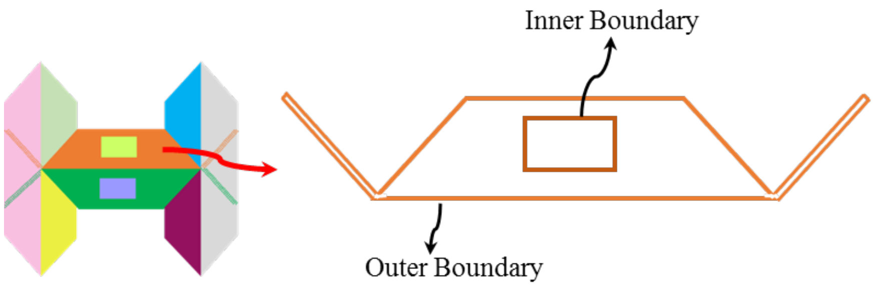


Figure 3.7. Inner and outer boundary derivation for the identification of intraclass competition for neighboring points

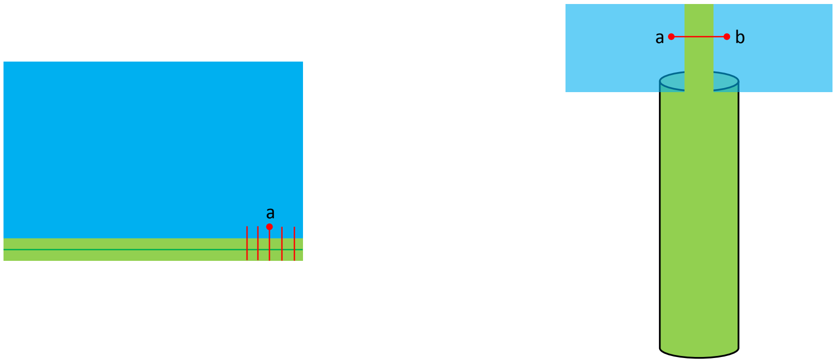
3. Single pole-like feature wrongly classified as a planar feature: To detect such instances (Figure 3.6-a illustrates the situation), PCA is conducted on the constituents of the individual planar features, and the resulting PCA-based normalized Eigen values will indicate the 1-D spread of such regions. Whenever this scenario is encountered, the LSA-based parameters of the fitted cylinder through this feature together with the respective a-posteriori variance factor are derived. The planar feature will be reclassified as a pole-like one if the latter's a-posteriori variance factor is almost equivalent to the planar-based one. For this case, the respective QC measure, $QC_{reclassified\ linear\ feaure}$ is represented by Equation 3.23, where $n_{reclassified\ lines}$ is the number of points within the reclassified linear features and $n_{plane\ total}$ is the total number of points within the originally-segmented planar features. In this case, a lower percentage indicates fewer instances of such a problem.

$$QC_{reclassified\ linear\ features} = {}^{n_{reclassified\ lines}}/_{n_{plane\ total}} \tag{3.23}$$

4. Mitigation of fully or partially misclassified pole-like features: For this problem, which is illustrated in Figure 3.6–c and Figure 3.6–g, the pole-like features or portions of pole-like features that are encompassed within neighboring planar features are identified. The process starts with

identifying neighboring pole-like and planar features where the axis of the pole-like feature is perpendicular to the planar-feature normal. Then, the constituents of the pole-like feature are projected onto the plane defined by the planar feature. Instances where the pole-like feature is encompassed – either fully or partially – within the planar feature, are identified by slicing the pole-like feature in the across direction to its axis. For each of the slices, the closest planar point(s) that does (do) not belong to the pole-like feature in question (e.g., point $\boldsymbol{a}$ in Figure 3.8–a or points $\boldsymbol{a}$ & $\boldsymbol{b}$ in Figure 3.8–b) are determined. If the closest point(s) happen to be immediate neighbor(s) of the constituents of that slice (as defined by the established LPS), then one can suspect that the portion of the pole-like feature in the vicinity of that slice might be encompassed within the neighboring planar region and that portion of the pole-like feature might be invading the planar region). To confirm or reject this suspicion, the normal distances between the constituents of the slice and the neighboring planar region are evaluated. If these normal distances are within the respective a-posteriori variance factor for the planar region, the slice is confirmed to be encompassed within the planar region. Whenever the pole-like feature is fully encompassed within the planar region (Figure 3.8–a), all the slices will have immediate neighbors from that planar region while having minimal normal distances. Consequently, the entire pole-like feature will be reassigned to the planar region. On the other hand, whenever the linear feature is partially encompassed within the planar region, the slices are identified where the closest neighbors to such slices are not immediate neighbors (Figure 3.8–b). The portion of the pole-like feature, which is defined by such slices, will be retained while the other portion will be reassigned to the planar region. The QC measure in this case is defined by Equation 3.24, where $n_{encompassed\ line\ points}$ represents the number of points within the pole-like features that are encompassed within the planar feature and $n_{line\ total}$ is the total number of points within the originally-segmented linear features. A lower percentage indicates fewer instances of such problem.

$$QC_{partially/fully\ misclassified\ pole-like\ features} = {}^{n_{encompassed\ line\ points}}\!/_{n_{line\ total}} \tag{3.24}$$



(a)

(b)

Figure 3.8. Slicing and immediate-neighbors concept for the identification of fully/partially misclassified pole-like features (a)/(b)

5. <u>Mitigation of fully or partially misclassified planar features:</u> The conceptual basis of the implemented procedure to detect instances of such problem (as illustrated by Figure 3.6–b, Figure 3.6–e, and Figure 3.6–f) is that whenever planar features are either fully (Figure 3.6–b and Figure 3.6–f) or partially (Figure 3.6–e) misclassified, a significant portion of the encompassing MBR will not be occupied by those features (refer to Figure 3.9–a). In this regard, one should note that the MBR denotes the smallest area rectangle that encompasses the identified boundary of the planar region in question. Therefore, to detect instances of such problem, the MBR for the individual planar regions first is defined. Then, the ration between the area of the planar region in question and the area of the encompassing MBR is evaluated. Whenever this area is below a pre-defined threshold, the planar feature in question is suspected of containing pole-like features,

which can take the form of tentacles to the original planar region (as can be seen in Figure 3.9–a). To identify such features, a 2D-linear feature segmentation procedure (Figure 3.9-b) is performed, which is similar to the one proposed earlier with the exception that it is conducted in 2D rather than 3D (i.e., the line parameters would include slope, intercept, and width. More specifically, a pre-defined percentage of seed points are established. Then, a distance-based region growing is carried out to define the seed regions with a pre-set size. A 2D-PCA and line fitting procedure is conducted to identify seed regions that represent 2D lines. Those seed regions are then incorporated within a region-growing process that considers both the spatial closeness of the points and their normal distance to the fitted 2D lines. Following the 2D-line segmentation, an over-segmentation QC is carried out to identify single linear features that have been identified as multiple ones. Moreover, as shown in Figure 3.9-c), the QC conducted in the previous step is implemented to identify partially misclassified linear features (i.e., the invading portion of the linear feature(s). The QC measure for this problem is evaluated according to Equation 3.25, where $n_{misclassified\ plane\ points}$ represents the number of points within the planar feature that belong to 2D lines and $n_{plane\ total}$ is the total number of originally-segmented planar points. A lower percentage indicates fewer instances of this problem.

$$QC_{partially/fully\ misclassified\ planar\ features} = {}^{n_{misclassified\ plane\ points}}/_{n_{plane\ total}} \quad (3.25)$$

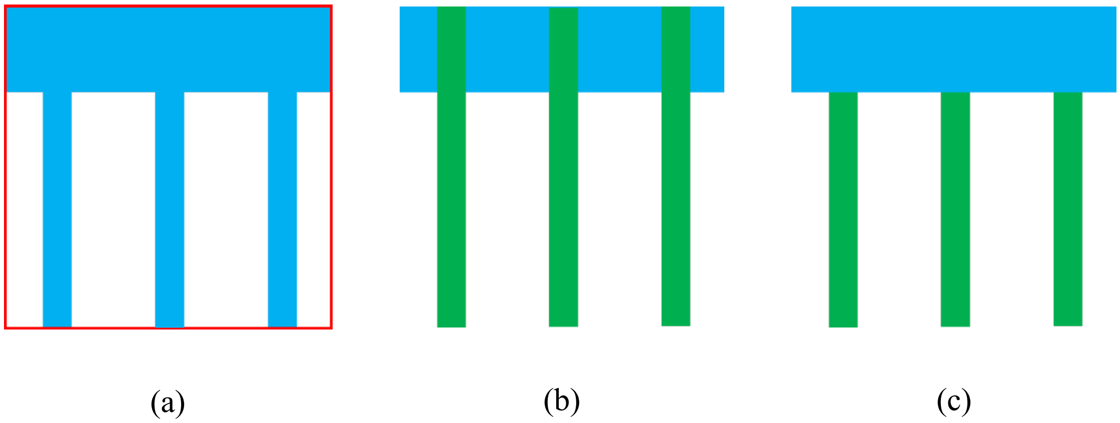


Figure 3.9. Segmented planar feature (in light blue) and the encompassing MBR (in red (a), segmented linear features (in green) (b), and final segmentation after the identification of partially-misclassified linear features (c)

3.6 Experimental Results

This chapter introduced the framework for initial point cloud processing, which can be summarized in two steps. First, an adaptive down-sampling method is conducted to efficiently remove the redundant points within planar neighborhoods. Second, a multi-class simultaneous segmentation is conducted to segment the planar, pole-like, and features. In addition, a succession of quality-control measures were introduced to evaluate and improve the segmentation results. The experiments conducted to confirm the results follow.

3.6.1 Data Description

<u>Mobile Terrestrial Laser Scanner Dataset – MTLS:</u> This dataset was captured by an Optech Lynx mobile mapping system with a 250K Hz pulse repetition rate. The noise level within the spatial extent of a given number of points is considered low. As it can be seen in Figure 3.10 (a), the covered area includes roads, grass-covered spaces, bushes, poles, cables, and traffic signs.

Airborne Laser Scanner Dataset – ALS: This dataset was captured by an Optech ALTM 3100 over an urban area that includes planar roofs, roads, and trees/bushes. The extent of the covered area is roughly 0.5km x 0.5km. Figure 3.10 (b) shows a perspective view of the ALS point cloud, where the color is based on the height of the different points.

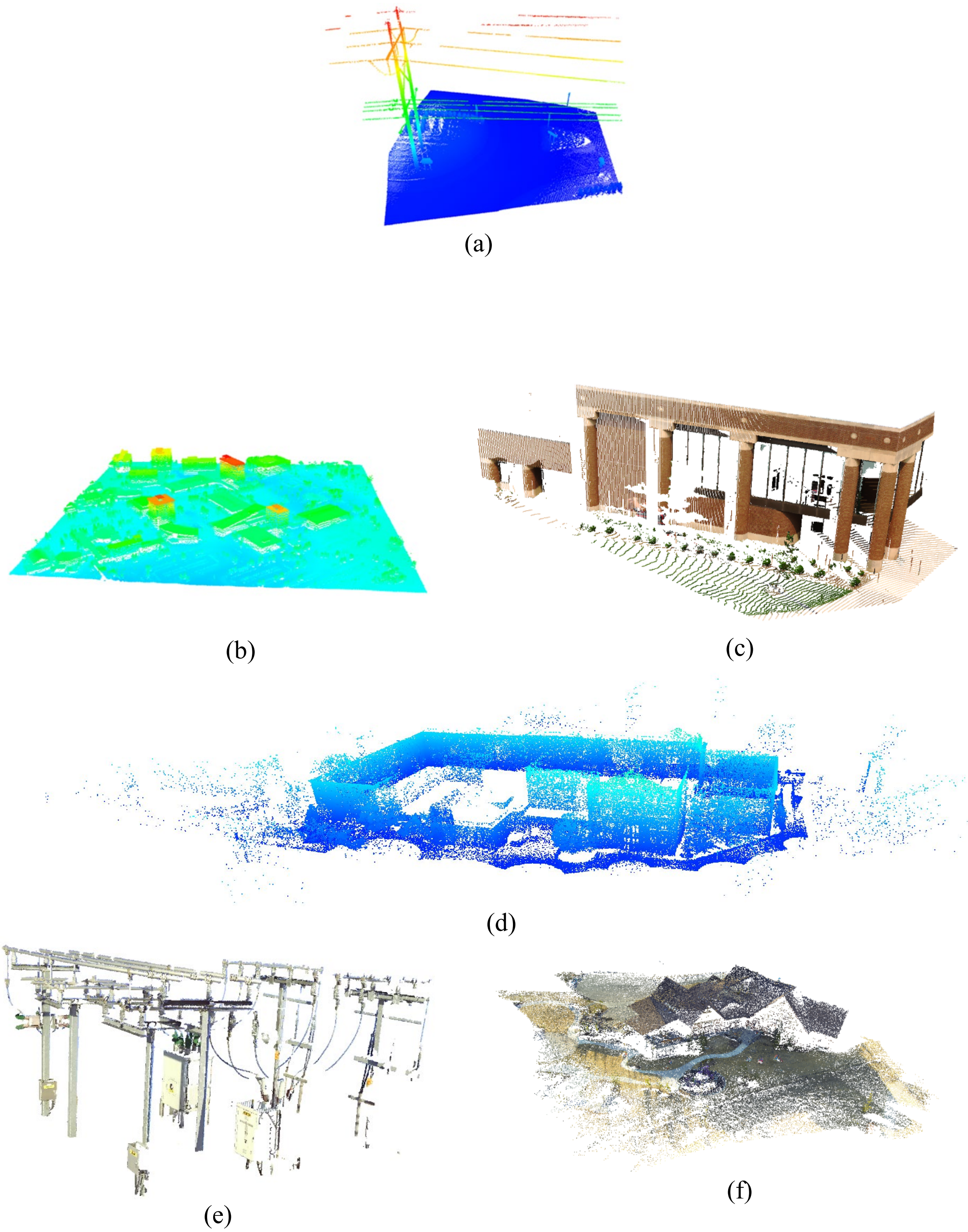
First Stationary Terrestrial Laser Scanner Dataset – STLS1: This dataset was captured by a FARO Focus3D X330 scanner. The effective scan distance for this scanner ranges from 0.6m up to 330m. The ranging error is ±2mm. The scanner was positioned in the vicinity of a building façade with planar and cylindrical features whose radii is almost 0.6 m. The extent of the covered area is approximately 35m x 20m x 10m. Figure 3.10 (c) illustrates the perspective view of this dataset with the colors derived from the scanner-mounted camera.

Second Stationary Terrestrial Laser Scanner Dataset – STLS2: This dataset was captured by a Leica HDS 3000 scanner. The effective scan distance for this unit ranges up to 300m with ±6mm position accuracy at 50m. The covered area includes a planar building façade, some light poles, and trees/bushes. The extent of the covered area is almost 250m x 200m x 26m. A perspective view of this dataset is illustrated in Figure 3.10 (d).

Third Stationary Terrestrial Laser Scanner Dataset – STLS3: This dataset covers an electrical substation and was captured by a FARO Focus3D X130 scanner. The effective scan distance ranges from 0.6m up to 130m. The ranging error is ±2mm. The dataset is mainly comprised of pole-like features with relatively small radii. The extent of the covered area is roughly 12m x 10m x 6m. A perspective view of this dataset is provided in Figure 3.10 (e) with the colors derived from the scanner-mounted camera.

Dense Image Matching Dataset – DIM: This dataset, which is shown in Figure 3.10 (f), was derived from a block of 28 images captured by a GoPro 3 camera onboard a DJI Phantom 2

UAV platform over a building with a complex roof structure. The extent of the covered area is approximately 100m x 130m x 17m. A Structure from Motion (SfM) approach developed by He and Habib [87] is adopted for automated determination of the frame camera EOPs as well as a sparse point cloud representing the imaged area relative to an arbitrarily-defined local reference frame. Then, a semi-global dense matching is used to derive a dense point cloud from the involved images [22]. The noise level within the spatial extent of a given number of points is considered high. This dataset has extremely dense point clouds).



(a)

(b)

(c)

(d)

(e)

(f)

Figure 3.10. Point clouds from the (a) MTLS, (b) ALS, (c) STLS1, (d) STLS2, (e) STLS3, and (f) DIM datasets

3.6.2 Experimental Results of Down-sampling Procedure

The proposed adaptive down-sampling procedure can reduce only the point density within planar neighborhoods whose LPD is larger than the desired one. Thus, it is expected that the linear/cylindrical and rough regions would be emphasized as no down-sampling would take place at local neighborhoods that belong to such regions. Then, the proposed multi-class segmentation technique was applied to evaluate the outcome from the down-sampled and original datasets according to their completeness and execution times. To evaluate the performance of the proposed adaptive down-sampling strategy against other existing approaches (uniform and point-spacing-based down-sampling), this section presents the experimental results from the MTLS, STLS2, and DIM point clouds. The point density in the ALS dataset was lower than that of the MTLS, STLS2, and DIM datasets and, there was very little variation in the LPD. Therefore, in this experiment, we did not consider the ALS dataset. The main objectives of the conducted experiments were as follows:

1. Investigate the impact of different down-sampling techniques on the LPD for the derived point clouds from MTLS, STLS2, and DIM;
2. Investigate the impact of the noise level within the point cloud on the down-sampling process;
3. Investigate the impact of different point classification techniques (i.e., dimensionality and threshold-based PCA classification of local neighborhoods) on the down-sampling process;
4. Investigate the impact of different down-sampling techniques on the execution time of subsequent data processing activities (mainly segmentation of the planar, linear/cylindrical, and rough regions); and

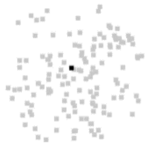
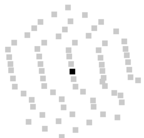
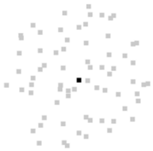
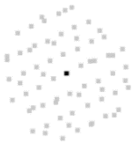
5. Compare the segmentation outcomes for the down-sampled datasets through the proposed and existing approaches in terms of their ability to maintain reliable identification of the planar, linear/cylindrical, and rough regions.

The following strategy was used to establish a common basis for the comparative analysis of performance of the different down-sampling techniques; 1) a desired point density was set to down-sample a given point cloud through the proposed adaptive down-sampling procedure; 2) the LPS corresponding to the desired LPD was used to thin the point cloud through the distance-based down-sampling approach; and 3) the resulting number of points from the adaptive down-sampling was used by the uniform down-sampling approach to produce the same number of points. Using this strategy, we conducted experiments with three different point clouds derived from MTLS, STLS2, and DIM. For these datasets, we investigated the impact of different strategies for PCA-based point classification and dimensionality and threshold-based PCA approaches. We also investigated the outcome of the multi-class segmentation applied to the original and down-sampled point clouds using the proposed and existing approaches. The following subsections describe the different datasets and segmentation results from the original point cloud as well as the adaptive, uniform, and point-spacing-based down-sampled datasets.

3.6.2.1 Data Characteristics

As mentioned earlier, one of the objectives of the conducted experiments was to evaluate the performance of the dimensionality and threshold-based PCA approaches for point classification. The noise level within the established extent by the used number of points to define the local neighborhood, which will be denoted here forward as the "*noise-level-to-area ratio,*" had a major impact on the classification approaches. Table 3.1 illustrates the *noise-level-to-area ratio* for different portions that belonged to the planar local neighborhoods in the MTLS, STLS2, and DIM datasets. More specifically, for these neighborhoods, the ratio was the RMSE of the normal distances between the sample representative points and the best-fitted plane for these neighborhoods ($RMSE_{nd}$) and the circular area defined by the distance between the query point and its n^{th}furthest neighbor. Table 3.1 clearly shows that the *noise-level-to-area ratio* within the DIM dataset was significantly higher than that for the other datasets. Therefore, it was expected that the classification of such datasets would be more challenging.

Table 3.1. *Noise-level-to-area ratio* for representative sample points in the MTLS, STLS2, and DIM datasets

	MTLS	STLS2 (Dense)	STLS2 (Sparse)	DIM
Illustration				
Number of Neighborhoods (pts)	70	70	70	140
$RMSE_{nd}\ (m)$	0.001	0.011	0.017	0.048
$r_n\ (m)$	0.075	0.273	0.551	0.173
$\frac{RMSE_{nd}}{\pi r_n^2}\ (m/m^2)$	0.076	0.050	0.018	0.513
r_n : the distance between the point in question and its nth-farthest neighbor $RMSE_{nd}$: Root Mean Square Error of the normal distance between the points and the fitted plane				

3.6.2.2 Results & Analysis

In this section, we compare the performance of three down-sampling strategies: 1) adaptive, 2) uniform, and 3) point-spacing-based. The adaptive down-sampling strategy was only applied on planar features according to Equation 3.14 after specifying the desired LPD. Uniform down-sampling was applied using "CloudCompare," given a pre-specified number of points, which was defined by the size of the adaptively down-sampled dataset. For the point-spacing-based down-sampling, points were removed according to the minimum spacing, which corresponded to the desired point density for the adaptive down-sampling. The desired point density and minimum distance for the adaptive and point-spacing-based down-sampling, respectively, for all the datasets are shown in Table 3.2. The dimensionality-based PCA classification approach was applied to all the datasets to identify the nature of the defined local neighborhoods (i.e., determine whether they represent planar, linear/cylindrical, or rough regions). The respective LPD for the classified local neighborhoods were then estimated according to Equations 3.10 through 3.13. For the MTLS and STLS2 datasets, 70 points were used to define the local neighborhoods. For the DIM dataset, on the other hand, 140 points were used for defining the local neighborhood of the individual points More points were used for the DIM dataset due to its extremely high-density and relatively high noise level. The derived point density statistics from the dimensionality-based PCA classification approach are shown in Table 3.3 through Table 3.5. To compare the results from the two PCA classification alternatives, the LPD for the DIM dataset using the threshold-based approach with relaxed values for the thresholds in Equation 3.7 through Equation 3.9 were evaluated, which were set as follows: threshold1 = 0.7, threshold2 = 0.6, and threshold3 = 0.25. The statistics of the estimated LPD for the DIM dataset using the threshold-based PCA classification approach are shown in Table 3.6. In Table 3.3 through Table 3.6, the number of planar, linear/cylindrical, and rough points denotes the number of points that are deemed to belong to the planar, linear/cylindrical, and rough local neighborhoods, respectively, using the dimensionality or threshold-based PCA classification approach. Figure 3.11 illustrates the point density maps for the original as well as the adaptive, uniform, and point-spacing-based down-sampled DIM datasets (one should note that the adaptive down-sampled dataset was generated while using the threshold-based PCA classification). From the density maps in Figure 3.11, one can see that the adaptive down-sampling thinned the points in the planar areas with high-density while it retained the points in the planar areas with sparse points as well as the linear/cylindrical and rough regions. This characteristic led to emphasized linear/cylindrical and rough regions, which are clearly visible in Figure 3.11 (b) where the gable roof edges and ridges as well as the trees/bushes maintained a high LPD. The emphasis of the linear/cylindrical features by the adaptive down-sampling procedure is also visible in terms of an increase in the number of points that belong to such features, especially for the DIM dataset that had a large noise-level-to-area ratio.

Table 3.2. Down-sampling parameters for the adaptive and point-spacing-based approaches

	Adaptive down-sampling Desired point density (pts/m^2)	Point-spacing-based down-sampling Minimum spacing between points (m)
MTLS	100	0.1
STLS2	50	0.141
DIM	50	0.141

Table 3.3. LPD statistics for the original and down-sampled MTLS datasets (Note that the dimensionality-based PCA classification is used for adaptive down-sampling)

	Original	Adaptive down-sampling	Uniform down-sampling	Point-spacing-based down-sampling
Number of Points	1,105,962	139,738	139,738	62,033
Max. Planar Point Density (pts/m^2)	30,605	10,652	2,033	142
Min. Planar Point Density (pts/m^2)	1.210	1.210	0.494	0.751
Mean Planar Point Density (pts/m^2)	2,669	106	336	58.484
Number of Planar Points	1,050,790	84,343	133,707	52,355
Max. Linear Point Density (pts/m)	1,119	1,020	N/A	N/A
Min. Linear Point Density (pts/m)	502	16.753	N/A	N/A
Mean Linear Point Density (pts/m)	976	823	N/A	N/A
Number of Linear Points	131	228	N/A	N/A
Max. Cylindrical Density (pts/m^2)	34,681	26,561	2,416	160
Min. Cylindrical Point Density (pts/m^2)	5.448	5.448	3.464	3.521
Mean Cylindrical Point Density (pts/m^2)	863	616	141	76.351
Number of Cylindrical Points	16,425	17,117	2,743	3,338
Max. Rough Density (pts/m^3)	482,291	427,195	1,677	425
Min. Rough Point Density (pts/m^3)	0.370	0.370	0.059	0.159
Mean Rough Point Density (pts/m^3)	3,971	3,173	124	135
Number of Rough Points	38,616	38,050	3,288	6,340

Table 3.4. LPD statistics for the original and down-sampled STLS2 datasets (Note that the dimensionality-based PCA classification is used for adaptive down-sampling)

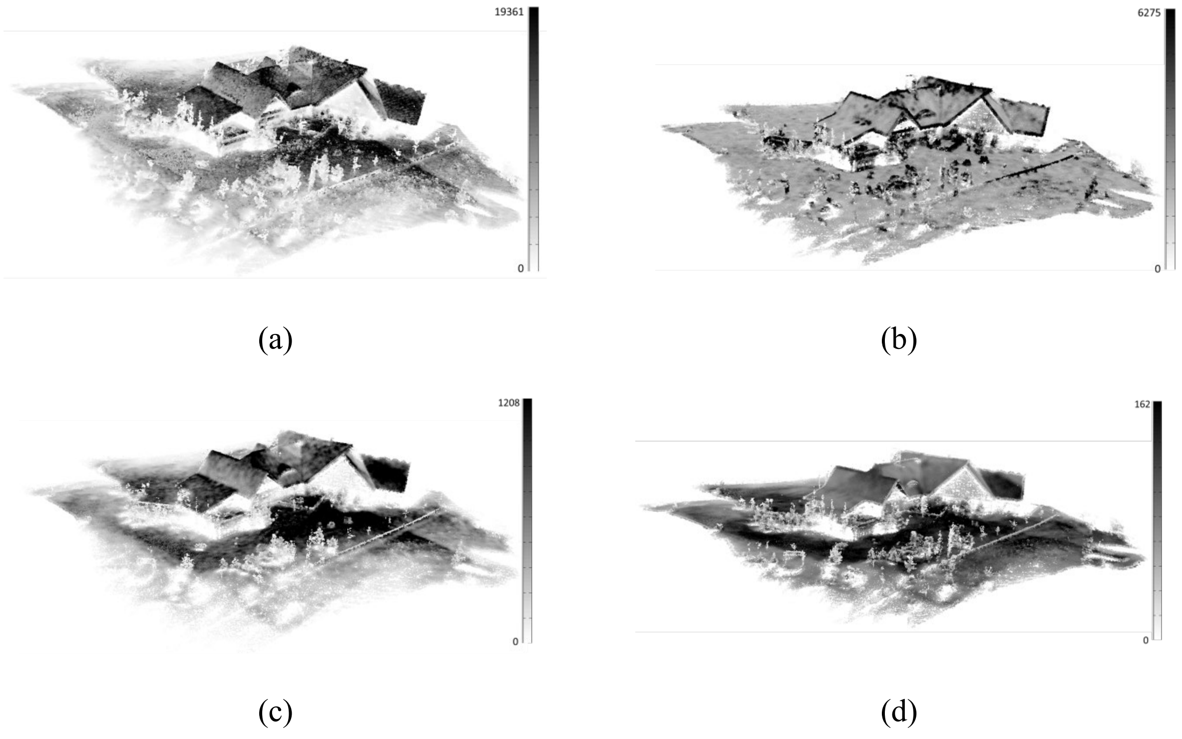
	Original	Adaptive down-sampling	Uniform down-sampling	Point-spacing-based down-sampling
Number of Points	1,916,238	1,077,791	1,077,791	408,723
Max. Planar Point Density (pts/m^2)	65,948	38,362	16,858	73.972
Min. Planar Point Density (pts/m^2)	0.001	0.001	0.001	0.001
Mean Planar Point Density (pts/m^2)	526	71	292	24.062
Number of Planar Points	1,266,090	434,340	704,631	225,504
Max. Linear Point Density (pts/m)	1,872	1,843	1,075	12.535
Min. Linear Point Density (pts/m)	1.669	1.669	5.960	1.875
Mean Linear Point Density (pts/m)	671	669	531	4.765
Number of Linear Points	12,938	12,954	3,292	19
Max. Cylindrical Density (pts/m^2)	41,584	41,901	30,805	74.338
Min. Cylindrical Point Density (pts/m^2)	0.830	0.830	1.709	0.731
Mean Cylindrical Point Density (pts/m^2)	1,420	1,176	851	19.288
Number of Cylindrical Points	170,941	162,847	87,220	14,644
Max. Rough Density (pts/m^3)	1,817,933	1,050,840	358,025	212
Min. Rough Point Density (pts/m^3)	≈0.000	≈0.000	≈0.000	≈0.000
Mean Rough Point Density (pts/m^3)	1,707	1,281	685	28.809
Number of Rough Points	466,269	467,650	282,648	168,556

Table 3.5. LPD statistics for the original and down-sampled DIM datasets (Note that the dimensionality-based PCA classification is used for adaptive down-sampling)

	Original	Adaptive down-sampling	Random down-sampling	Point-spacing-based down-sampling
Number of Points	4,027,753	1,364,996	1,364,996	447,047
Max. Planar Point Density (pts/m^2)	3,045	2,296	992	97.096
Min. Planar Point Density (pts/m^2)	0.019	0.019	0.018	0.018
Mean Planar Point Density (pts/m^2)	590	197	223	49.206
Number of Planar Points	3,069,682	588,963	1,268,002	389,970
Max. Linear Point Density (pts/m)	N/A	N/A	N/A	N/A
Min. Linear Point Density (pts/m)	N/A	N/A	N/A	N/A
Mean Linear Point Density (pts/m)	N/A	N/A	N/A	N/A
Number of Linear Points	N/A	N/A	N/A	N/A
Max. Cylindrical Density (pts/m^2)	4,094	3,991	689	84.418
Min. Cylindrical Point Density (pts/m^2)	4.236	3.315	5.975	3.600
Mean Cylindrical Point Density (pts/m^2)	303	360	106	29.550
Number of Cylindrical Points	60,547	102,013	23,317	7,180
Max. Rough Density (pts/m^3)	19,361	18,049	2,555	130.526
Min. Rough Point Density (pts/m^3)	0.028	0.028	0.087	0.026
Mean Rough Point Density (pts/m^3)	2,308	1,591	271	38.445
Number of Rough Points	897,524	674,020	73,677	49,897

Table 3.6. LPD statistics for the original and down-sampled DIM datasets (Note: the threshold-based PCA classification is used for adaptive down-sampling)

	Original	Adaptive down-sampling	Uniform down-sampling	Point-spacing-based down-sampling
Number of Points	4,027,753	761,825	761,825	447,047
Max. Planar Point Density (pts/m^2)	3,083	1,050	598	93.005
Min. Planar Point Density (pts/m^2)	0.019	0.019	0.017	0.010
Mean Planar Point Density (pts/m^2)	634	73.519	128	47.189
Number of Planar Points	3,706,080	519,580	697,767	367,411
Max. Linear Point Density (pts/m)	N/A	N/A	N/A	N/A
Min. Linear Point Density (pts/m)	N/A	N/A	N/A	N/A
Mean Linear Point Density (pts/m)	N/A	N/A	N/A	N/A
Number of Linear Points	N/A	N/A	N/A	N/A
Max. Cylindrical Density (pts/m^2)	3,913	3,383	315.478	74.746
Min. Cylindrical Point Density (pts/m^2)	4.236	2.683	3.938	1.482
Mean Cylindrical Point Density (pts/m^2)	262	247	59.215	27.944
Number of Cylindrical Points	44,775	73,524	10,050	8,260
Max. Rough Density (pts/m^3)	19,361	6,275	1,208	162.629
Min. Rough Point Density (pts/m^3)	0.002	0.002	0.118	≈0.000
Mean Rough Point Density (pts/m^3)	882	181	77.530	60.880
Number of Rough Points	276,898	168,721	54,008	71,376



(a)

(b)

(c)

(d)

Figure 3.11. Point density maps of (a) original as well as the (b) adaptively, (c) uniformly, and (d) point-spacing-based down-sampled DIM datasets (Note: the scale bars show the LPD along planar surfaces in $pnts/m^2$)

Through closer inspection of the reported LPD values in Table 3.3 through Table 3.6, one can see that the minimum point density for planar local neighborhoods within the original and adaptively down-sampled datasets were identical, which confirmed the expectation that the adaptive down-sampling approach retains points in sparse planar neighborhoods. However, within the uniform and point-spacing-based down-sampled datasets, the minimum point density decreased since neither one of these approaches considers the LPD during the thinning procedure. Additional observations regarding the LPD estimates for the original and down-sampled datasets include the following:

1. For the MTLS and STLS2 datasets, the adaptively down-sampled datasets had the closest point density along planar neighborhoods to the desired one when compared with the uniform and

point-spacing-based down-sampled datasets (refer to the Mean Planar Point Density in Table 3.3 and Table 3.4).

2. For all the datasets, the adaptively down-sampled datasets had the closest point density for the linear/cylindrical and rough regions to that associated with the original point cloud. On the other hand, the uniform and point-spacing-based down-sampling procedures led to significant reduction in the LPD for linear/cylindrical and rough neighborhoods (refer to the Mean Linear/Cylindrical/Rough Point Density values in Table 3.6).

3. For the DIM point cloud, the uniform down-sampling led to insufficient reduction of the LPD of planar neighborhoods when compared to the desired one.

4. For the DIM point cloud, the point-spacing-based down-sampling led to a point density closer to the desired one when compared with the adaptive down-sampling. This was attributed to the relatively high noise level within the established extent by the number of points used for defining the local neighborhoods. More specifically, the high noise-level-to-area ratio led to the fact that some of the rough points changed their classification to planar points after thinning some of the planar points. This can be seen in the significant reduction and the number of points that belong to rough neighborhoods (refer to the Number of Rough Points in Table 3.5 and Table 3.6 for the original and adaptively down-sampled datasets).

Comparing the dimensionality and threshold-based PCA classification approaches, one can make the following observations:

1. The dimensionality-based PCA was quite conservative in terms of the planar classification of the local neighborhoods. In other words, for datasets where the noise level within the extent defined by a given number of points was high, which was the case for the DIM data, relatively planar local neighborhoods were classified as rough neighborhoods. This sensitivity led to higher

point densities along planar neighborhoods after the adaptive down-sampling (refer to the Mean Planar Point Density in Table 3.5 for the adaptively down-sampled dataset).

2. The threshold-based PCA classification approach can be relaxed by using more tolerant thresholds to better classify relatively planar local neighborhoods in the presence of noise. However, relaxing the thresholds did not sufficiently make the LPD after the adaptive down sampling close enough to the desired one (refer to the Mean Planar Point Density in Table 3.6 for the adaptively down-sampled dataset).

Following the down-sampling of the MTLS, STLS2, and DIM datasets, the multi-class simultaneous segmentation and QC procedures were applied to the different point clouds. The multi-class simultaneous segmentation execution times and results after the QC procedure for the different datasets are shown in Table 3.7 and Table 3.8. One should note that the adaptively down-sampled DIM dataset in Table 3.7 and Table 3.8 refers to the one derived from threshold-based PCA classification since it produces better classification of planar features.

Table 3.7. Multi-class simultaneous segmentation execution times for the different datasets

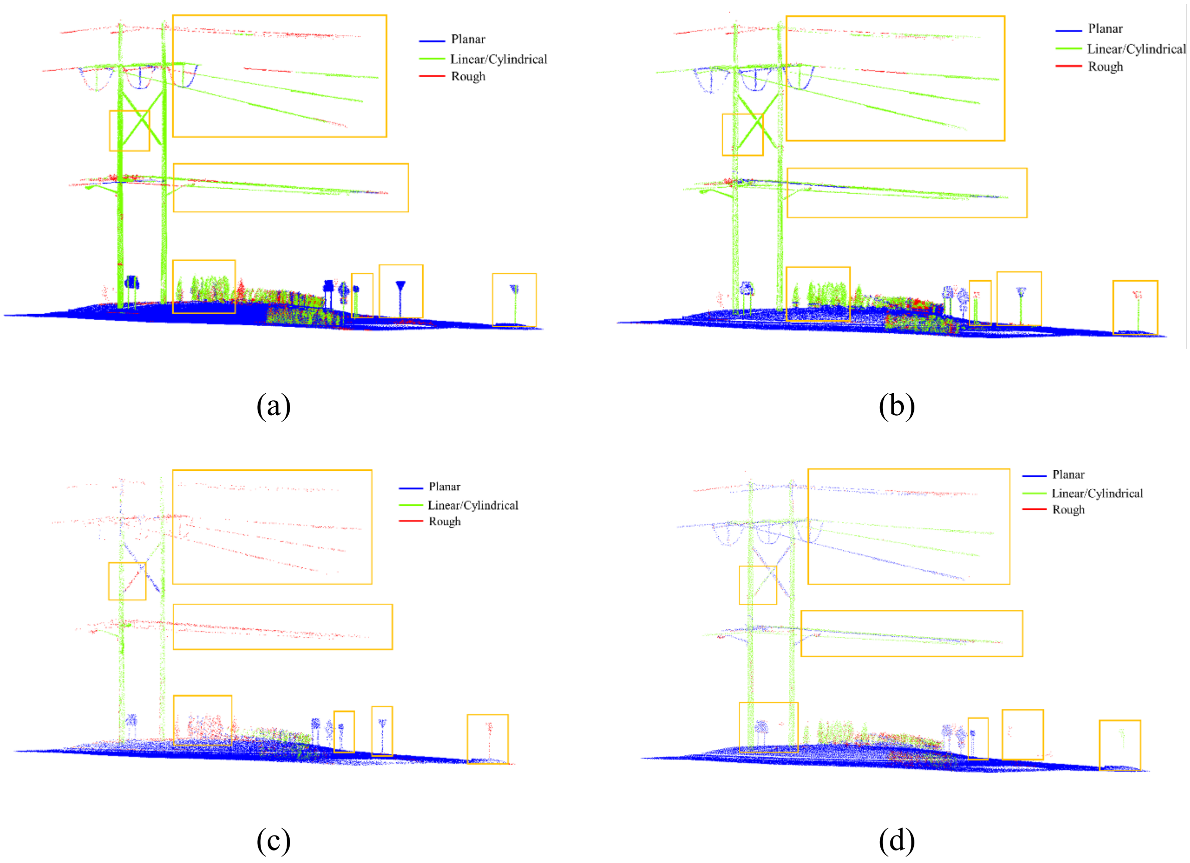
Dataset		Original Dataset	Adaptive down-sampled dataset	Uniform down-sampled dataset	Point-spacing-based down-sampled dataset
MTLS	Number of Points	1,105,962	139,738	139,738	62,033
	Time (hh:mm:ss)	01:33:50	00:04:49	00:04:13	00:01:33
STLS2	Number of Points	1,916,238	1,077,791	1,077,791	408,723
	Time (hh:mm:ss)	02:16:05	01:51:52	01:34:30	00:17:12
DIM	Number of Points	4,027,753	761,825	761,825	447,047
	Time (hh:mm:ss)	04:07:13	00:58:40	00:56:24	00:30:15

Table 3.8. Segmentation results after the QC procedure for the different datasets

	Original Dataset	Adaptive Down-sampled Dataset	Uniform Down-sampled Dataset	Point-spacing-based Down-sampled Dataset
MTLS				
Number of Planar Segments	40	24	20	22
Number of Linear/Cylindrical Segments	146	130	23	37
Number of Rough Segments	36	31	11	6
Total Number of Segments	222	185	54	64
STLS2				
Number of Planar Segments	1,687	1,257	1,128	792
Number of Linear/Cylindrical Segments	3,833	3,727	2,083	184
Number of Rough Segments	1	3	2	4
Total Number of Segments	5,521	4,987	3,212	980
DIM				
Number of Planar Segments	457	424	214	239
Number of Linear/Cylindrical Segments	794	846	155	133
Number of Rough Segments	28	31	3	3
Total Number of Segments	1,279	1,301	372	375

As shown in Table 3.7, the proposed down-sampling process led to a reduction in the segmentation execution time. In this regard, it is worth noting that the adaptive and uniform down-sampling approaches led to similar execution times since both of them produces the same number of points. The point-spacing-based down-sampling procedure, on the other hand, had the shortest execution time since it led to the smallest dataset. As for the segmentation outcome in Table 3.8, it is quite clear that the total number of segmented regions (planar, linear/cylindrical, and rough regions) was only maintained through the adaptive down-sampling procedure, which was confirmed by comparing the number of segmented regions from the down-sampled and original datasets). On the other hand, the uniform and point-spacing-based down-sampling procedures led to significant reductions in the number of segmented regions. In addition, the adaptive down-sampling led to the most complete segmentation outcome when compared with the segmentation of the original data. Besides maintaining the number of segmented regions, it also maintained the completeness of the segmented regions as will be shown in the next paragraph.

An illustration of the segmentation results for the original and down-sampled MTLS, STLS2, and DIM datasets are shown in Figure 3.12 – Figure 3.14. More specifically, Figure 3.12 portrays the segmented planar, linear/cylindrical, and rough regions in different colors, for the MTLS dataset. Figure 3.13 presents the planar and linear/cylindrical segmentation results for the STLS2 dataset with the different segments shown in different colors. Finally, Figure 3.14 presents the planar and linear/cylindrical feature segmentation results for the DIM dataset where the planar and linear/cylindrical features are shown in light and dark shades of grey, respectively. A closer inspection of Figure 3.12 reveals that the segmentation-based classification of the adaptively down-sampled dataset is the closest one to that derived from the original dataset (highlighted areas in subfigures 3.12 b–d). Figure 3.13 shows snippets that illustrate the comparative impact of different down-sampling techniques on the segmentation results from the STLS2 dataset. In these snippets, one can see that the segmentation outcome from the adaptively down-sampled dataset is the most complete one (the highlighted areas in subfigures 3.13 b–j). For the DIM segmentation results in Figure 3.14, it is clearly visible that the adaptive down-sampling procedure yielded the best extraction of the linear/cylindrical features, even when compared with the segmentation outcome from the original point cloud.



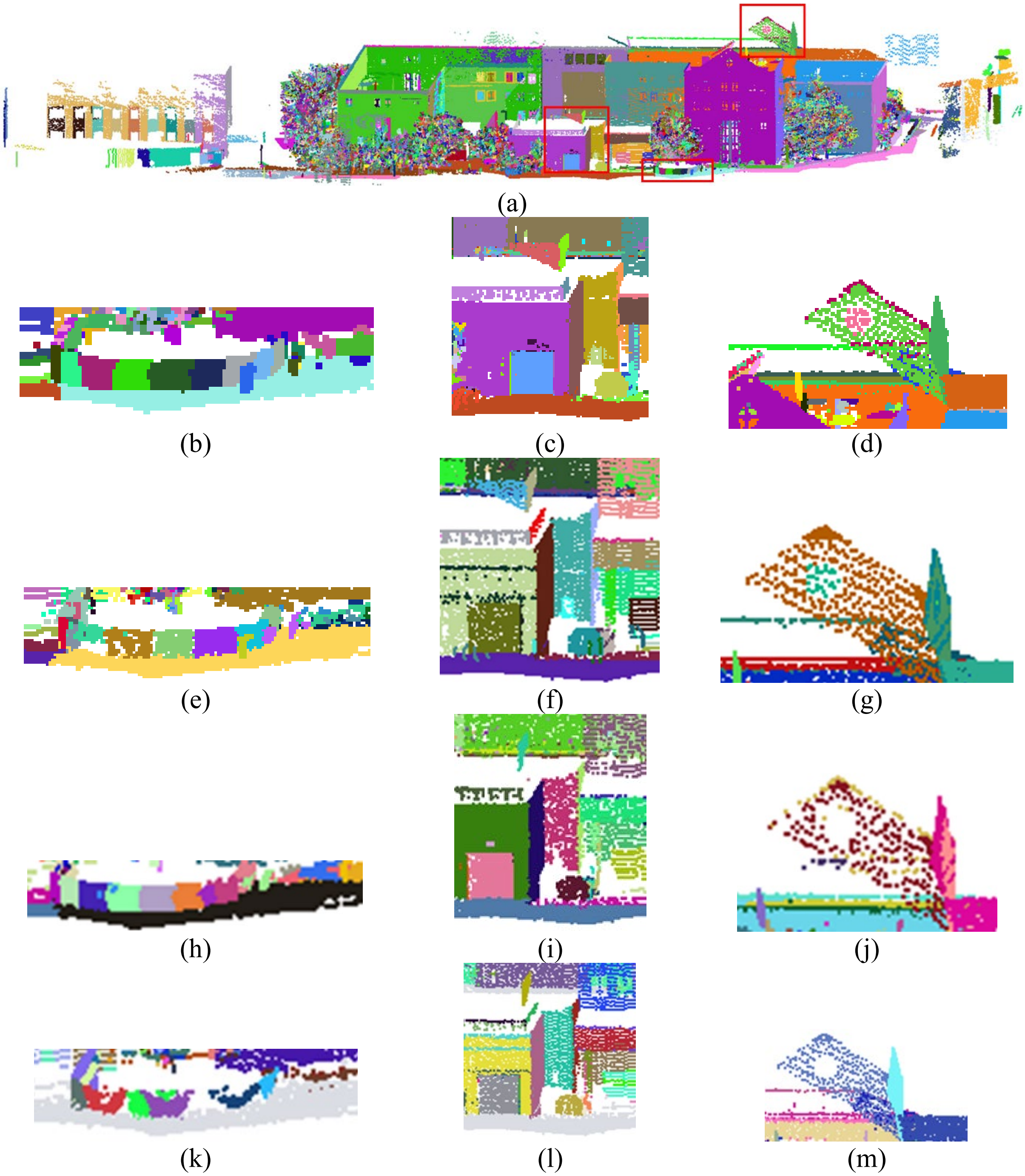
(a)

(b)

(c)

(d)

Figure 3.12. Segmentation-based classification of the (a) original MTLS dataset as well as (b) adaptively, (c) uniformly, and (d) point-spacing-based down-sampled datasets



(a)

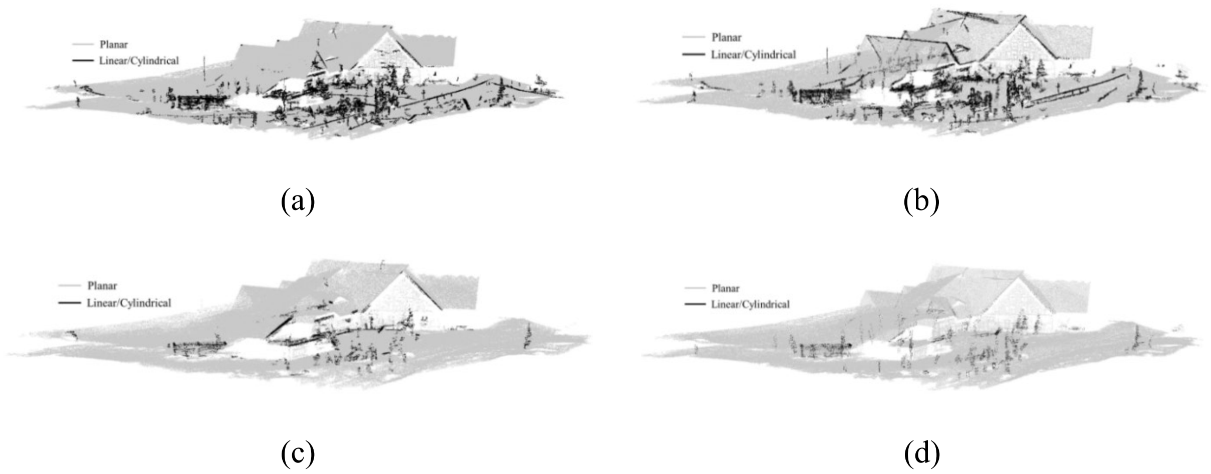
(b) (c) (d)

(e) (f) (g)

(h) (i) (j)

(k) (l) (m)

Figure 3.13. Planar and linear/cylindrical segmentation results of the (a) original STLS2 dataset as well as snippets that show zoomed-in areas of the segmentation outcome for the (b – d) original, (e – g) adaptively, (h – j) uniformly, and (k-m) point-spacing-based down-sampled datasets



(a)

(b)

(c)

(d)

Figure 3.14. Planar and linear/cylindrical feature segmentation results for the (a) original DIM dataset as well as the (b) adaptively, (c) uniformly, and (d) point-spacing-based down-sampled datasets

3.6.3 Experimental Results of Multi-Class Simultaneous Segmentation and QC Procedure

To illustrate the performance of the segmentation and QC procedure, this section provides the segmentation and QC results using ALS, STLS1, STLS2, STLS3, and DIM datasets. The main objectives of the conducted experiments were as follows:

1. Prove the feasibility of the proposed segmentation procedure in handling data with significant variations in LPD/LPS as well as the inherent noise level,
2. Prove the feasibility of the proposed segmentation procedure in handling data with different distributions and concentrations of planar, pole-like, and rough regions,
3. Prove the capability of the proposed QC procedure in detecting and quantifying instances of the hypothesized segmentation problems, and
4. Prove the capability of the proposed QC procedure in mitigating instances of the hypothesized segmentation problems.

The following subsections provide the segmentation results and the outcome of the QC procedure.

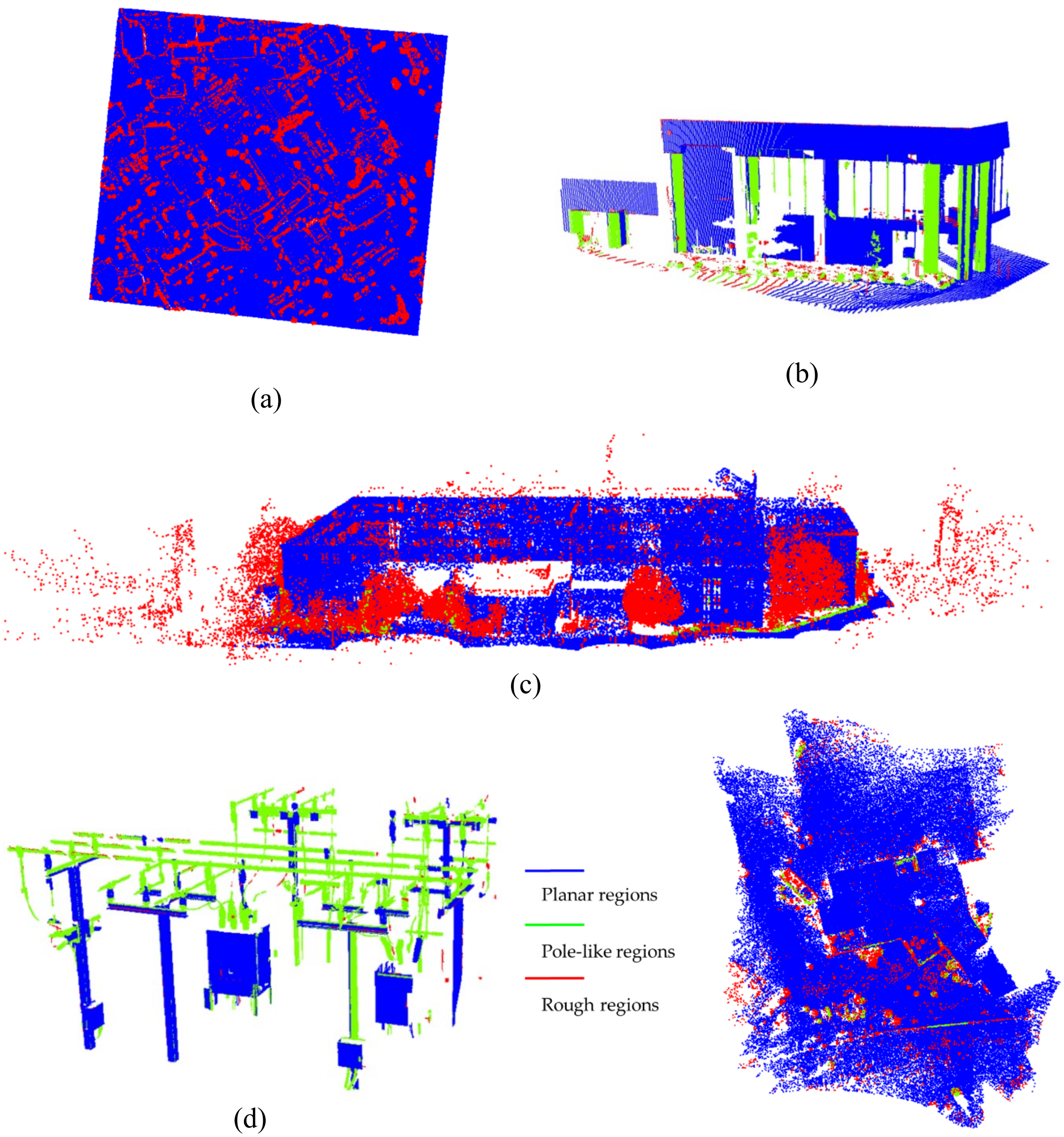
3.6.3.1 Multi-Class Simultaneous Segmentation Results

This section provides the segmentation results for the planar, pole-like, and rough regions from the different datasets. The proposed region-growing segmentation methodology involves three thresholds: 1) Percentage of randomly-selected seed points relative to the total number of available points within the dataset – For the above datasets, this percentage was set to 10%. One should note that using a larger percentage value did not make a significant impact on the segmentation results. 2) Pre-set size of the seed regions – This size should be set-up in a way to ensure that the seed region is large enough for reliable estimation of the model parameters associated with that region. For the conducted tests, the pre-set region size was 100. 3) Finally, the normal distance threshold – In general, the normal distance threshold for the region-growing process is based on the derived a-posteriori variance factor from the LSA parameter estimation procedure. However, the upper threshold values depended on the sensor specifications (i.e., the normal distance thresholds are not allowed to go beyond these values). For the conducted experiments, the ALS-based region-growing normal distance was set to 0.2 m. For the STLS and DIM datasets, the normal distance threshold was set to 0.05 m. The proposed methodology was implemented in C#. The experiments were conducted using a computer with 16 GB RAM and Intel(R) Core(TM) i7-4790 CPU @3.60GHz. The time performance of the proposed data structuring, characterization, and segmentation are shown in Table 3.9.

Table 3.9. Time performance of the proposed segmentation

	ALS 1	**STLS1**	**STLS2**	**STLS3**	**DIM**
Number of Points	812,980	170,296	201,846	455,167	230,434
Data Structuring and Characterization (mm:ss)	08:15	01:48	02:02	06:57	02:34
Segmentation Time (mm:ss)	11:40	02:55	01:33	06:37	03:56
Total Time (mm:ss)	19:55	04:43	03:35	13:34	06:30

Figure 3.15 and Figure 3.16 present the feature classification and segmentation results, respectively. For the classification results in Figure 3.15, the planar, pole-like, and rough regions are shown in blue, green, and red, respectively. As can be seen in Figure 3.15, the ALS, STLS2, and DIM datasets are mainly comprised of planar and rough regions. STLS1 and STLS3, on the other hand, mainly include planar and pole-like features, where large-radii pole-like features are present in STLS1 and the majority of STLS3 is comprised of small-radii cylinders. In Figure 3.16, the segmented planar, pole-like, and rough regions are shown in different colors. Visual inspection of the results in Figure 3.16 indicates that a good segmentation was achieved. To quantitatively evaluate the quality of this segmentation, the previously-discussed QC measures were used to denote the frequency of detected artifacts.



(a)

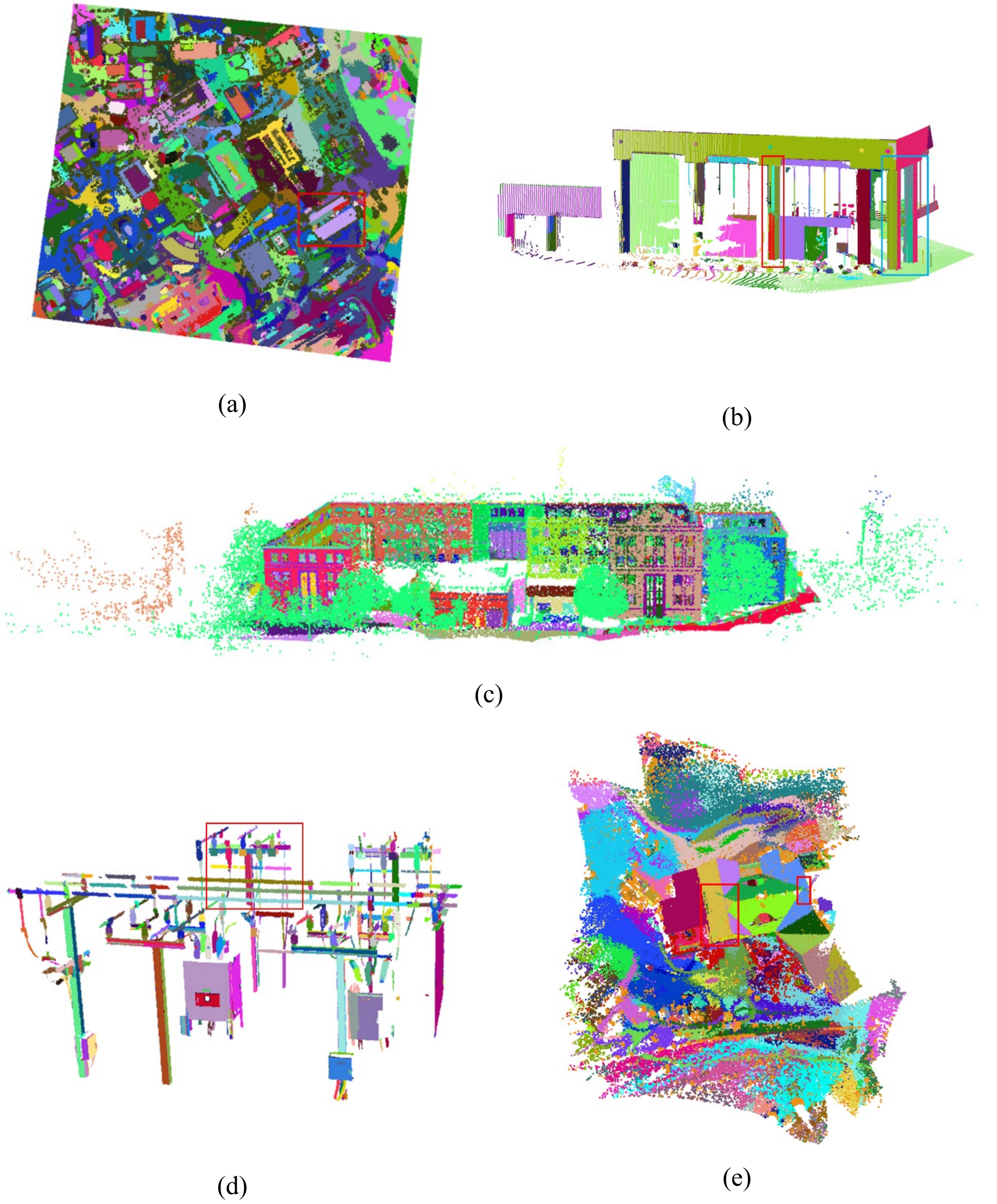
(b)

(c)

(d)

(e)

Figure 3.15. Perspective views of the classified point clouds for the ALS (a), STLS1 (b), STLS2 (c), STLS3 (d), and DIM (e) datasets (planar, pole-like, and rough regions are shown in blue, green, and red, respectively)



(a)

(b)

(c)

(d)

(e)

Figure 3.16. Perspective views of the segmented point clouds for the ALS (a), STLS1 (b), STLS2 (c), STLS3 (d), and DIM (e) datasets (different segments are shown in different colors)

3.6.3.2 Results of QC Outcomes

The QC procedure was implemented according to the following sequence: QC1) detection and mitigation of single pole-like features that have been misclassified as planar ones, QC2) initial mitigation of interclass competition for neighboring points, QC3) detection and mitigation of over-segmentation problems, QC4) detection and mitigation of intraclass competition for neighboring points, QC5) detection and mitigation of fully/partially misclassified pole-like features, and finally QC6) detection and mitigation of fully/partially misclassified planar features. One should note that for QC3, the respective over-segmentation measure was evaluated as the ratio between the merged segments in a given class relative to the total number of segments in that class. Figure 3.17 presents the segmentation results following these QC procedures. For STLS1 and STLS3, the segmentation results for planar and pole-like features are presented separately since those datasets have significant portions that pertain to such classes. Figure 3.18 illustrates examples of the detected/mitigated problems through the different QC measures. More specifically, Figure 3.18(a) shows portions of a cylindrical column, as highlighted by the red rectangle, that were originally classified as planar regions and after QC1 were correctly reclassified as pole-like features. Figure 3.18(b) shows examples of points from other classes, in red, that have been incorporated into planar and pole-like features, in yellow, after implementing QC2. An example of corrected over-segmentation of pole like features after QC3 is illustrated in Figure 3.18(c), compared to the segmentation results in Figure 3.16(b)). Detection and mitigation of intraclass competition for neighboring points after QC4 is shown in Figure 3.18(d) (refer to the highlighted regions within the red rectangles before and after QC4). The results of mitigating fully/partially misclassified linear regions after QC5 are shown in Figure 3.18(e) (refer to the results after the over-segmentation in Figure 3.18(c) and those in Figure 3.18(e), where one can see the correct mitigation of partially-misclassified pole-like features). Finally, Figure 3.18(f) shows an example of the segmentation results after applying QC6, which identifies/corrects partially/fully misclassified planar features (compare the results in Figure 3.16(b) and Figure 3.18(f)). The proposed QC procedures provide quantitative measures that indicate the frequency of the segmentation problems. Such quantitative measures are presented in Table 3.10, where closer investigation reveals the following:

1. For STLS1 and STLS3, which include a significant number of pole-like features, a higher percentage of misclassified single pole-like features (QC1) can be observed. STLS1 has pole-like features with larger radii, which means that there is a higher probability that the seed regions along the cylindrical features with a high point density are misclassified as planar regions. For TLS3, the misclassified pole-like features are caused by having several thin beams in the dataset.
2. For interclass competition for neighboring points (QC2), airborne datasets with predominantly planar features have a higher percentage of $QC_{interclass\ competition}(planar)$; refer to the results for the ALS and DIM datasets. On the other hand, $QC_{interclass\ competition}(pole-like)$ have a higher percentage of datasets with significant portions belonging to cylindrical features (i.e., STLS1 and STLS3).
3. Due to the inherent noise in the datasets as well as the strict normal distance thresholds as defined by the derived a-posteriori variance factor, over-segmentation problems (QC4) were present. In this regard, one should note that over-segmentation problems are easier to handle than under-segmentation, which can arise from relaxed normal-distance thresholds.
4. Intraclass competition for neighboring points (QC4) was quite minimal, which is evident by the reported low percentages for this category.

5. For partially/misclassified pole-like features, higher percentages of QC5 when dealing with a low number of points in such classes was not an indication of a major issue in the segmentation procedure (e.g., QC5 for ALS and DIM where the percentages of the points that belong to pole-like feature are almost 0% and 4%, respectively).
6. For partially/misclassified planar features, higher percentages of QC6 should be expected when dealing with datasets that have pole-like features with large radii or several interconnected linear features that are almost coplanar as was the case for STLS1 and STLS3, respectively.

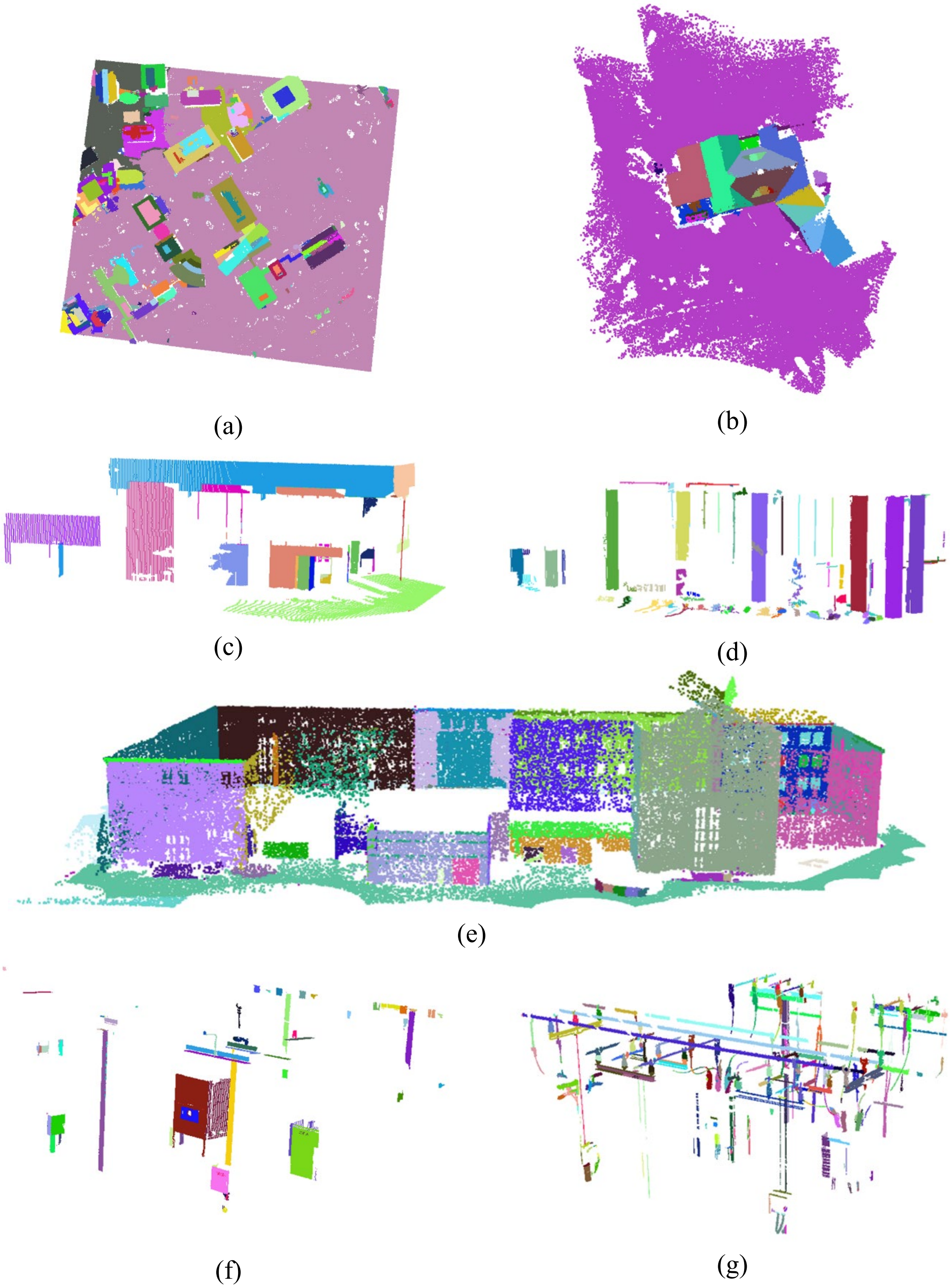
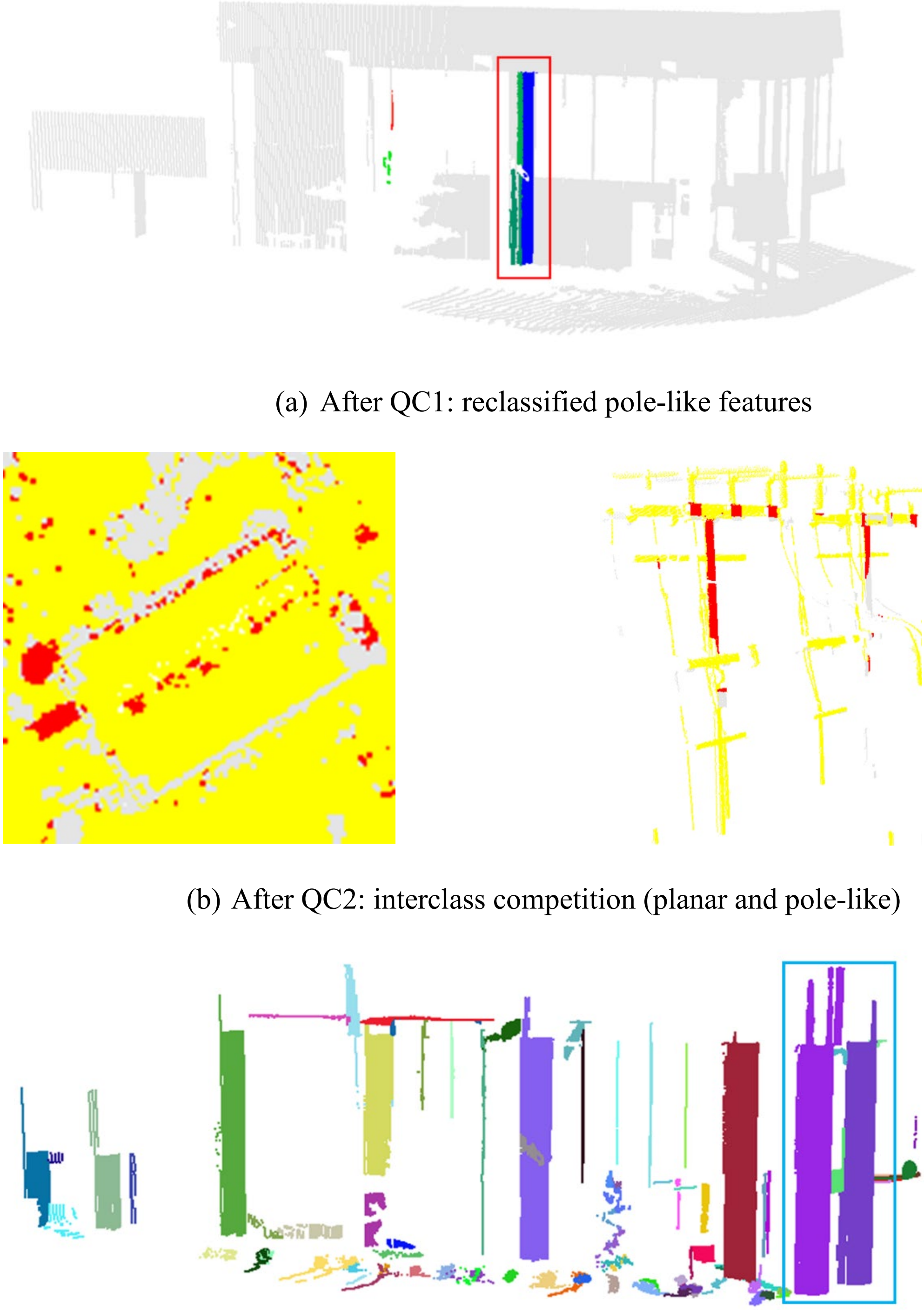


Figure 3.17. Perspective views of the segmented point clouds after the QC procedure for the ALS–planar (a), DIM–planar (b), STLS1–planar (c), STLS1–pole-like (d), STLS2–planar (e), STLS3–planar (f), and STLS3–pole-like (g) datasets; different segments shown in different colors

Table 3.10. QC measures for the different datasets

		ALS	STLS1	STLS2	STLS3	DIM
$QC1$	$n_{reclassified\ lines}/$	101/	5,439/	402/	25,484/	71/
	$n_{plane\ total}/$	716,628/	123,370/	126,193/	224,635/	211,553/
	$QC_{reclassified\ linear\ feaure}$	≈0.000	0.044	0.003	0.113	≈0.000
$QC2$ Planar	$n_{incorporated}/$	31,700/	5,457/	2,788/	24,469/	3,991/
	$n_{potential\ candidates}/$	96,453/	52,365/	76,055/	256,016/	18,952/
	$QC_{interclass\ competition}$	0.328	0.104	0.036	0.095	0.210
$QC2$ Pole-like	$n_{incorporated}/$	0/	22,193/	4,379/	29,340/	5,198/
	$n_{potential\ candidates}/$	812,879/	123,042/	194,014/	208,937/	227,486/
	$QC_{interclass\ competition}$	0	0.180	0.022	0.140	0.022
$QC3$ Planar	$n_{merged\ planar}/$	618/	23/	278/	8/	163/
	$n_{segmented\ planar}/$	801/	59/	367/	86/	195/
	$QC_{over\ segmentation}$	0.771	0.389	0.757	0.093	0.835
$QC3$ Pole-like	$n_{merged\ linear}/$	0/	21/	8/	152/	38/
	$n_{segmented\ linear}/$	4/	113/	144/	430/	55/
	$QC_{over\ segmentation}$	0	0.185	0.055	0.353	0.69
$QC4$	$n_{invading}/$	21,690/	857/	4,521/	5,427/	3,381/
	$n_{plane\ total}/$	748,227/	123,388/	128,579/	223,620/	215,473/
	$QC_{intraclass\ competition}$	0.028	0.006	0.035	0.024	0.015
$QC5$	$n_{encompassed\ line\ points}/$	101/	5,841/	1,866/	9,748/	3,427/
	$n_{line\ total}/$	101/	69,447/	12,211/	275,570/	8,146/
	$QC_{misclassified\ pole-like}$	1	0.084	0.152	0.035	0.420
$QC6$	$n_{misclassified\ planar}/$	N/A	29,647/	N/A	73,187/	N/A
	$n_{plane\ total}/$		123,388/		223,620/	
	$QC_{misclassified\ planar}$		0.240		0.327	



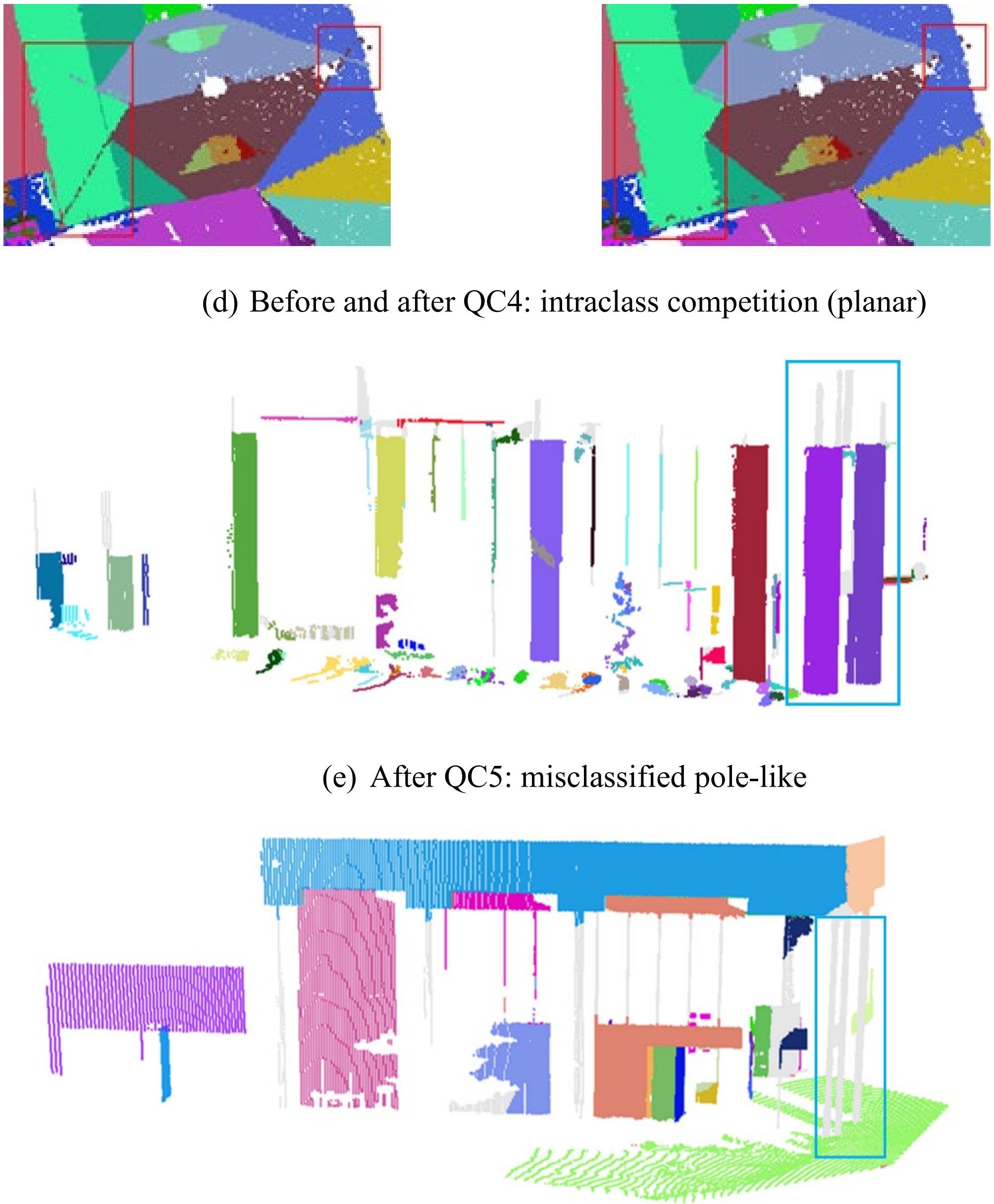
(a) After QC1: reclassified pole-like features

(b) After QC2: interclass competition (planar and pole-like)

(c) After QC3: over-segmentation (pole-like)

Figure 3.18. Examples of improved segmentation quality by the different QC measures

Figure 3.18. Continued



(d) Before and after QC4: intraclass competition (planar)

(e) After QC5: misclassified pole-like

(f) After QC6: misclassified planar

3.7 Summary

This chapter introduced a new framework for initial point cloud processing procedures, which include data characterization, adaptive down-sampling, multi-class simultaneous segmentation, and QC of segmentation outcomes. The presented adaptive down-sampling strategy only thins point clouds, which may be generated from either laser scanning or dense image matching techniques, in planar local neighborhoods while considering the local point density within such datasets. More specifically, a probabilistic approach is introduced in this thesis to randomly thin a given point cloud in planar local neighborhoods that exhibit higher local point density compared to a user-defined approach. This adaptive down-sampling strategy is based on the hypothesis that further processing of adaptively down-sampled datasets (segmentation of planar, linear/cylindrical, and rough regions) will not be compromised by the proposed thinning strategy. Moreover, the linear/cylindrical features may even be enhanced through the proposed down-sampling process, especially when a dataset has a large noise-level-to-area ratio.

4. DIGITAL BUILDING MODEL GENERATION FOR COMPLEX BUILDINGS

4.1 Introduction

After discussing the proposed segmentation technique, this chapter will address how the derived segments are used for a given application, specifically a digital building reconstruction application. The input segments of this application can come from general segmentation approaches. The previous steps ensured the derivation of reliable planar, pole-like, and rough segments. In this chapter, the desired objects are buildings with planar rooftops. Therefore, the procedures here forward focus on processing only planar segments. The digital building model generation procedure consists of the following steps: 1) building primitive boundary extraction and spike removal, 2) boundary regularization using Enhanced Recursive Minimum Bounding Rectangle (ERMBR) approach, 3) boundary regularization using breakpoint detection approach, 4) integration of the outcome of the two regularization strategies, 5) water-tight DBM generation. The flowchart of the proposed strategy is shown in Figure 4.1.

First, building primitives are identified from the planar segments using ground and non-ground classification and building hypothesis generation. Then, boundaries of primitives are extracted and a spike removal procedure is applied to remove the outliers. Due to the discrete nature of point clouds, the initially extracted building boundaries are irregular. Two boundary regularization strategies are introduced in this chapter. The first one is ERMBR approach which can regularize boundaries with right-angle corners. The second one is using breakpoint detection approach to regularize boundaries with multi-orientation and curved segments. The outcome of the two regularization strategies are integrated to produce building models with right-angle, multi-orientation, and curved boundaries. At the end, a water-tight DBM generation approach is presented to consider the topology between adjacent primitives and generate seamless building models.

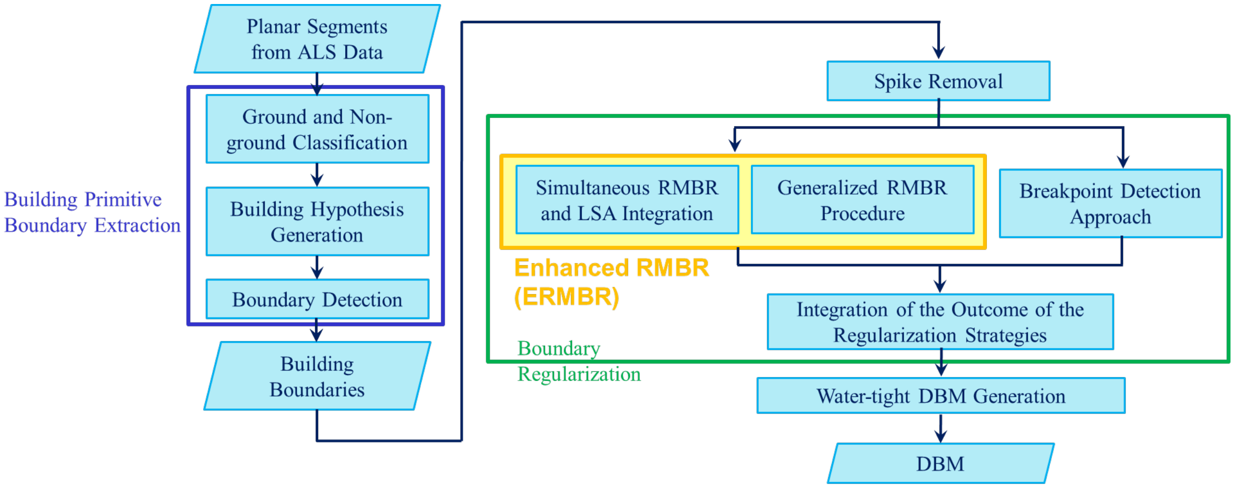
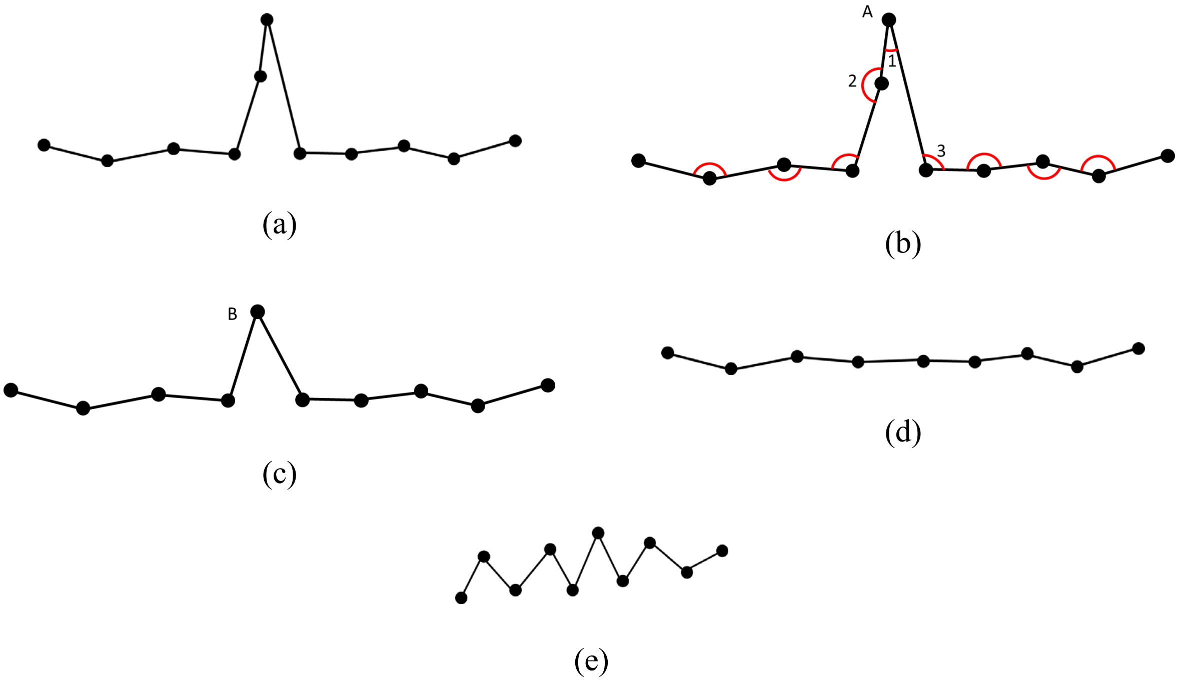

Planar Segments from ALS Data
Ground and Non-ground Classification
Building Hypothesis Generation
Boundary Detection
Building Primitive Boundary Extraction
Building Boundaries
Spike Removal
Simultaneous RMBR and LSA Integration
Generalized RMBR Procedure
Breakpoint Detection Approach
Enhanced RMBR (ERMBR)
Integration of the Outcome of the Regularization Strategies
Boundary Regularization
Water-tight DBM Generation
DBM

Figure 4.1. Flowchart of digital building model generation procedure

4.2 Building Primitive Boundary Extraction and Spike Removal

The extracted planar segments include both ground and non-ground segments. A heuristic ground/non-ground classification technique [88] therefore was implemented in this thesis to identify non-ground segments. This approach considers the physical properties (the area and slope) of the derived planar segments and the discontinuity among them (the ratio of the height difference to the planimetric distance (ΔH/D) as well as the height difference between the adjacent planar segments). For the extracted non-ground planar segments, two criteria are used to generate the building hypotheses. First, the area of the planar segments should not be too small. The second hypothesis is based on the height difference between the planar segment in question and its adjoining ground segment. Thus, when this height difference and the area of the planar segment are both greater than their respective thresholds, then the segment could be recognized as a building hypothesis. A building can be composed of several differently oriented planar surfaces, denoted here forward as the building primitives. Then, the sequential boundary points of the building primitives are tracked based on a search radius and are extracted using the minimum convex hull boundary detection method [86]. It is worth mentioning that the search radius for the boundary detection method is derived from the LPS of the building primitives.

The extracted boundary points of a building are irregular and might consist of points that do not belong to the rooftop. For example, this could occur when there is a tree next to the building primitive in question, which will be manifested as spikes in the boundary. A partial boundary with a spike is shown in Figure 4.2(a). To detect such spikes in a nearly straight boundary, the adjacent boundary points first are joined by line segments. From the figure, it can be seen that if the smaller angle between two adjacent line segments is smaller than a predefined threshold, then it could indicate a spike. Therefore, the included angle between each pair of adjacent line segments is calculated (Figure 4.2(b)), and the potential spikes are recognized (i.e., angle < Th_{angle}). However, a highly irregular boundary or dense boundary points could also result in small included angles (Figure 4.2(e)). In such cases, to avoid eliminating building boundary points, the angles between the preceding and succeeding pairs of edges (angles 2 and 3 in Figure 4.2(b)) are also considered. When the difference between the potential spike angle and its preceding or succeeding angles is large (i.e., angle > Th_{nei}), then the point in question is labeled as a spike and removed. For example, in Figure 4.2(b), if the difference between angles 1 and 2 or angles 1 and 3 are large, then the potential spike (point A) are removed, as shown in Figure 4.2(c). This process is repeated until no further spikes are found. After spike removal, Figure 4.2(a) is transformed into Figure 4.2(d).



(a)

(b)

(c)

(d)

(e)

Figure 4.2. Illustration of spike removal procedure: (a) a partial boundary with a spike, (b) smaller angles between two adjacent edges, (c) boundary after removing spike A, (d) boundary after removing spike B, and (e) highly irregular boundary

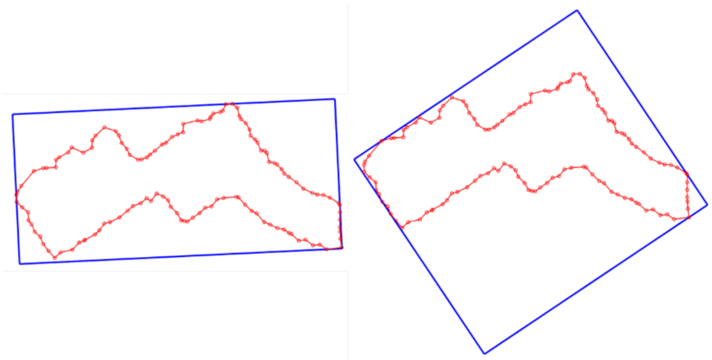
4.3 Boundary Regularization Using ERMBR Approach

Because of the discrete nature of a point cloud, the extracted boundaries are irregular. An ERMBR procedure is introduced in this sub-section to regularized boundaries with right-angle corners. First, the RMBR concept [7] and its shortcomings are discussed. Then, the proposed ERMBR procedure is introduced. The ERMBR procedure include two parts: (1) simultaneous RMBR and LSA integration and (2) generalized RMBR procedure.

4.3.1 Conceptual Basis of RMBR

This approach is based on the assumption that the buildings of interest have right-angled corners, i.e., either rectangular or a combination of adjoining rectangles. An MBR is a rectangle with the least area that can contain all the given points among the rectangles with arbitrary orientations that encompass the region comprised of those points [89]. The RMBR algorithm [7] uses a model-driven approach to determine the MBRs for the boundary points. In the case of a building that is composed of adjoining rectangles, the MBR will not be fully aligned with the entire boundaries. Thus, a recursive approach is applied to determine the boundary points that do not overlap with this 1st level MBR and subsequent MBR levels then are derived. The combination of these MBRs, as shown later, will define the shape of the derived building model.

To determine the 1st level MBR, the rotating calipers algorithm [90] is used. First, a local coordinate system (*UVW*) is defined based on the extracted boundary points. A boundary edge connecting two neighboring boundary points is defined as the u-axis, and then the v-axis follows a right-handed coordinate system. The bounding box is determined using (u_{max}, u_{min}, v_{max}, v_{min}) in the newly defined coordinate system and the area is calculated. Each of the boundary edges will result in a different local coordinate system, and a bounding box is determined for each of these cases. Finally, the bounding box with the least area is recognized as the MBR. However, such box may not align with the main orientation of the actual building, as illustrated in Figure 4.3. Hence, instead of using the area of the rectangle as the criterion, Kwak and Habib [7] used the number of points overlapping the boundary of the rectangle as the criterion (i.e., the bounding box with the largest value for this overlap ratio would be recognized as the MBR). One should note that the extracted boundary points are irregular so the overlapping points are defined using a threshold (Th_{nd}) on the normal distance of the points from the closest edge of the MBR. Kwak and Habib [7] determined this threshold according to the average point spacing of the whole dataset. However, in this thesis, it is determined according to the local point spacing for the building primitive in question.



(a)

(b)

Figure 4.3. (a) MBR with minimum area and (b) MBR with maximum overlapping points

As mentioned before, the 1st level MBR cannot capture all the details of some complex building boundaries, as shown in Figure 4.4(a) and Figure 4.4(b). Therefore, non-overlapping boundary points, as shown in Figure 4.4(c), are extracted to reconstruct sub-level MBRs. Every stretch of contiguous non-overlapping points marking the deviation from the 1st level MBR is regarded as a segment for constructing a new MBR. According to Kwak and Habib [7], a threshold for the number of such points (Th_{pt}) is used to decide whether to construct a sub-level MBR. This value is determined based on the point density and the desired level of detail. For the reconstruction of the second-level MBRs, non-overlapping points in each segment are projected to the corresponding edges of the 1st level MBR, as shown in Figure 4.4(d). Then, non-overlapping points of the segment as well as their projections are used together to define a new MBR (Figure 4.4(e)). After determining the 2nd level MBRs, the shape of the building model is updated according to Equation 4.1, where i represents the MBR level and n represents the total number of RMBR levels. The shape of the building model after the incorporation of the 2nd level MBRs is illustrated in Figure 4.4(f). Then, the 3rd level MBR is reconstructed, as illustrated in Figure 4.4(g-h). This process is iterated to determine subsequent MBR levels and to update the RMBR building model until no further non-overlapping points can be found. The final shape of building model is shown in Figure 4.4(i).

$$\text{RMBR building model} = \sum_{i=1}^{n}(-1)^{i+1}MBR_i \qquad (4.1)$$

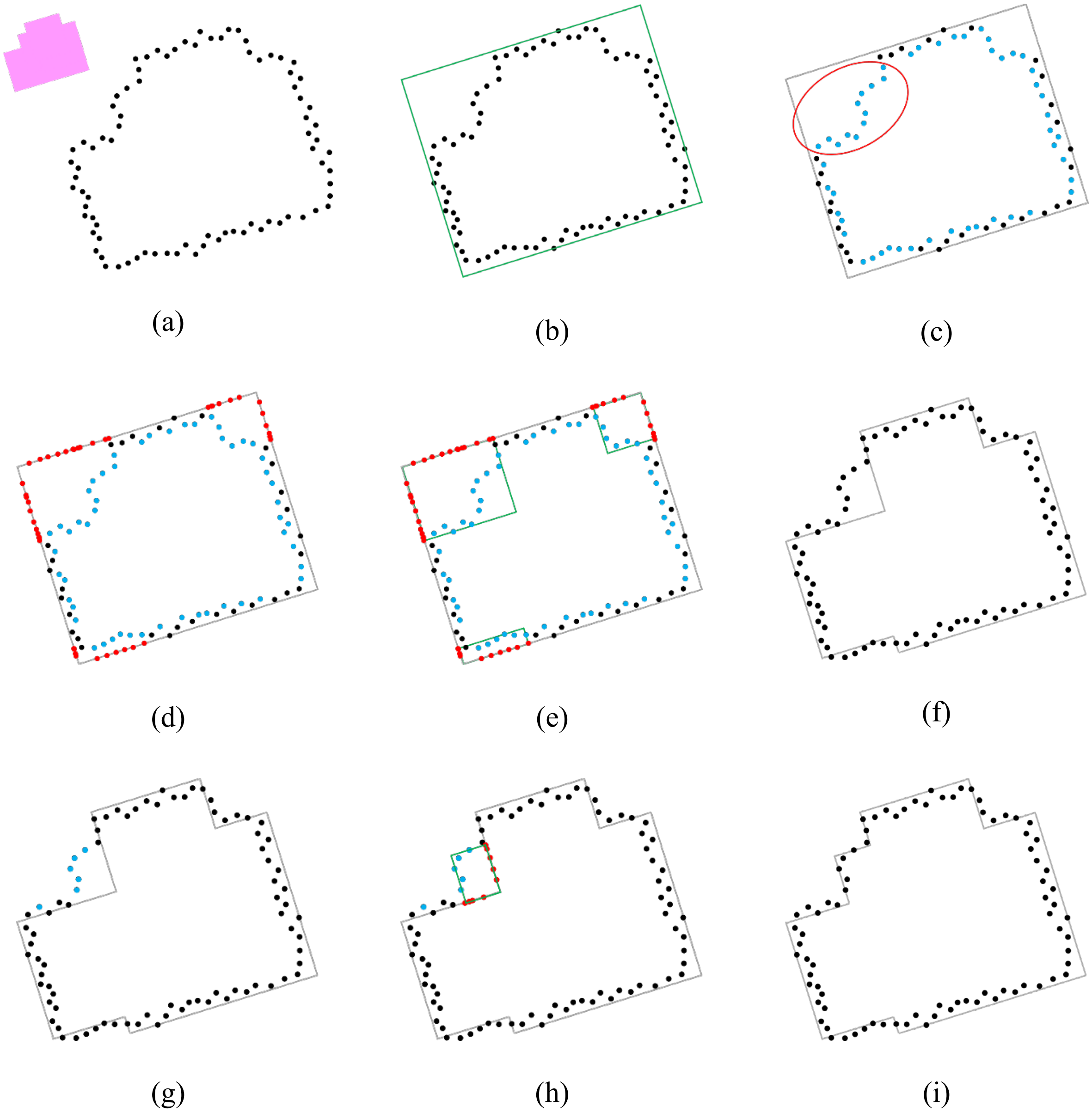
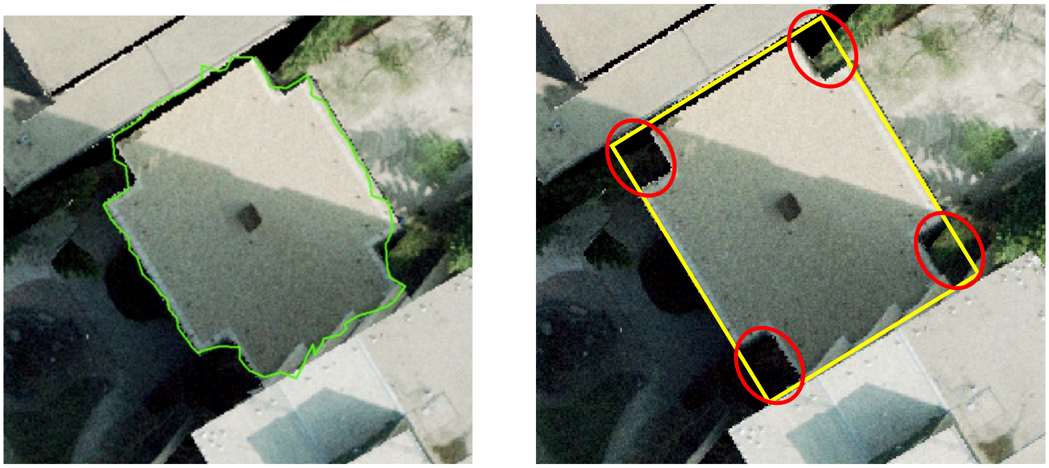


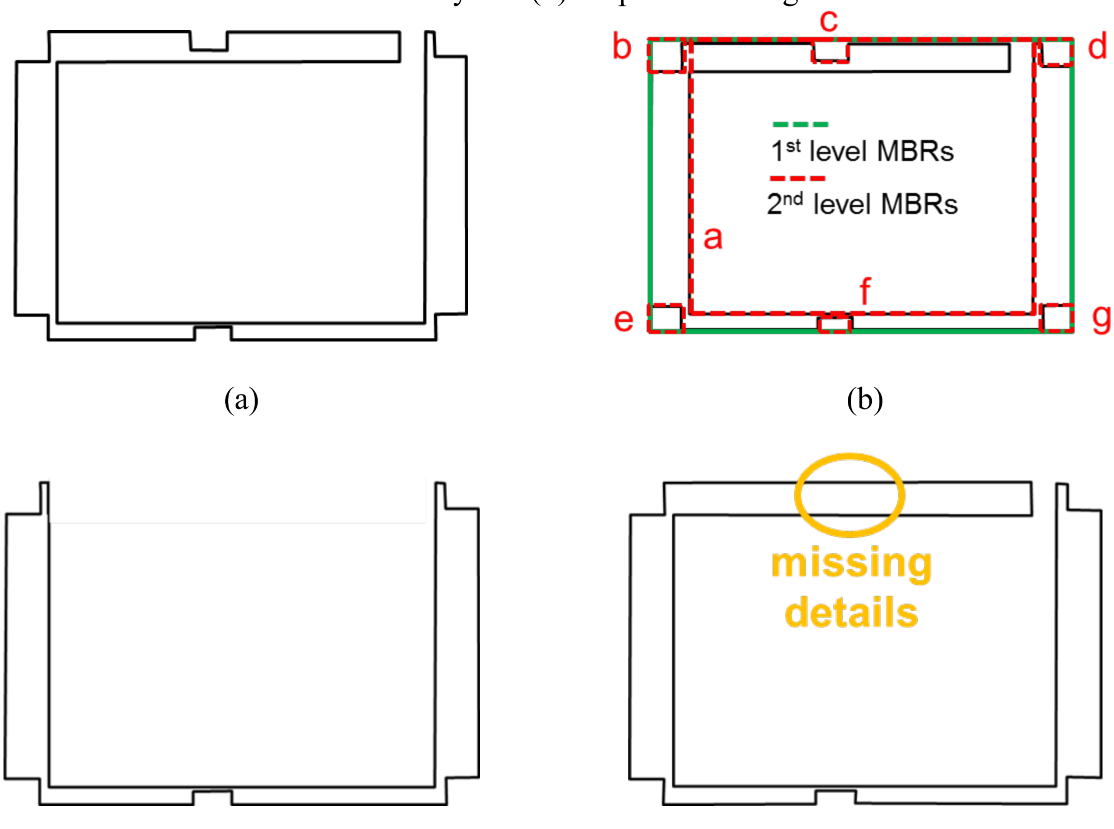
Figure 4.4. Illustration of the RMBR procedure: (a) initially-extracted boundary, (b) 1st level MBR generation, (c) non-overlapping point identification, (d) projected point generation, (e) 2nd level MBRs generation, (f) the shape of building model after the incorporation of the second-level MBRs, (g)-(i) repeat the procedure of (c) to (f) while considering 3rd level MBR

While this approach can retrieve the boundaries for most right-angled buildings, it has some shortcomings. First, in some cases, it can be sensitive to the normal distance threshold (Th_{nd}), which is used to define non-overlapping boundary points. As a result, if the extracted boundary points are highly irregular, then some of the points may be incorrectly classified as non-overlapping points for smaller threshold values, leading to erroneous sub-level MBRs. Second, this approach may be sensitive to the threshold (Th_{pt}) for the number of contiguous non-overlapping points, which is used to decide whether to use the corresponding segment to construct a sub-level MBR. If this threshold (Th_{pt}) is small, irregularities in the boundary points may be regarded as details as well and subsequently extracted as part of the boundary. On the other hand, when Th_{pt} is large, some important boundary details may not be extracted, such as indicated in red in Figure 4.5. Finally, the previous RMBR strategy is suitable in most cases, but some right-angled buildings, such as the one shown in Figure 4.6(a), cannot be correctly reconstructed as the 2nd level MBRs (rectangles a and c in Figure 4.6(b)) are overlapping. Subtracting these 2nd level MBRs from the 1st level MBRs will result in an incorrect building model, as shown in Figure 4.6(c); and after adding the third-level MBRs, the shape of building model is different from the actual building model, as shown in Figure 4.6(d).



(a) (b)

Figure 4.5. Illustration of incorrect building model as a result of inappropriate Th_{pt} value: (a) extracted boundary and (b) shape of building model



(a) (b)

(c) (d)

Figure 4.6. Illustration of previous RMBR strategy leading incorrect shape of building model (a) the shape of a building, (b) 2nd level MBRs (MBRs – b, d, e, f, and g as well as overlapping MBRs – a and c), (c) updated shape of the building model after subtracting the second-levels, and (d) incorrect final shape of building model after adding the 3rd level MBR

4.3.2 ERMBR Procedure

In this section, an ERMBR strategy is proposed that can overcome the shortcomings mentioned in the previous section. This approach is more generic and can generate right-angle building models without any exceptions. The generalized procedure includes two parts: (1) simultaneous RMBR and LSA approach and (2) generalized RMBR procedure.

4.3.2.1 Simultaneous RMBR and LSA Integration

1st Level MBR Reconstruction and LSA Integration

After applying spike removal to the LiDAR boundary points, the first-level MBR needs to be determined for the building primitive in question. In order to do this, a planar fitting first is done to derive the parameters for the best-fitting plane for the primitives in question. Next, a local coordinate system (*UVW*) is defined with the U and V-axes lying along the plane and the W-axis being normal to the plane. Then, the procedure described in Section 4.3.1 is used to determine the first-level MBR for the building primitive in this local coordinate system. As mentioned before, the number of boundary points overlapping with the bounding rectangle is used as the criteria to identify the MBR. These overlapping points are detected using a threshold (Th_{nd}) for their normal distance from the closest edge of the bounding rectangle. In previous research, Th_{nd} was determined according to the average point spacing of the whole dataset. However, the average point spacing for each building primitive could be different because of the variations in scan angles, surface orientation, surface reflectance properties, and overlapping strips. This thesis suggests that Th_{nd} should be defined as a function of the average local point spacing of each primitive. Before proceeding to extract the sub-level MBRs, the obtained 1st level MBR are refined to ensure that each edge of the 1st level MBR is the best-fitting line for the corresponding overlapping boundary points. Therefore, an LSA is applied to the 1st level MBR to obtain a more reliable rectangle. In the LSA procedure, the sum of the squared normal distances (*nd*) of the overlapping boundary points (shown in orange in Figure 4.7(a)) from the corresponding edges of the MBR would be minimized to derive a best-fitting rectangle. Five parameters are derived and used in the LSA to refine the 1st level MBR: a reference point *(u_0, v_0)*, length (L_1), width (W_1), and the orientation of the rectangle *(κ)* as shown in Figure 4.7(b). The target function is shown in Equation 4.2 and the normal distances for one of the edges of the rectangle are depicted in Figure 4.7(c). The refined MBR, as shown in Figure 4.7(d) as the red rectangle, would be aligned with the first-level overlapping boundary points.

$$nd = \frac{|(v_i - v_{i+1})u_l + (u_{i+1} - u_i)v_l + (u_i v_{i+1} - u_{i+1} v_i)|}{\sqrt{(u_{i+1} - u_i)^2 + (v_{i+1} - v_i)^2}} = 0 \qquad (4.2)$$

where, (u_l, v_l) denote the coordinates of the boundary point in question

(u_i, v_i) denote the coordinates of the corners of the first-level MBR and given by:

$$(u_i, v_i) = f(u_0, v_0, L_1, W_1, \kappa),\ i = 0{\sim}3$$

$$\begin{bmatrix} u_1 \\ v_1 \end{bmatrix} = \begin{bmatrix} u_0 \\ v_0 \end{bmatrix} + L_1 \begin{bmatrix} \cos\kappa \\ \sin\kappa \end{bmatrix}$$

$$\begin{bmatrix} u_2 \\ v_2 \end{bmatrix} = \begin{bmatrix} u_0 \\ v_0 \end{bmatrix} + L_1 \begin{bmatrix} \cos\kappa \\ \sin\kappa \end{bmatrix} + W_1 \begin{bmatrix} -\sin\kappa \\ \cos\kappa \end{bmatrix}$$

$$\begin{bmatrix} u_3 \\ v_3 \end{bmatrix} = \begin{bmatrix} u_0 \\ v_0 \end{bmatrix} + W_1 \begin{bmatrix} -\sin\kappa \\ \cos\kappa \end{bmatrix}$$

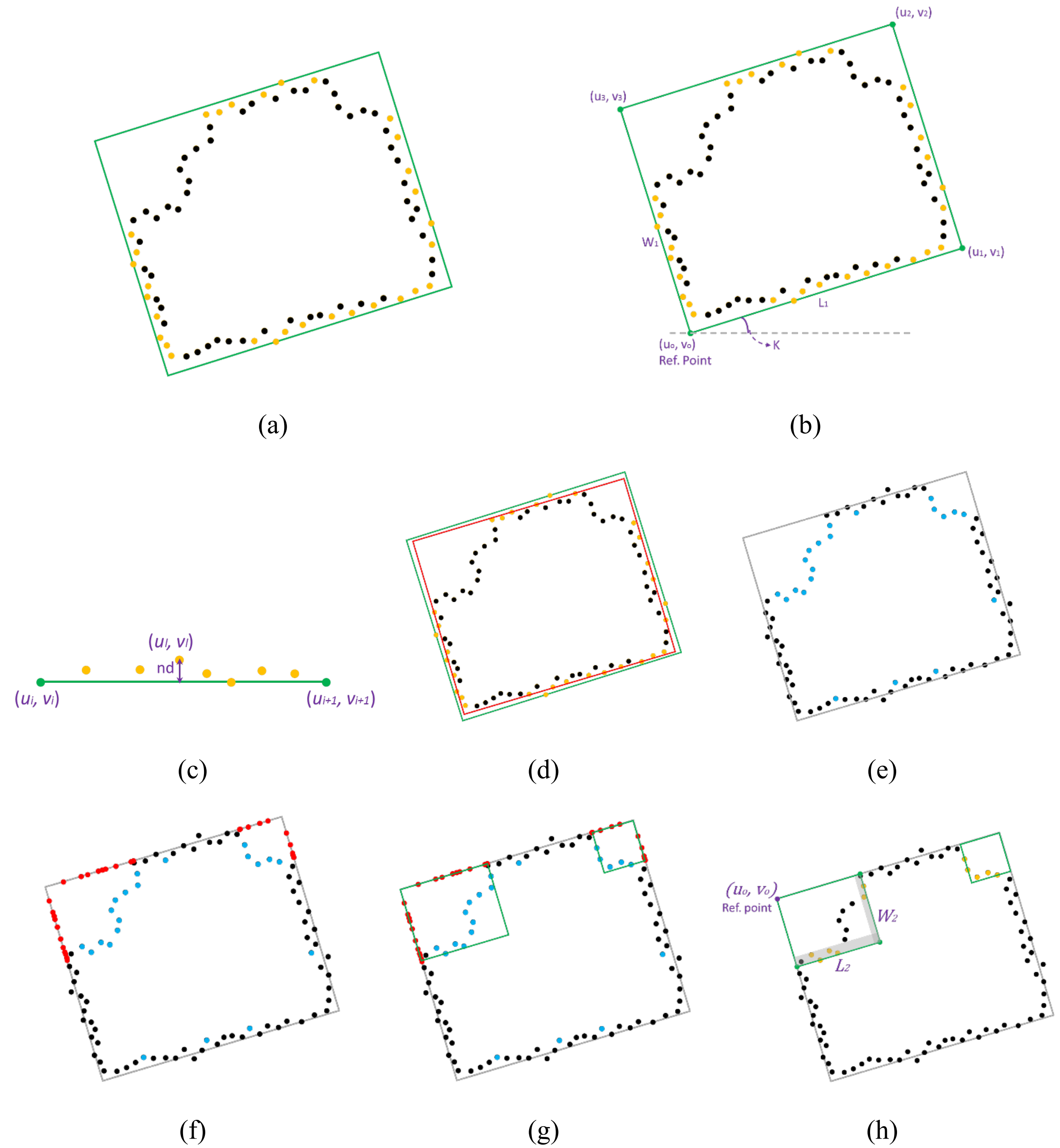
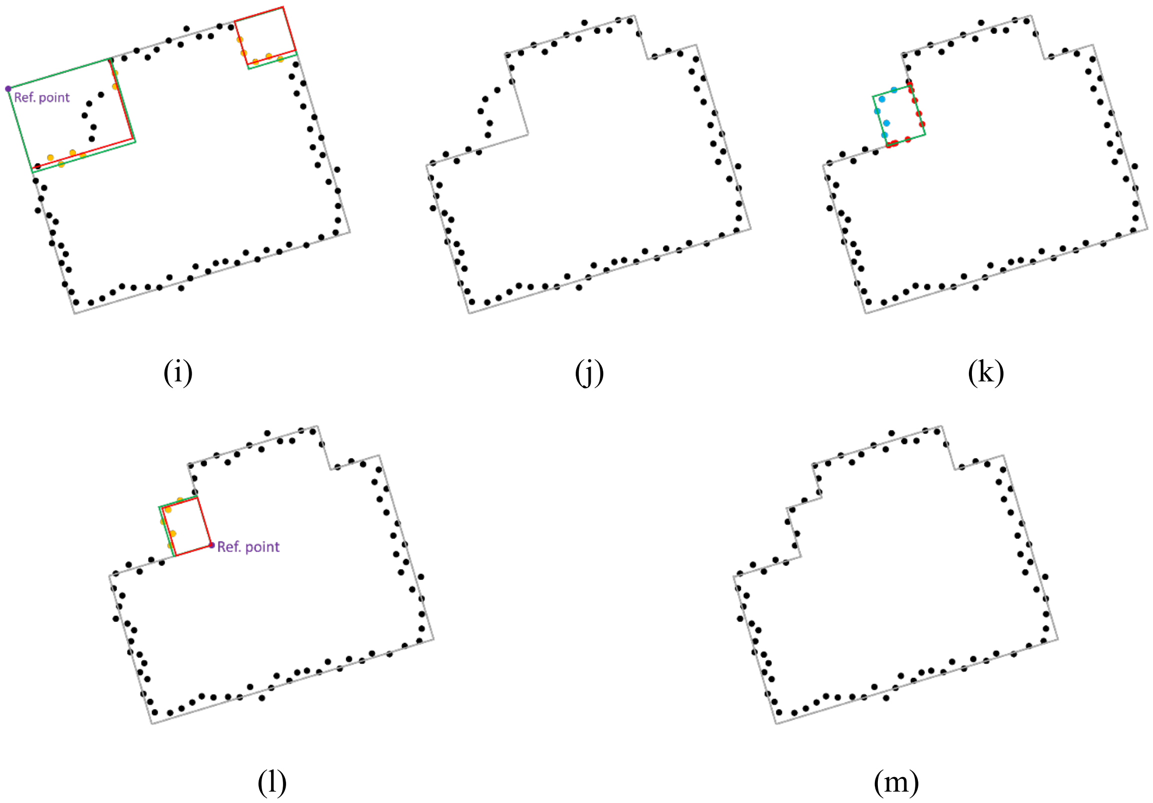


Figure 4.7. Illustration of simultaneous RMBR and LSA boundary regularization: (a) overlapping point identification (orange points), (b) derived parameters from 1st level MBR adjustment using LSA, (c) normal distance for an overlapping point to a corresponding edge of the MBR, (d) refined 1st level MBR in red, (e) identification of non-overlapping points (blue points), (f) projection of non-overlapping boundary points (red points), (g) 2nd level MBR generation, (h) LiDAR boundary points overlapping with 2nd level MBRs (orange points), (i) refined 2nd level MBRs in red, (j) the shape of building model after the incorporation of the second-level MBRs, and (k)-(m) repeat the procedure of (f) to (j)

Figure 4.7. Continued



(i)

(j)

(k)

(l)

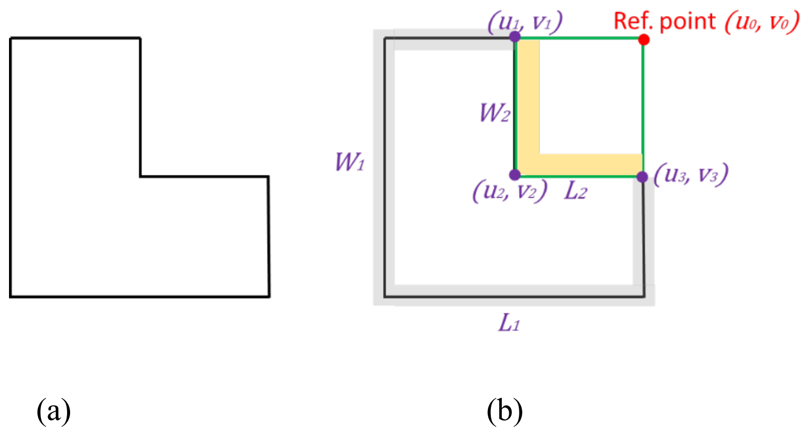
(m)

Sub-level MBR Reconstruction and LSA Integration

After deriving the refined 1st level MBR, the non-overlapping boundary points, whose distances are larger than the normal distance threshold (Th_{nd}), are detected (shown in blue in Figure 4.7(e)). As discussed before, previous research used the number of contiguous non-overlapping boundary points to determine whether to construct a sub-level MBR for the segment. However, this threshold (Th_{pt}) is sensitive to the local point spacing, the noise level of the data, and the desired level of detail. In the current approach, the size of a newly defined sub-level MBR is used instead to determine whether to incorporate it in the RMBR process. In order to do this, every segment containing contiguous non-overlapping boundary points is used to construct a sub-level MBR and its area then is calculated and checked against an area threshold (Th_{ar}). The sub-level MBR is constructed by first projecting the non-overlapping points in a segment to the corresponding edges of the current MBR, as shown in red in Figure 4.7(f). Then, these points, along with their projected points, are used to define an MBR with the same orientation as that of the 1st level MBR by using the projections on the two edges of the 1st level MBR to complete the sub-level MBR (Figure 4.7(g)). Again, this sub-level MBR needs to be refined in a similar manner as the 1st level MBR, except that the parameters being solved for in the LSA would be different as the orientation of the sub-level MBR will be the same as that of the 1st level MBR and is not to be included in the LSA. This is illustrated in Figure 4.7(h-i), but the selection of parameters will be discussed in more detail later. Finally, if the size of the refined sub-level MBR is larger than the threshold (Th_{ar}), it is incorporated into Equation 4.1 to update the shape of the RMBR building model, as shown in Figure 4.7(j). The newly derived current level then is used to derive further sub-level MBRs (Figure 4.7(k)), and this process is repeated until no further significant sub-levels can be generated. The final outcome can be seen in Figure 4.7(m).

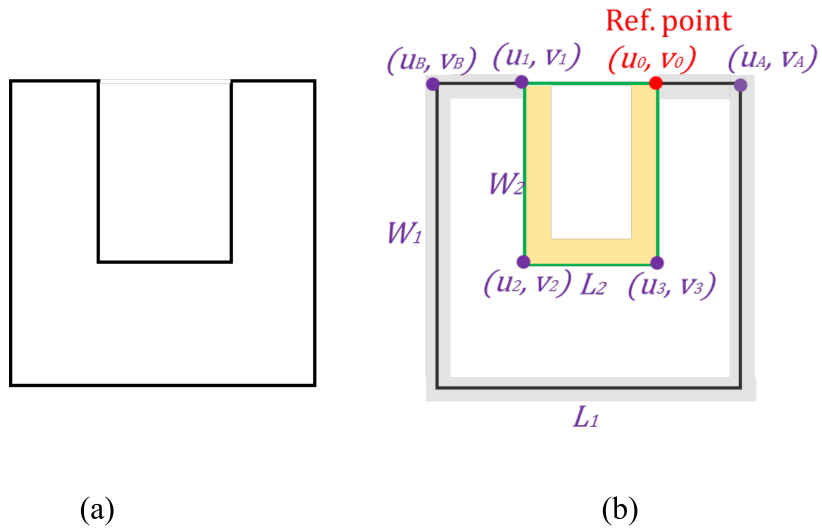
There are two general cases that can arise while generating and refining sub-level MBRs, each leading to a different parameter set. 1) The shared-corner case occurs when a sub-level MBR and its parent MBR share a common corner. Figure 4.8 illustrates a situation where the reference point *(u_0, v_0)* is common to the 1st and 2nd level MBRs. 2) The no shared-corners case occurs when a sub-level MBR and its parent MBR have no common corners. This is illustrated in Figure 4.9, where none of the corners of the 2nd level MBR *$(u_{0\sim3}, v_{0\sim3})$* belong to the 1st level MBR. For both cases, the four corners and the orientation of the 1st level MBR already have been adjusted. Therefore, these refined 1st level parameters should not be changed while modifying sub-level MBRs. The gray areas in Figure 4.8(b) and Figure 4.9(b) depict the section of the building boundary that is accurate and should not be adjusted, and the yellow area shows the zone to be adjusted within the sub-level MBR generation. For the shared-corner case, the only parameters to be modified are the length (L_2) and width (W_2) of the sub-level MBR (Figure 4.8(b)). As for the non-shared-corner case, the reference point is not fixed but rather is constrained to lie along the common edge between the first and second-level MBRs. Hence, the reference point coordinates *(u_0, v_0)* for the 2nd level MBR can be derived using the two corners defining the common edge in the 1st level MBR, as shown in Equation 4.3. In this case, three parameters are estimated in the LSA: the scale (λ), and the length (L_2) and width (W_2) of the sub-level MBR (Figure 4.9(b)).

$$\begin{bmatrix} u_0 \\ v_0 \end{bmatrix} = \begin{bmatrix} u_A \\ v_A \end{bmatrix} - \lambda \begin{bmatrix} u_B - u_A \\ v_B - v_A \end{bmatrix} \qquad (4.3)$$



(a) (b)

Figure 4.8. Refinement of sub-level MBR for a shared-corner case

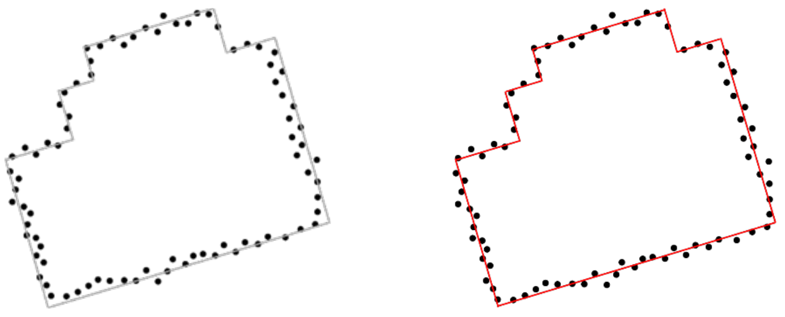

Ref. point
(u_B, v_B) (u_1, v_1) (u_0, v_0) (u_A, v_A)
W_2
W_1
(u_2, v_2) L_2 (u_3, v_3)
L_1

(a) (b)

Figure 4.9. Refinement of sub-level MBR for a non-shared-corner case

Global Boundary Refinement

As discussed earlier, an LSA is applied to each level of MBRs generated for a building primitive to increase the accuracy of the final building model. However, the target function for LSA used to refine these boundaries only takes into account the overlapping points that lie within the normal distance threshold (Th_{nd}), as defined by the LPS for that primitive. For example, some of the boundary points may not have been used for LSA initially as they may have a normal distance greater than the threshold, Th_{nd}. In order to ensure the generation of a more accurate building model, an iterative MBR edge refinement procedure therefore is adopted for each MBR edge while keeping its orientation unchanged, where the boundary points to be used in the edge refinement procedure are updated after each iteration. The objective function is to have the refined edge such that the sum of the normal distances of these overlapping boundary points from the edge is equal to zero. In the first iteration, the initially determined MBR edge is refined using the boundary points lying within the normal distance threshold (Th_{nd}). To derive a refined edge, a set of overlapping boundary points is identified and then a summation of normal distances with sign from overlapping points to the edge is calculated. The shifting amount of an MBR edge is based on the summation of normal distances and the shifting direction is perpendicular to the direction of the MBR edge. In the next iteration, a new set of overlapping boundary points is identified and used to further refine the edge. This process is repeated until the maximum number of overlapping points are detected for an edge (i.e., no more points could be added). This approach also ensures that the refined boundaries are generated using only the LiDAR points that actually define the boundary, instead of including the noise-induced points which might result in degradation of the quality of the building model. A building boundary before and after refinement are shown in Figure 4.10 (a) and Figure 4.10 (b), respectively.



(a) (b)

Figure 4.10. Illustration of boundary: (a) before refinement and (b) after refinement

4.3.2.2 Generalized RMBR Procedure

Using RMBR algorithm, reliable MBRs in different levels can be identified. These rectangles are used to derive the final shape of the building boundary by the alternate subtraction and addition of each sub-level of MBRs to the 1st level MBR, as given by Equation 4.1. However, as mentioned before, some right-angled buildings, as in Figure 4.6(a), cannot be accurately reconstructed using this approach. More specifically, this approach cannot be adopted when two or more MBRs belonging to the same sub-level have overlapping areas. The building primitives that can give rise to such a scenario are identified by checking whether the MBRs at the same sub-level have overlapping projections of the non-overlapping boundary points.

Figure 4.11 illustrates an instance of such a case. The green rectangle is the 1st level MBR. The red rectangles are 2nd level MBRs generated from the non-overlapping boundary points and their projections. The projections of the 2nd level MBRs (MBR_{2a} and MBR_{2c}) are overlapping. Therefore, those 2nd level MBRs are marked as *overlapping MBRs*. Moreover, the sub-level MBRs of an *overlapping MBR* are also labeled as *overlapping MBRs* and are grouped together (denoted as a single branch of *overlapping* MBRs). The contribution of these *overlapping MBRs* to the generation of RMBR is modeled differently.

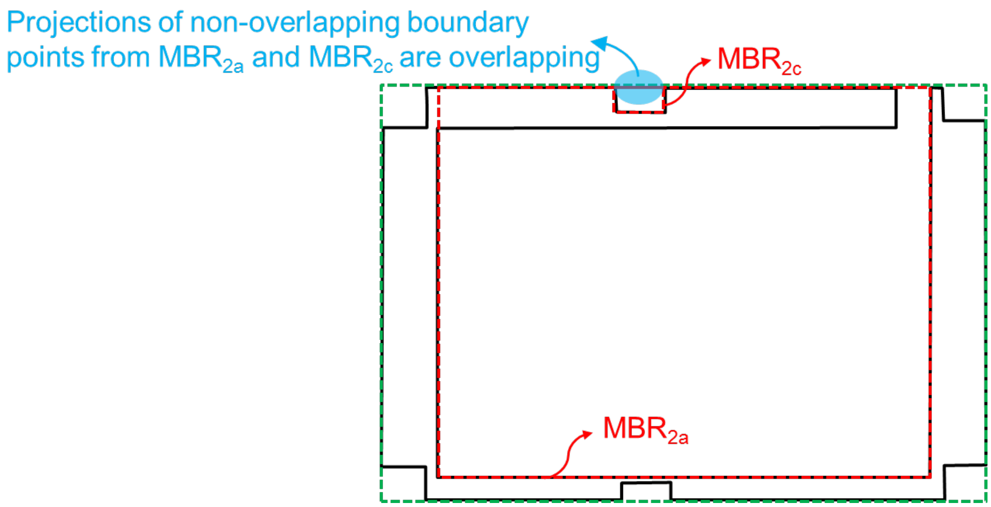

Projections of non-overlapping boundary points from MBR$_{2a}$ and MBR$_{2c}$ are overlapping
MBR$_{2c}$
MBR$_{2a}$

Figure 4.11. Detection of overlapping sub-level MBRs

The underlying principle of this newly devised model for the current case is that the general MBRs and each branch of *overlapping MBRs* are modeled individually. The model for the general MBRs (excluding the branches of *overlapping MBRs*) is the same as described before and is given by the term *A* (*General Model*) in Equation 4.4. Moreover, while modeling each branch of *overlapping MBRs* (consisting of a parent *overlapping MBR* and its sub-level MBRs), the parent *overlapping MBR* is considered to be the 1st level MBR (denoted by $oMBR_1$), and its sub-level MBRs are alternately subtracted or added to the former. The process is given by the term *B* (*Overlapping Model*) in Equation 4.4. Finally, subtracting *B* from *A* generates the final building model for such cases. This approach is discussed in detail later using an example and illustrated schematically in Figure 4.12.

$$\text{building model} = A - B \qquad (4.4)$$

$$General\ Model: A = \sum_{i=1}^{n} (-1)^{i+1} MBR_i$$

$$Overlapping\ Model: B = \sum_{j=1}^{m} (-1)^{j+1} oMBR_j$$

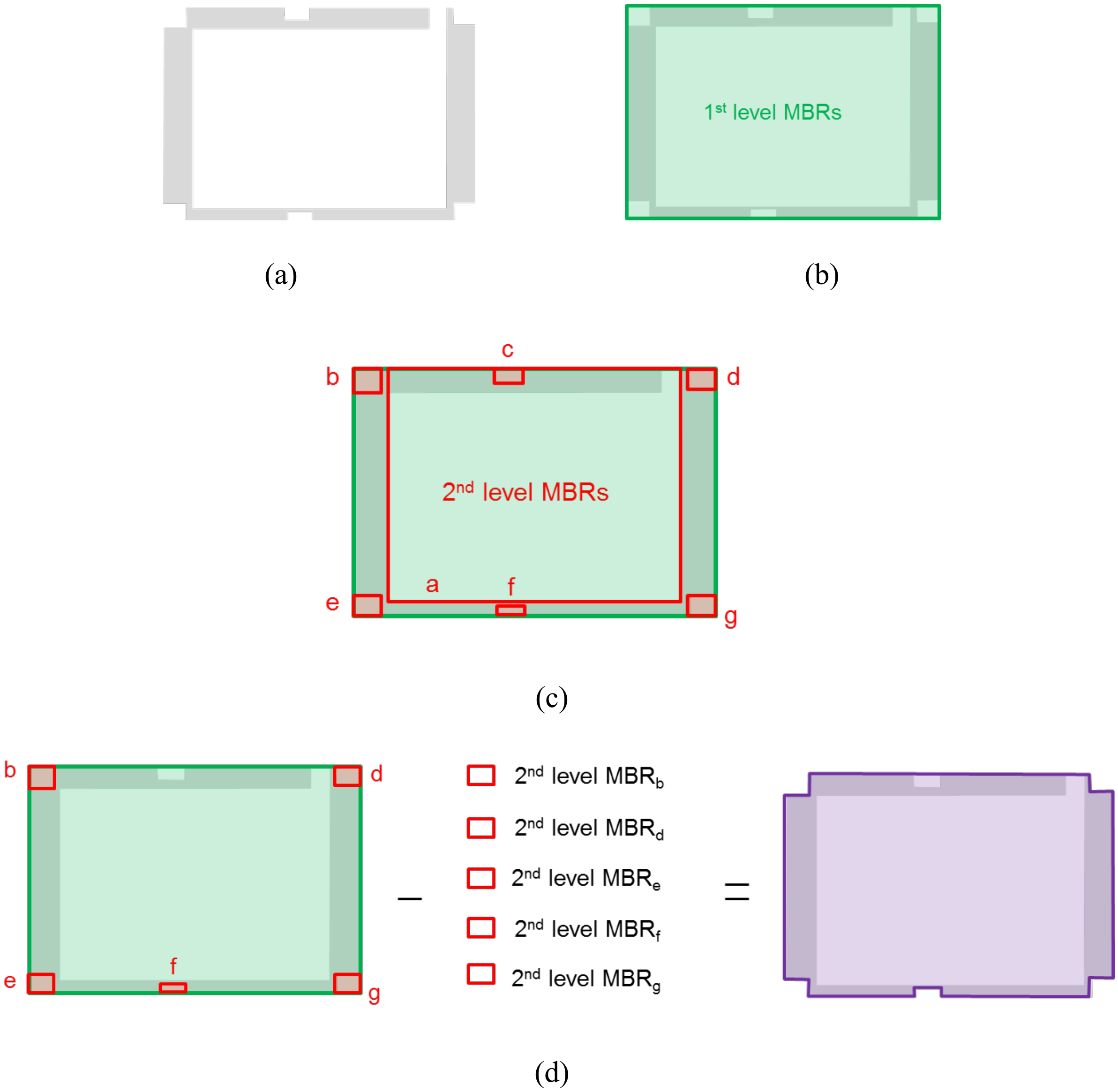
where, MBR_i = the MBRs at *i^th^* level

n = the maximum number of MBR

m = the maximum level in a branch of *overlapping MBRs* (*oMBR*)

Figure 4.12 (a) shows the building that was previously detected as having overlapping sub-level MBRs and the approach discussed above is applied to generate its building model. The green rectangle in Figure 4.12 (b) denotes the 1st level MBR for this building, and Figure 4.12 (c) shows the corresponding 2nd level MBRs. The 2nd level MBRs (b, d, e, f, and g) are general MBRs and, the 2nd level MBRs (a and c) are *overlapping MBRs* (as discussed previously in

Figure 4.11). Therefore, the general sub-level MBRs are subtracted from the 1st level MBR, as shown in Figure 4.12 (d). The resultant RMBR corresponds to the term *A* in Equation 4.4. MBR_a is an *overlapping MBR* and its sub-level MBR is depicted as blue rectangles in Figure 4.12 (e). These *oMBRs* are grouped together as a single branch. According to the definition of the term *B* in Equation 4.4, the blue rectangles are subtracted from the red rectangle, as shown in Figure 4.12 (f). Finally, the resultant RMBR corresponding to term *B* (*Overlapping model*) is subtracted from the RMBR corresponding to term *A* (*General model*) to generate the appropriate building model for this case, as shown in Figure 4.12 (g).



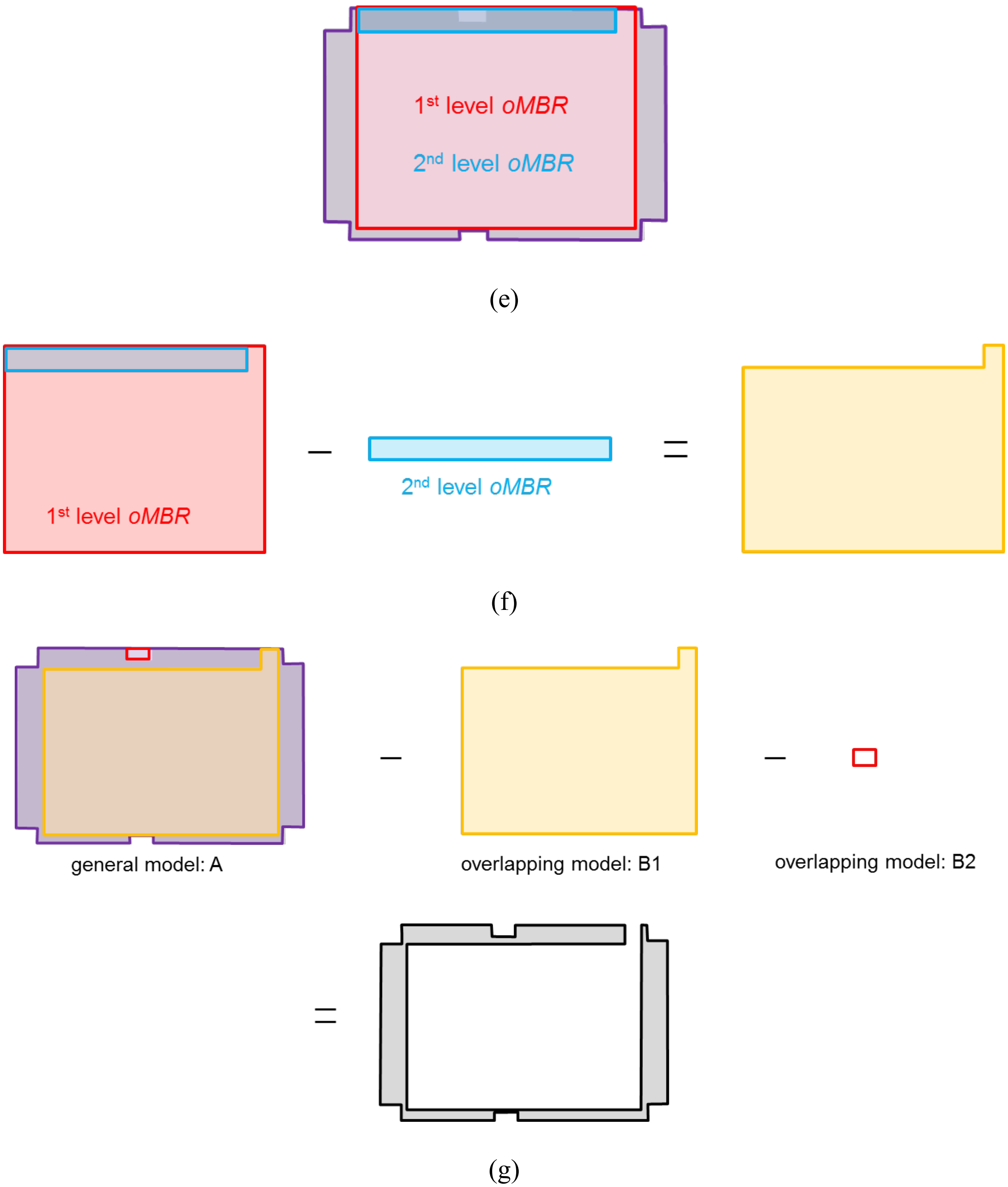
(a) (b)

(c)

(d)

Figure 4.12. Generalized RMBR procedure: (a) original building, (b) 1st level MBR, (c) 2nd level MBRs, (d) subtraction of general 2nd level MBRs, (e) RMBR process of *overlapping MBRs*, (f) subtraction of overlapping 2nd level *oMBR*, and (g) final building model obtained from the subtraction of overlapping models (B1 and B2) from the general model (A)

Figure 4.12. Continued



(e)

(f)

(g)

4.4 Boundary Regularization Using Breakpoint Detection Approach

The ERMBR approach is suitable for right-angle buildings with single orientation, but it cannot deal with multi-orientation buildings as well as curved boundaries. Therefore, this section presents another strategy to regularized boundaries using breakpoint detection approach. The approach, inspired from Forstner [91], *models for reconstructing one-dimensional surface profiles*. Forstner [91] reconstructed 1D profile under an assumption that the curvature between the consecutive profile points should be zero. According to the assumption, breakpoints of the profile can be detected since breakpoints do not have zero curvature between the consecutive points. The building boundary can have the same assumption that the curvature between the consecutive boundary points should be zero and corners can be detected because the corners do not follow the assumption. The breakpoint detection approach for 1D profile is introduced in Section 4.4.1. The breakpoint detection approach for 2D building boundary points is discussed in Section 4.4.2. Then, the boundary regularization using the detected breakpoints is introduced in Section 4.4.3.

4.4.1 Conceptual Basis of Breakpoint Detection Approach for 1D Profile

The one-dimensional profile model assumes the unknowns $\{x(i), i = 1,....,I\}$ are an equal space sequence. Since the distances between a set of given observations are different, the observations are densified into equal space. The observation with noise can be model as Equation 4.5. The noise of the observation is assumed to be independent, which means that the observations are statistically independent and have the same weights $w_y = 1/\sigma_y^2$.

$$y_m = x(i) + e_y \tag{4.5}$$

where,

$m = 1 \dots \dots M$

M: number of observations

$x(i)$: unknown profile value at the given observation with a range from 1 to M

e_y: noise ~ $(0,\sigma_y^2)$, independent of m

The objective is to estimate the unknowns from a given set of observations. An illustration of modeling a 1D profile is shown in Figure 4.13. The solid circles are the observations, and the hollow circles are the unknowns (the densified and smoothed points). To derive the unknowns, an assumption is introduced that the curvature should be zero for the neighboring densified points. Setting curvature as zero means that the profile should be smooth. The approximate curvature is derived by the second derivative. The stochastic constraint of the curvature is shown in Equation 4.6, which is also regarded as smoothness constraint. The smoothness constraint random deviation has mean zero and a standard deviation σ_c which can be used as a weight in profile reconstruction, as shown in Equation 4.7.

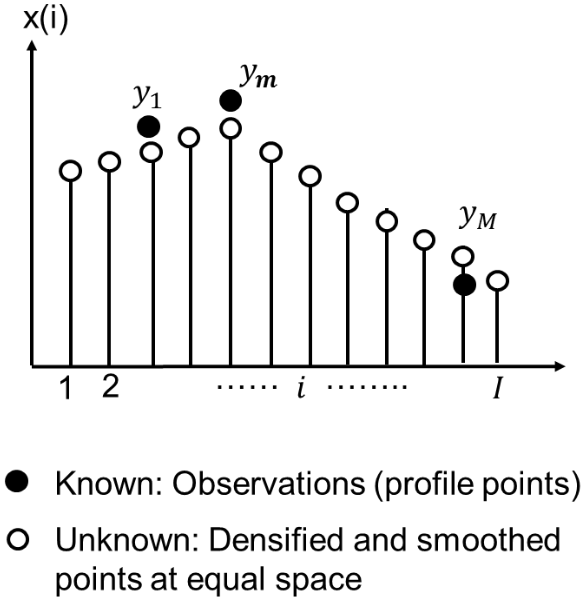

x(i)
y_1
y_m
y_M
1 2 ...... i I
● Known: Observations (profile points)
○ Unknown: Densified and smoothed points at equal space

Figure 4.13. Illustration of modeling the surface profile

$$c_i = x(i-1) - 2x(i) + x(i+1) + e_c \tag{4.6}$$

where,

i = 2, ... I-1

I = number of profile points to be densified and smoothed

e_c: smoothness random deviation ~ $(0,\sigma_c^2)$

$$w_c = 1/\sigma_c^2 \tag{4.7}$$

The profile reconstruction considers observation equations with stochastic constraints, as shown in Equations 4.8 and 4.9. Hence, the profile reconstruction can be expressed with observation equations and stochastic constraints as shown in Equation 4.10. The unknowns can be derived by minimizing the weighted sum of the squared residuals, as shown in Equation 4.11. Since the curvature, c_i, is assumed to be zero, the normal equation can be rewritten into Equation 4.12. The stochastic constraints have large residuals at breakpoints. The weights of stochastic constraints with large residuals are adjusted through an iterative LSA procedure while reducing the weight for the curvature constraint with large residuals. In the iterative LSA procedure [91], the Huber weight function is adopted in the first three iterations to ensure that the weights of stochastic constraints with large residuals can be reduced quickly and the global optimum is guaranteed. The exponential weight function is used in the last three iterations to ensure that stochastic constraints with large residuals have no influence on the estimations. At the last iteration, the weights of the stochastic constraints with a small value are assigned a zero weight, and the weights of the rest constraints are assigned equal weight to derive the unknowns (densified and smoothed points).

$$y_m = x(i) + e_y,\ \ D(y_m) = \ \sigma_y^2\ , \quad where\ m = 1\ \ \dots\dots M \tag{4.8}$$

$$c_i = x(i-1) - 2x(i) + x(i+1) + e_c,\quad D(c_i) = \ \sigma_c^2,\ \ i = 2\ \ \dots\dots I - 1 \tag{4.9}$$

$$\begin{bmatrix} y \\ c \end{bmatrix} = \begin{bmatrix} A1 \\ A2 \end{bmatrix} x + \begin{bmatrix} e_y \\ e_c \end{bmatrix}\ , D\left(\begin{bmatrix} y \\ c \end{bmatrix}\right) = \begin{bmatrix} \Sigma_{yy} & 0 \\ 0 & \Sigma_{cc} \end{bmatrix} = \begin{bmatrix} \sigma_y^2\, I_M & 0 \\ 0 & \sigma_c^2\, I_{I-2} \end{bmatrix} \tag{4.10}$$

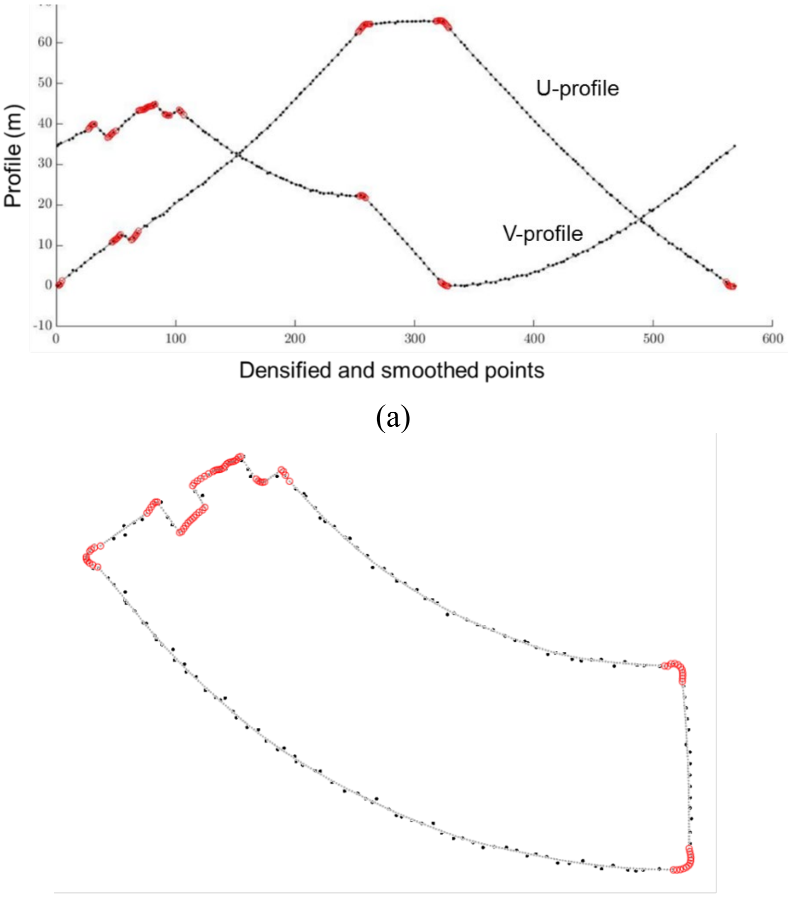
$$\Omega(x) = \ e_y^T \Sigma_{yy}^{-1} e_y + e_c^T \Sigma_{cc}^{-1} e_c \tag{4.11}$$

$$\left(A_1^T W_{yy} A_1 + A_2^T W_{cc} A_2\right) x = A_1^T W_{yy} l \tag{4.12}$$

4.4.2 Breakpoint Detection Approach for 2D Building Boundary

The introduced breakpoint detection strategy is used for a one-dimensional profile. However, LiDAR boundary points are in UV coordinates which mean that boundary points are in 2D. In order to use this approach, the building boundary is treated as two one-dimensional profiles in U-direction and V-direction, separately, to identify breakpoints.

First, the boundary points are regarded as observations in UV coordinates and the densified and smoothed points are treated as unknowns. The 1D breakpoint detection approach is applied to boundary points in both U and V directions, as shown in Figure 4.14 (a). A corner can be identified when a densified and smoothed point is detected as a breakpoint either in the U or V profile. Figure 4.14 (b) shows the boundary points in the U-direction and V-direction (black dot), densified and smoothed points (gray dot), and breakpoints (red circle).



(a)

(b)

Figure 4.14. Breakpoint detection for building boundary (a) 1D (U-direction and V-direction) profiles of building boundary, (b) 2D building boundary, black: LiDAR boundary points, gray: densified and smoothed points, red: breakpoints

4.4.3 Boundary Regularization Using Detected Breakpoints

A boundary regularization using the detected breakpoints is proposed in this sub-section. First, the straight-line and curved segments of a building primitive are identified using the detected breakpoints. Second, the segments identified from wrongly extracted breakpoints are recognized and corrected. Third, the vertices of the breakpoint detection (BD) boundary are generated through an intersection of neighboring segments.

After recognizing the breakpoints, segments are extracted by tracking the successive densified and smoothed points which are not deemed to be breakpoints. Then, the corresponding LiDAR boundary points to each segment are identified through assigning each LiDAR boundary point to the closest segment. For a segment, two model-fitting procedures (i.e., straight-line model and arc model) are applied to the corresponding LiDAR boundary points to derive both straight-line/arc parameters together with the respective a-posteriori variance factor. A segment is determined as a curved segment instead of straight-line segment when the former's a-posteriori variance factor is much smaller than the straight-line-based one. Incorrected breakpoints could be extracted because of highly irregular boundary points, which is shown in Figure 4.15. Therefore, neighboring straight-line segments are checked to ensure that they do not belong to a single straight line. When neighboring straight-line segments are deemed to belong to the same line, those two segments are combined into one straight-line segment. Two criteria are used to merge the neighboring straight-line segments into one segment. First, neighboring straight-line segments are close to being parallel. Second, the a-posterior variance factors of fitted merged straight-line segment is close to the a-posterior variance factors before merging.

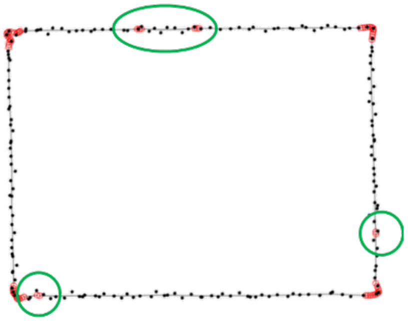


Figure 4.15. Wrongly detected breakpoints

After identifying the segments of a building primitive, vertices of a BD boundary are generated through the intersection of the neighboring segments, either a vertex between two straight-line segments or a vertex between a straight-line segment to a curved segment. Three situations could happen in the intersection procedure. First, an intersection of two neighboring segments (namely, *extracted segments*) is close to the breakpoints between the segments (i.e., distance < $Th_{vertices\text{-}dist}$). Then, this intersection is regarded as a correct vertex, as shown in Figure 4.16.

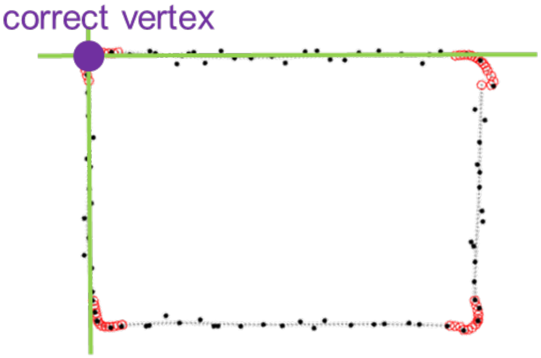
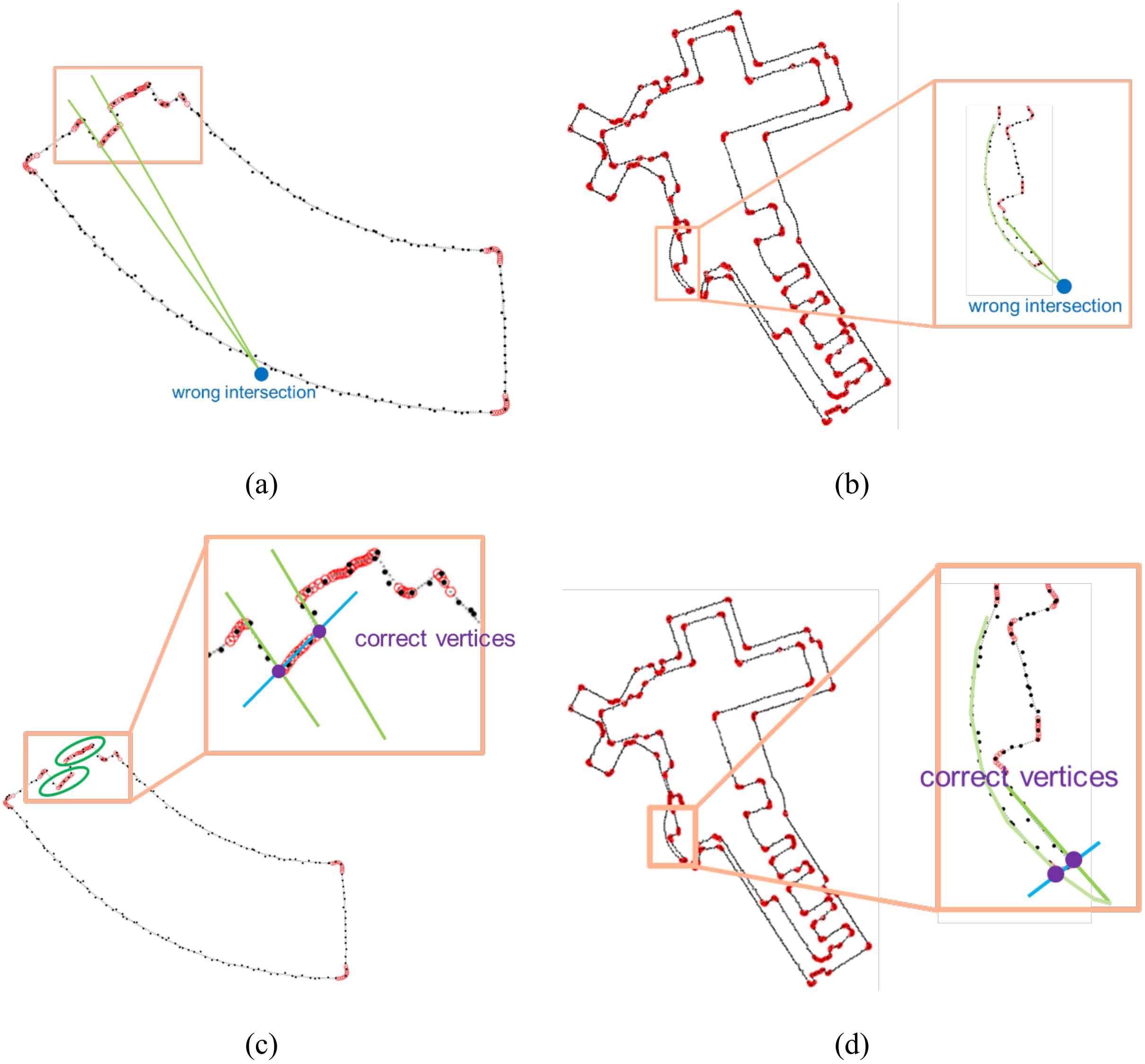


Figure 4.16. Correct vertex derived from an intersection of neighboring segments

Second, an intersection of two neighboring segments (*extracted segments*) is far from the breakpoints between the segments (i.e., distance > $Th_{vertices\text{-}dist}$). The intersection is regarded as a wrong vertex. This situation could happen when the segments are not really neighboring segments. More specifically, a straight-line segment is missing between those two segments because the missing segment is identified as breakpoints. Figure 4.17 (a-b) shows the wrong vertices derived from the intersection of two straight-line segments and the intersection of a straight-line segment to a curved segment. In this situation, a straight-line segment is fitted through the breakpoints (namely, *breakpoint segment*) between the two neighboring segments in question. Then, the vertices are derived from the extracted segment and breakpoint segment, as described in Figure 4.17 (c-d).



(a) (b)

(c) (d)

Figure 4.17. Wrong vertex derived from the intersection of neighboring segments and the derivation of correct vertices: (a, c) a straight-line segment to a straight-line segment, (b, d) a straight-line segment to an curved segment

Third, both the intersection of two neighboring *extracted segments* as well as the intersection of *breakpoint segment* and *extracted segment* are far from the breakpoints (i.e., distance > $Th_{vertices\text{-}dist}$), as shown in Figure 4.18. This situation could happen when the densified points are over-smoothed and breakpoints are not detected correctly. In this case, the middle breakpoint between two neighboring *extracted segments* is used as a vertex. One should note that the derived vertex in this case could be incorrect. Therefore, an integration of the outcome of the BD and ERMBR strategies are introduced in next section.

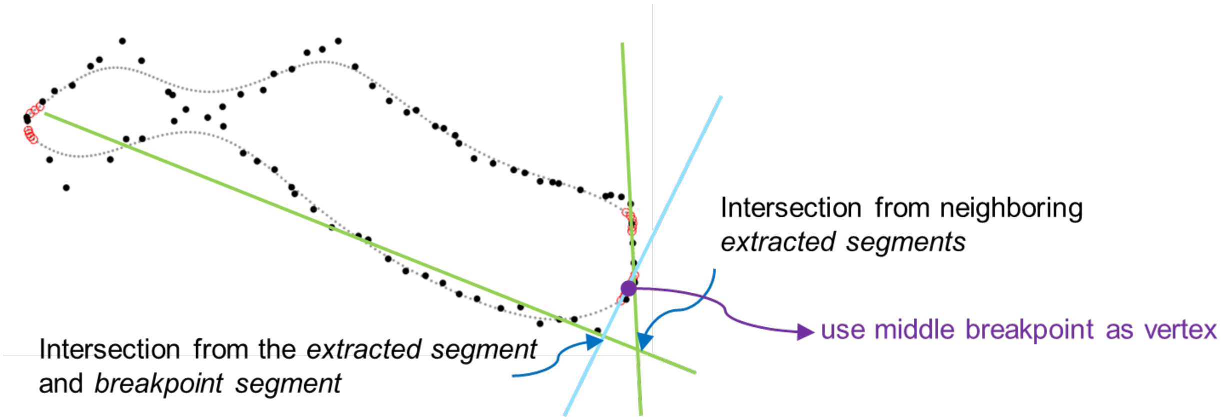
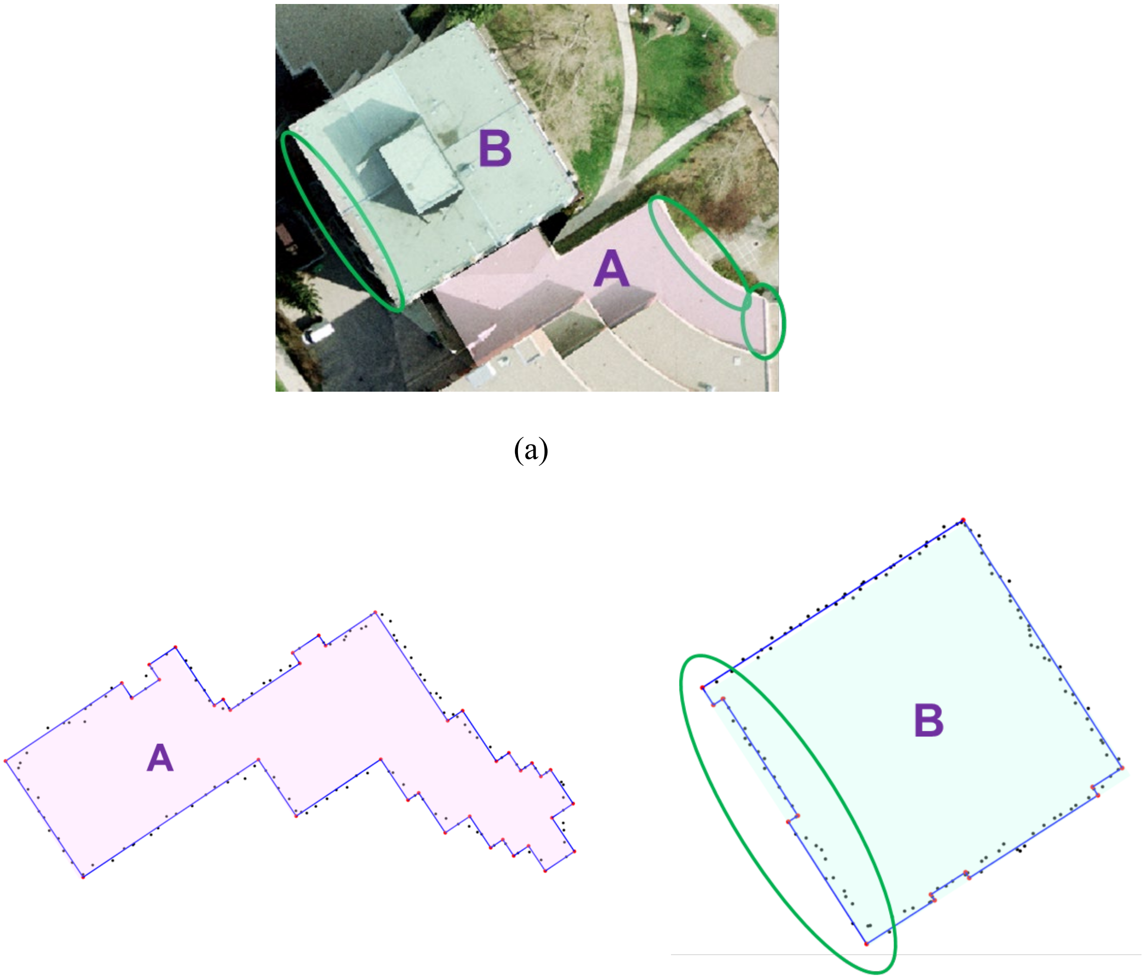

Intersection from neighboring
extracted segments
use middle breakpoint as vertex
Intersection from the *extracted segment*
and *breakpoint segment*

Figure 4.18. Wrong vertex from both *extracted segments* and *breakpoint segment*

4.5 Integration of the Outcome of the Regularization Strategies

Both ERMBR and BD approaches have their strengths and weaknesses. ERMBR strategy can produce reliable outcome for single orientation and right-angle buildings, but it cannot reconstruct buildings with multi-orientation and curved boundaries. Moreover, since the stopping criteria for reconstructing the next layer MBR is the size of the area, when boundary points are highly irregular, the regularized boundaries could be partially incorrect. Examples of incorrectly regularized boundaries generated from ERMBR strategy are shown in Figure 4.19. On the other hand, the BD strategy can help in regularizing building primitives with multi-orientation or curved boundaries. However, the densified points could be over-smoothed in BD approach, so breakpoints cannot be identified correctly and regularized boundaries cannot be generated correctly. Furthermore, this approach cannot maintain right-angle in the regularization process. Examples of the disadvantages of BD approach are depicted in Figure 4.20. Therefore, this section addresses a strategy to utilize the strengths of those two regularization approaches and derive an integration boundary from BD boundary and ERMBR boundary.

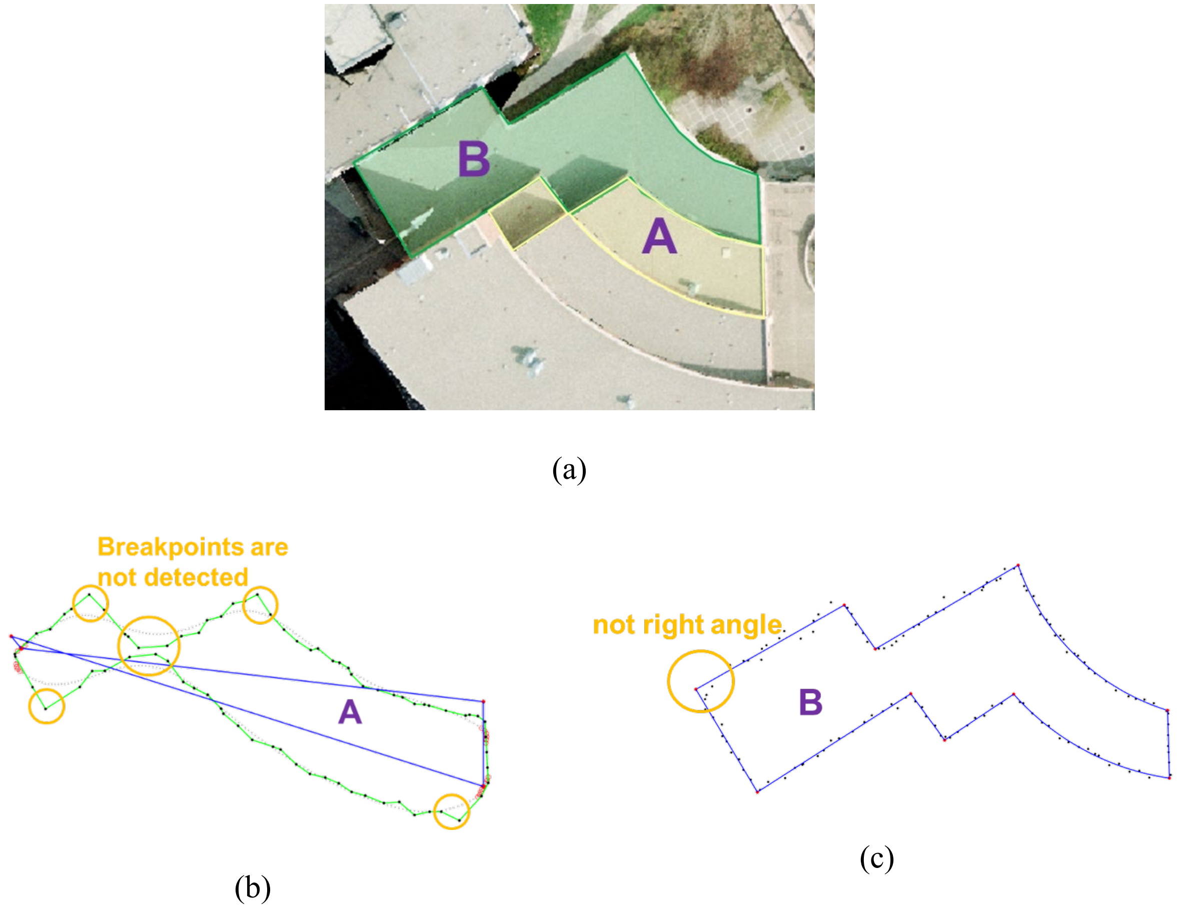


(a)

(b)

(c)

Figure 4.19. Example of the disadvantage of ERMBR approach, (a) orthophoto, (b) primitive with multi-orientation and curved boundary, (c) primitive with highly irregular boundary points, red dot: vertices of ERMBR boundaries, black dot: LiDAR boundary points, blue line: regularized boundaries



(a)

(b)

(c)

Figure 4.20. Example of the disadvantage of BD approach, (a) orthophoto, (b) incorrectly regularized boundary, (c) regularized boundary without right-angle corners, red dot: vertices of BD boundaries, black dot: LiDAR boundary points, grey dot: densified and smoothed points, red circle: breakpoints, blue line: regularized boundaries, green line: LiDAR boundary

To integrate the outcome from ERMBR and BD approaches, every segment of a BD boundary is classified as a trustable segment or a non-trustable segment based on the a-posterior variance factor of the fitted segment. The threshold for the segment classification is based on LPS of each primitive. Then, corresponding segment pairs between ERMBR boundaries and BD boundaries are identified. To identify corresponding segment pairs, two endpoints of each segment from a BD boundary can correspond to two closet vertices of an ERMBR boundary. Segments within two vertices of an ERMBR boundary are regarded as correspondences to the segment from a BD boundary in question. Every segment of a BD boundary corresponds to one/many segments of an ERMBR boundary. Three situations are considered in the integration of a corresponding segment pair.

For the first situation, a trustable segment of a BD boundary corresponds to one/many segments of an ERMBR boundary and the orientation of a trustable segment is close to the orientation of ERMBR (i.e., angle difference < $Th_{segment\text{-}ori}$). The ERMBR orientation is regarded as a reliable orientation since it is derived from the whole boundary instead of a segment. The trustable segment is fitted to a straight line which aligns with the ERMBR orientation through the corresponding LiDAR boundary points. The fitting procedure has been mentioned in Section 4.3.2.2. When a segment pair is one segment to one segment, which is depicted as PairA in Figure 4.21, the segment from the ERMBR boundary can be adopted without the fitting process. However, when the segment pair is one segment to many segment, the fitting process is applied to adjust the trustable segment. One should notice that this step can avoid the artificial segments of ERMBR boundary (i.e., segments are introduced by noise of point clouds) and also allow for regularized primitive with right-angle corners, as shown in Figure 4.22.

For the second situation, a trustable segment of a BD boundary corresponds to one/many segments of an ERMBR boundary and the orientation of a trustable segment is different from the orientation of ERMBR (i.e., angle difference > $Th_{segment\text{-}ori}$). The trustable segment is adopted in the integrated boundary. This step can allow for regularized primitives with multi-orientation. The illustration of the second situation is shown as PairB in Figure 4.21. For the third situation, a non-trustable segment of a BD boundary corresponds to one/many segments of an ERMBR boundary. The corresponding segments from an ERMBR boundary are used, as described as PairC in Figure 4.21. In this way, the incorrect segments from ERMBR boundaries can be avoided. After integrated the outcome from BD and ERMBR boundaries, the vertices of the integrated boundary are updated through the intersection of neighboring segments.

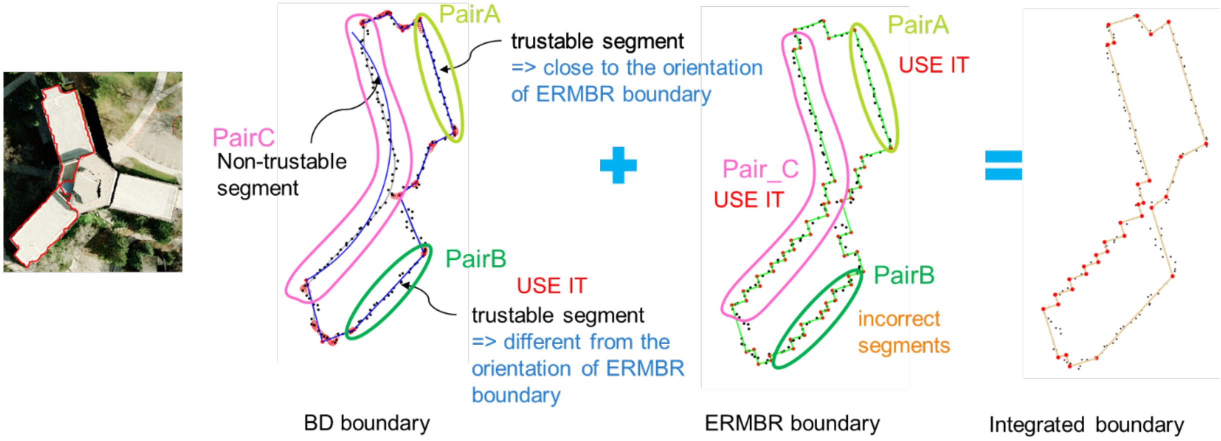


Figure 4.21. Illustration of integration of the outcome of the BD and ERMBR strategies

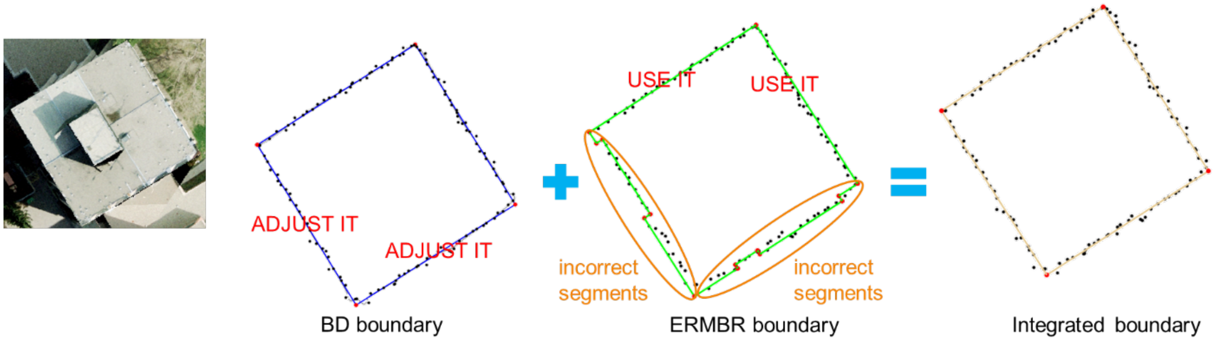


Figure 4.22. Illustration of integration strategy that can avoid incorrect segments from ERMBR boundaries and derive the boundaries with right-angle corners

4.6 Water-tight DBM Generation

Neighboring building primitives, which share a common vertical wall in the real world, should have seamless models which have no gaps between them. However, the regularized boundary for each building primitive is derived separately and no topological relationship is established or considered between the boundaries, so gaps might exist between adjacent boundaries. In this section, a water-tight DBM generation approach is proposed for the alignment and merging of the integrated boundaries for neighboring building primitives. First, we need to identify adjacent segments from neighboring primitives. Second, neighboring primitives with all-straight line segments and similar orientation are adjusted to the same orientation. Finally, adjacent segments shared amongst the neighboring primitives are merged together depending on whether they are 1) parallel straight-line segment (Figure 4.23 - a), 2) non-parallel straight-line segment (Figure 4.23 - b), 3) curved segment to straight-line segment (Figure 4.23 – c), or 4) curved segment to curved segment (Figure 4.23 - d). The above steps for water-tight DBM generation are discussed in detail in the forthcoming sections.

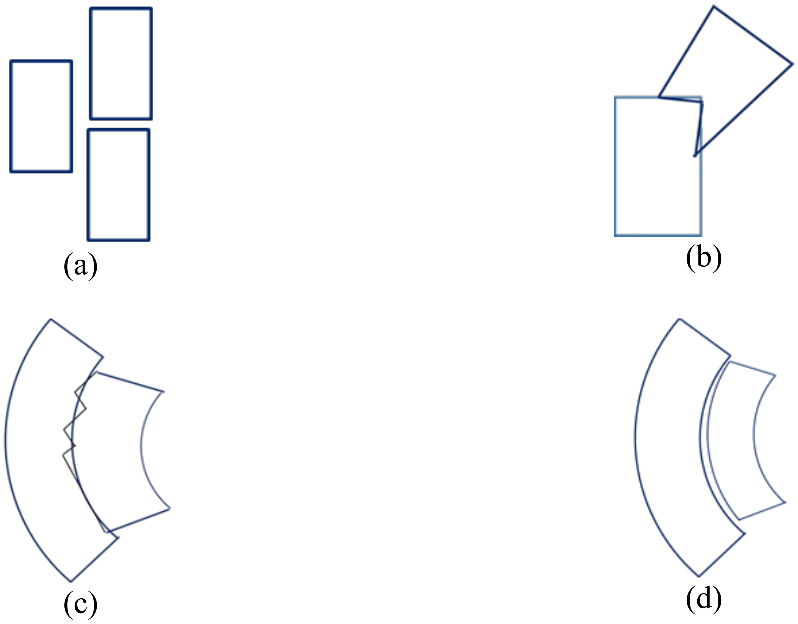


Figure 4.23. Illustration of different types of edge merging (a) parallel straight-line segment merge, (b) non-parallel straight-line segment merge, and (c) curved segment to straight-line segment merge, and (d) curved segment to curved segment merge

4.6.1 Identification of Adjacent Segments from Neighboring Primitives

The first step in the generation of water-tight building models is to identify adjacent segments among the different building primitives. This is done using the original LiDAR boundary points and LPS of their respective primitives to identify neighboring primitives. Adjacent segments of regularized integrated boundaries between neighboring primitives are identified using the corresponding LiDAR boundary points and their LPS. Every LiDAR boundary point would belong to the closest segment. Neighboring primitives in two dimension, XY-direction, of a LiDAR boundary in question are identified by defining a search radius for every LiDAR boundary point based on the LPS. The boundary points within the search radius can represent the closest segments of neighboring primitives. Therefore, the adjacent segments of regularized boundary among the neighboring primitives can be derived. An example of identification of adjacent segments from neighboring primitives is shown in Figure 4.24. Figure 4.24 shows two neighboring primitives with their LiDAR boundary points. The neighboring primitive and the adjacent segment can be identified using a search radius of a point in question. In this case, primitives A and B are deemed to be neighboring to each other. Moreover, segment 2 of primitive A and segment 4 of primitive B are regarded as being adjacent to each other.

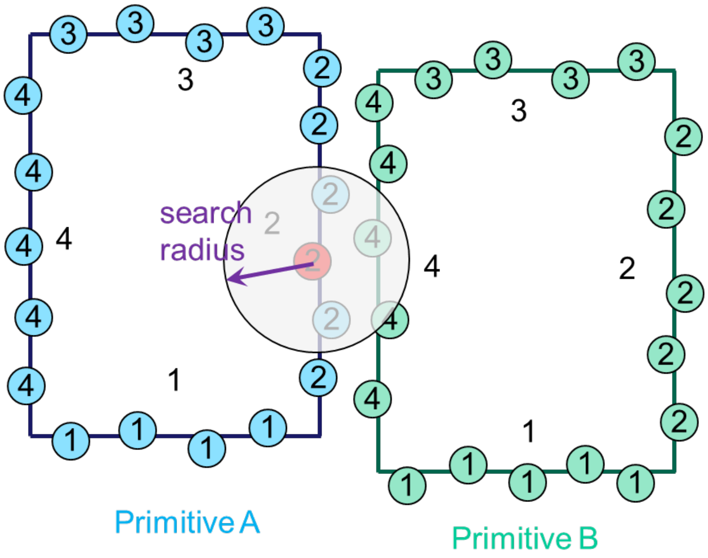
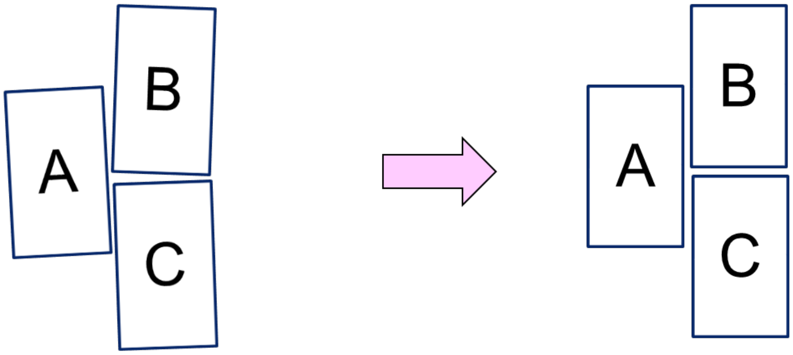


Figure 4.24. Illustration of identification of adjacent segments from neighboring primitives

4.6.2 Alignment for Primitive with All Straight-line Segments

Before merging parallel straight-line segment, the entire cluster of similarly oriented neighboring primitives with all straight-line segments needs to be commonly oriented to align the primitives in the same direction. This is required to maintain the right-angled nature of the primitives after segment merging. When the orientation difference between neighboring primitives with all straight-line segments is smaller than a threshold ($Th_{primitive\text{-}ori}$), they are grouped into the same cluster. The modified orientation for the cluster is derived by computing the weighted average of the individual primitive orientations. The weights are determined based on the area of the primitives, with a higher weight assigned to the primitives with larger area, as the orientation of a primitive with larger area is considered to be more reliable. An illustration of primitive alignment is shown in Figure 4.25.



Cluster Orientation
(weighted average of the individual primitive orientation):

$$\frac{Ori_A \times Area_A + Ori_B \times Area_B + Ori_C \times Area_C}{Area_A + Area_B + Area_C}$$

Figure 4.25. Illustration of primitive alignment for primitives with all-straight line segments

4.6.3 Segment Merging

After adjusting the orientation of primitives with all straight-line segments and identifying adjacent segments, the adjacent segments are merged and adjusted depending on their types. For adjacent parallel straight-line segments, the LiDAR boundary points corresponding to adjacent parallel segments are fitted a new straight line in the same orientation for these segments. Then, the vertices of regularized merged primitives are updated through the intersection of neighboring segments. Figure 4.26(a) and (b) show neighboring primitives before and after segment merging, respectively. For adjacent non-parallel straight-line segment pair, as depicted in Figure 4.27 (a), the longer segment would be regarded as the reliable one. So, its position and orientation are fixed during the merging process and it is merged with the other segment by modifying the latter. This is done by directly projecting the short segment onto the longer one and then, the vertices of regularized merged primitives are updated through the intersection of neighboring segments, as shown in Figure 4.27 (b). For adjacent straight-line to curved line segment pair, the process is similar to the non-parallel straight-line segment merging. The longer segment is regarded as the reliable one and the short segment is projected onto the longer one. The projection is done by assigning the parameters (i.e., arc parameters or straight-line parameters) of the longer segment to the shorter segment. In the Figure 4.28 (a), there are totally six adjacent segment pairs. Hence, the projection would be applied six times. Those six segments from primitive B would be assigned the same arc parameters of the adjacent segment from primitive A. After the projection, the vertices of regularized merged primitives are updated through the intersection of neighboring segments. In the vertices update procedure, when the neighboring segments have the same parameters (i.e., arc parameters or straight-line parameters), the intersection would not be apply, as shown in Figure 4.28 (b). For adjacent curved segment pair, the LiDAR boundary points corresponding to adjacent curved segments are fitted a new curved line, as shown in Figure 4.29. After merging the adjacent segments, vertices of regularized merged primitives are updated through the intersection of neighboring segments.

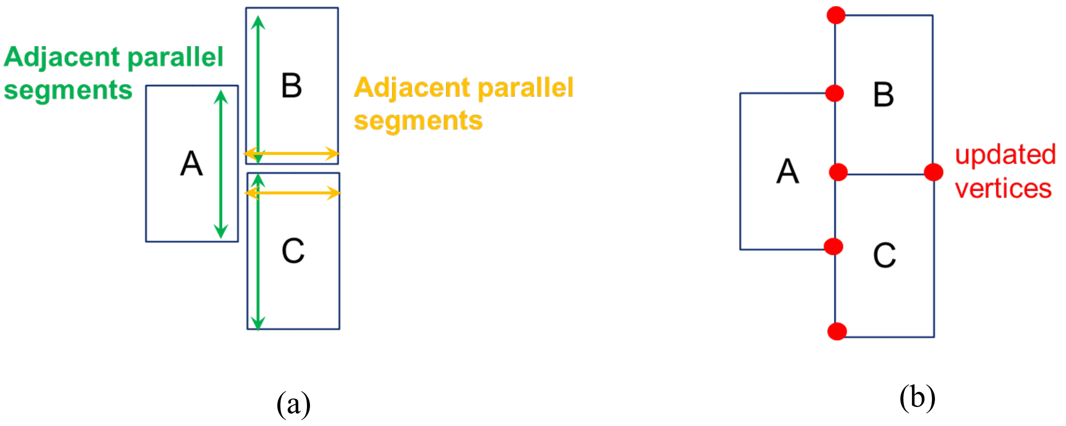
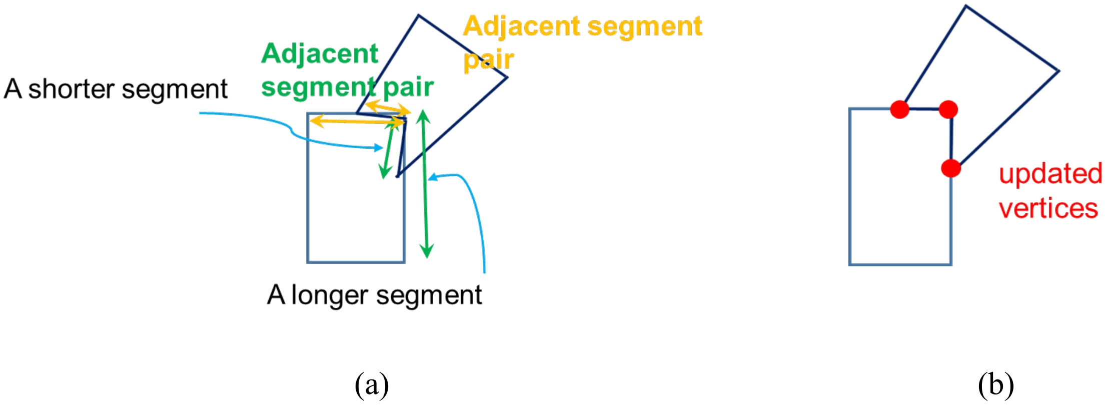


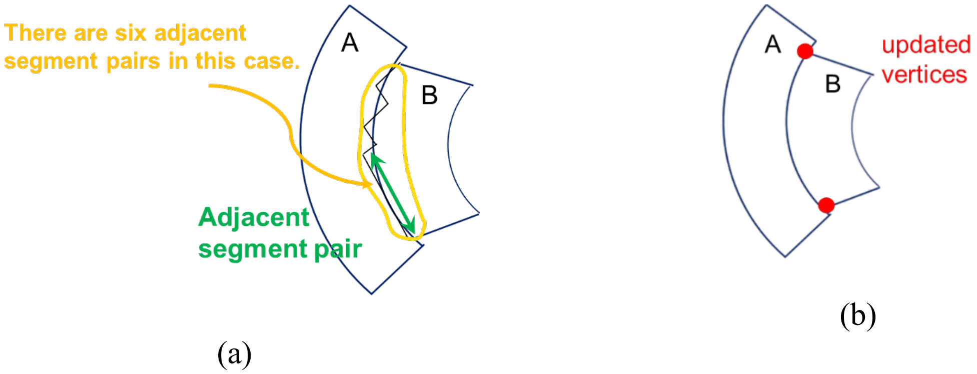
Figure 4.26. Illustration of (a) before, and (b) after parallel straight-line segment merging



(a)

(b)

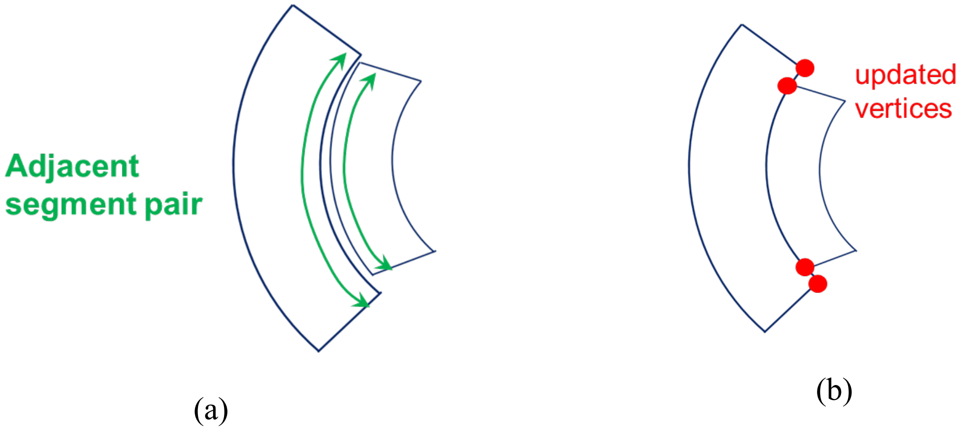
Figure 4.27. Illustration of (a) before, and (b) after non-parallel straight-line segment merging



(a)

(b)

Figure 4.28. Illustration of (a) before, and (b) after straight-line and curved segment merging



(a)

(b)

Figure 4.29. Illustration of (a) before, and (b) after curved segment merging

4.7 Experimental Results

To illustrate the performance and the feasibility of the proposed methodology, this section provides experimental results from real airborne LiDAR dataset. This dataset is captured by an Optech 3100 from two different flying heights, 1 km and 1.4 km. The target area is a semi-urban area consisting of buildings with right-angled, multi-orientation, and curved boundary. The extent of the covered area is 2 km x 1.4 km and the average point spacing of the whole dataset is approximately 0.62m. Figure 4.30 shows a perspective view of the point cloud data which is colored based on height. The following sections provides the building identification and boundary extraction results, boundary regularization results, and water-tight DBM results. Before discussing the experimental results, different thresholds, the rationale for setting them up, and the utilized numerical values are introduced first and shown in Table 4.1.

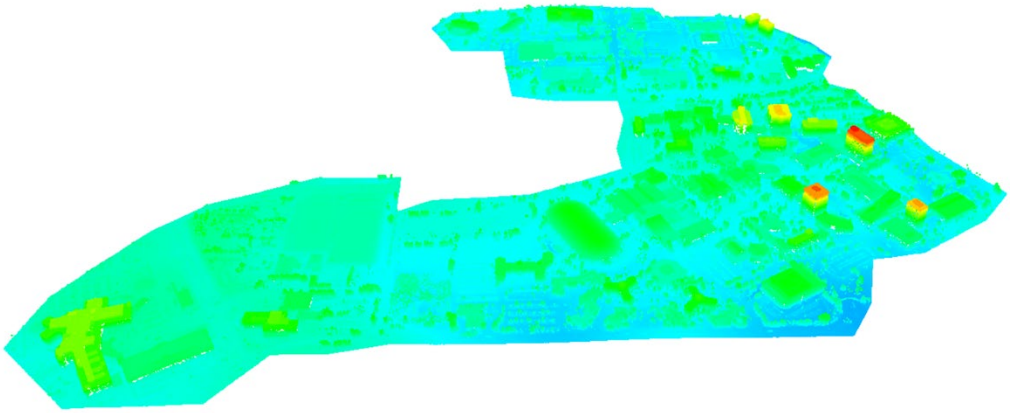


Figure 4.30. Perspective view of the LiDAR point cloud colored by height

Table 4.1. Pre-defined thresholds used for DBM generation

Name of Threshold	Value	Rationale
Building Hypotheses Identification (Section 4.2)		
Area of segments	30 m^2	A value represents the minimum size of primitive in that area.
Height difference between segments and ground segments	1.5 m	A value represents the lowest building in that area.
Spike Removal (Section 4.2)		
Size of included angle to define potential spike (Th_{angle})	80°	The potential spike should be large to ensure that all the spikes can be identified.
The difference between the potential spike angle and its neighboring angles (Th_{nei})	40°	The difference should be large while avoiding the risk of missing any spike.
ERMBR Procedure (Section 4.3)		
Area threshold to incorporate the sub-level MBR (Th_{ar})	5 m^2	The area size should be set-up in a way to ensure that the area is small enough to generate the details of building models while avoiding the risk of producing the artifact details.
Boundary Regularization Using Breakpoint Detection Approach (Section 4.4)		
Distance threshold to define correct vertex ($Th_{vertices\text{-}dist}$)	2 m	The discrete boundaries may not include the corners of buildings. A correct vertex could have a considerable distance from boundaries. Therefore, $Th_{vertices\text{-}dist}$ should be large enough to ensure that a correct vertex can be identified while avoiding the risk of mis-judgement the wrong vertex as correct vertex.
Integration of the Outcome of the Regularization Strategies (Section 4.5)		
Orientation threshold to define whether segments have the similar orientation with ERMBR boundary ($Th_{segment\text{-}ori}$)	6°	The value should be small to avoid modifying segments into wrong orientation.
Water-tight DBM Generation (Section 4.6)		
Orientation threshold to define whether primitives should be clustered ($Th_{ptimitve\text{-}ori}$)	6°	The value should be small to avoid modifying primitives into wrong orientation.

4.7.1 Building Identification and Boundary Extraction Results

The data processing starts by applying a multi-class simultaneous segmentation to partition the point cloud into planar, pole-like, and rough segments, which are shown in Figure 4.31 (a). Since the purpose of this thesis is to reconstruct buildings with planar rooftops, it only focuses on the planar segments. The segmented result could have over-segmentation and invasion issues as discussed in Section 3.5; therefore, a QC process is applied to the planar segments. Figure 4.31 (b) shows the planar segments after the QC procedure. Then, a ground and non-ground classification technique is used to identify above-ground objects. Figure 4.31 (c) shows non-ground segments in red and ground ones in green. After extracting non-ground objects, two criteria listed in Table 4.1 are used to identify the building hypotheses, as shown in Figure 4.31 (d). Finally, the boundary points of the generated building hypotheses are extracted in order to reconstruct the building models, as shown in Figure 4.31 (e).

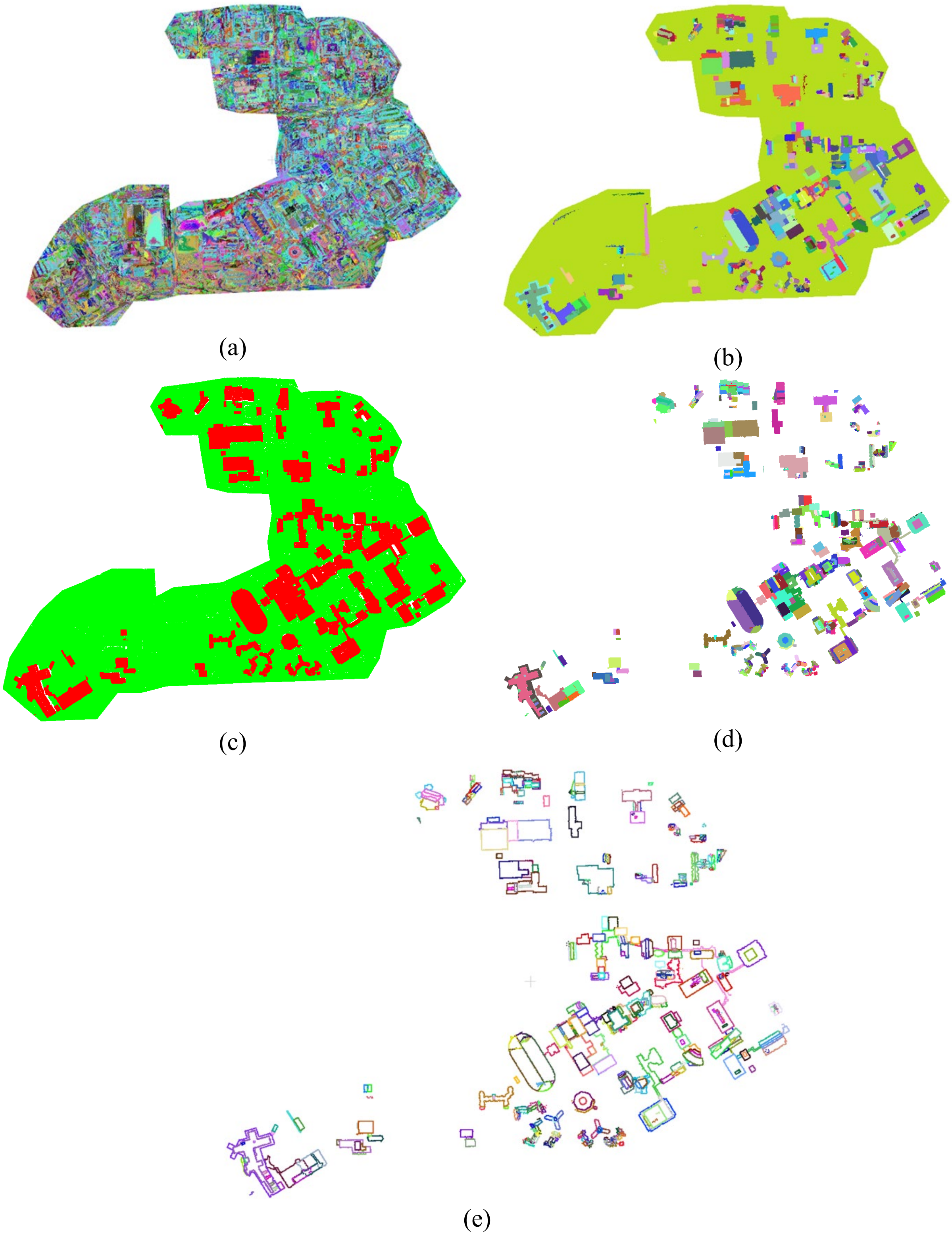
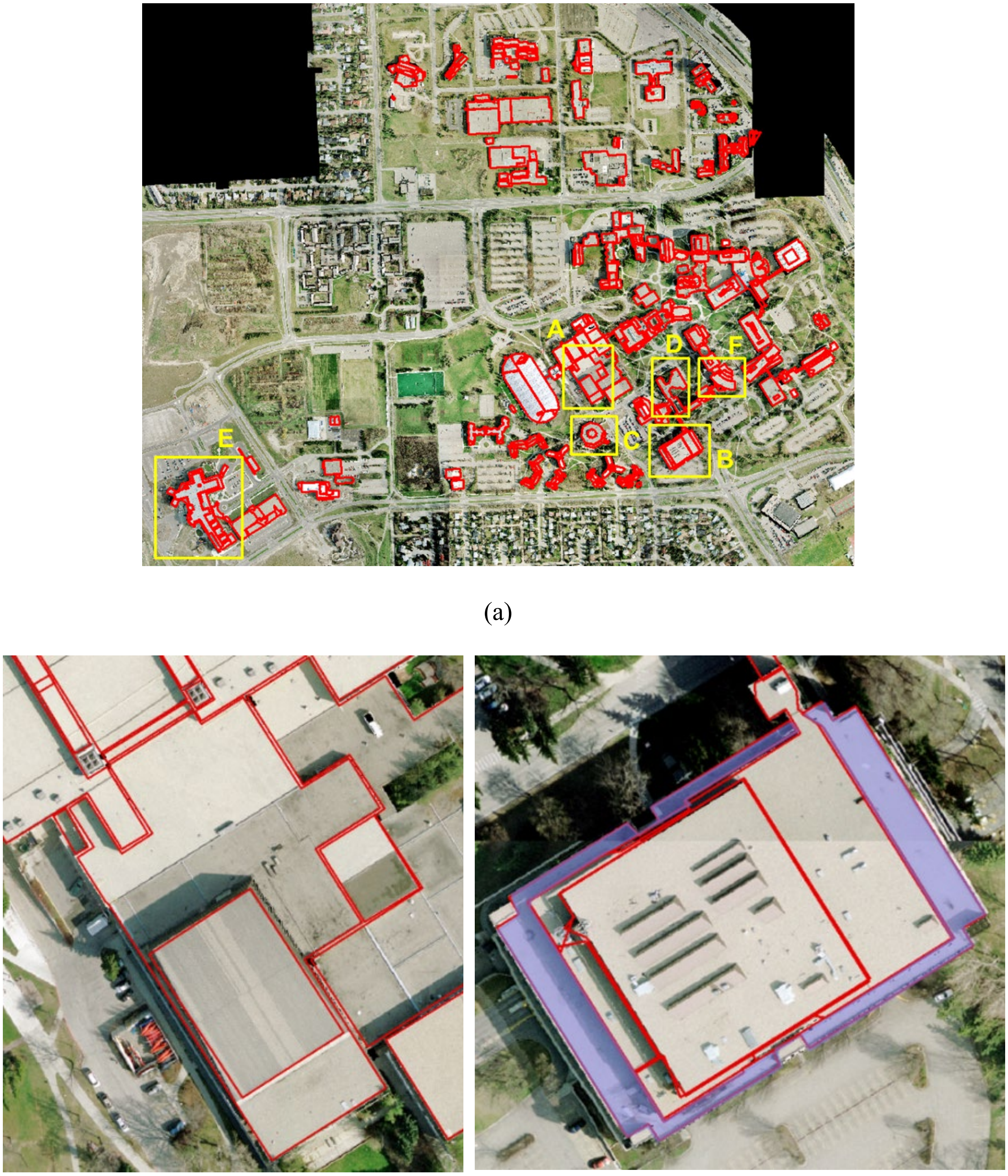


Figure 4.31. (a) Segmentation results, (b) planar segments after QC procedure, (c) ground/non-ground classification, (d) building hypothesis, (e) building boundary

4.7.2 Boundary Regularization Results

This section provides the boundary regularization results from the integration of the outcome of the ERMBR and DB strategies. The integrated models are shown on top of the orthophoto in Figure 4.32 (a). Some complex areas are marked in Figure 4.32 (a) and closer inspections are shown in Figure 4.32 (b-g). Figure 4.32 (b) shows that right-angle buildings were reconstructed correctly. Figure 4.32 (c) shows a primitive colored in purple with overlapping RMBR which could not be reconstructed correctly using the strategy proposed by Kwak and Habib [7] but was reconstructed correctly using the strategy proposed in this thesis. Figure 4.32 (d) shows building primitives with multi-orientation that could be reconstructed correctly. However, the building model for the inner structure colored in yellow mainly came from ERMBR procedure because its corners could not be identified in the BD strategy. Figure 4.32 (e & f) shows that multi-orientation and right-angle buildings were reconstructed correctly. Figure 4.32 (g) shows buildings with right-angle, multi-orientation, and curved boundaries. For the primitive colored in green, its corners could not be identified in the BD strategy. Therefore, the segments of this model came from the ERMBR approach.



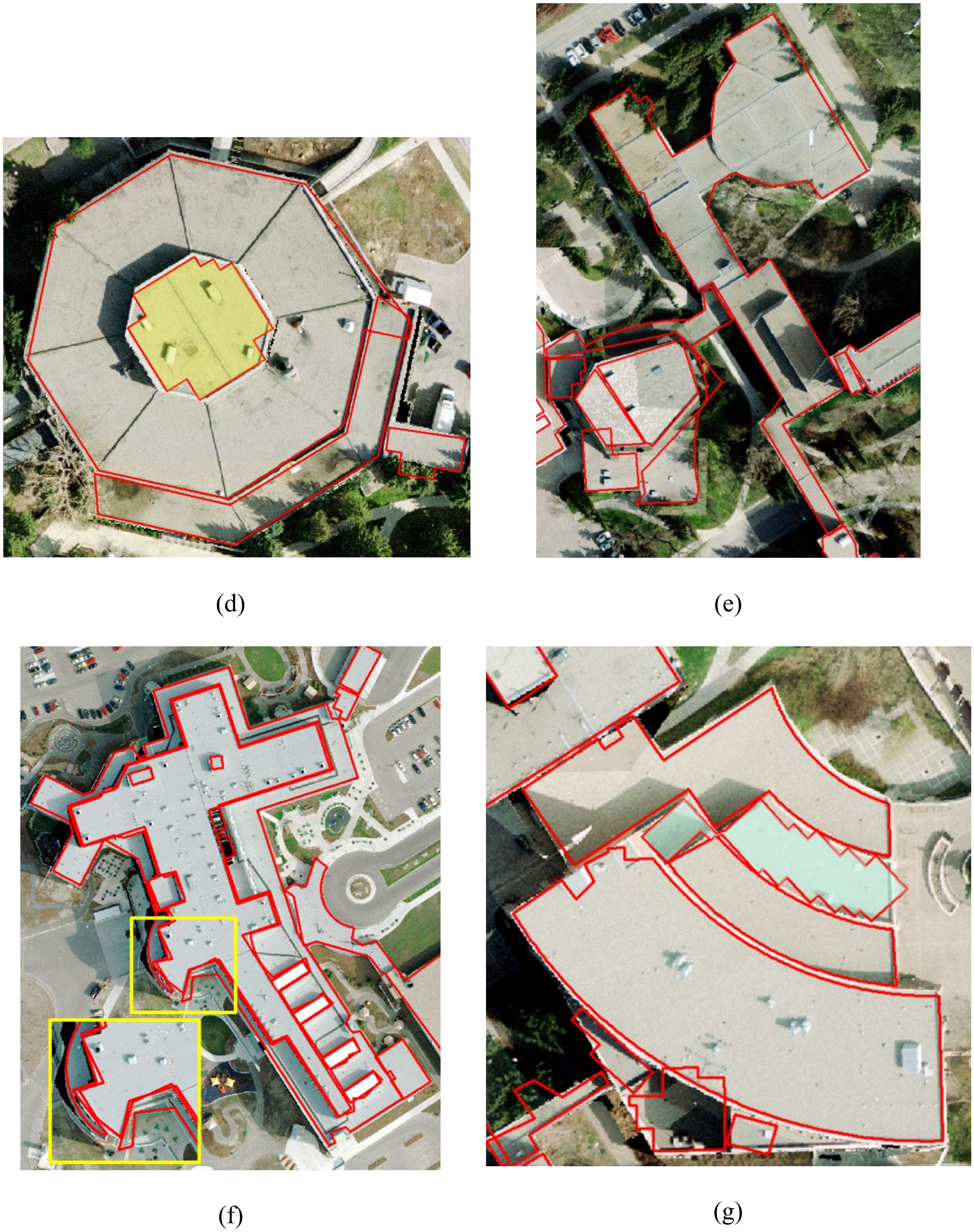
(a)

(b)

(c)

Figure 4.32. Integrated models derived from the integration of the outcome of ERMBR and BD approaches (a) overview of whole area, (b)-(g) zoom-in areas

Figure 4.32. Continued



(d)

(e)

(f)

(g)

In order to evaluate the performance of the proposed approach, the integrated models need to be quantitatively evaluated against a reference dataset that is highly accurate. The reference dataset for comparison was generated by manually measuring corners in the corresponding true orthophoto [92] that has a GSD of 20 cm. The true orthophoto was produced using a set of aerial imagery scanned for a 6 cm GSD and the reconstructed building models [93]. The test area used for this study consists of 447 primitives. The planimetric coordinates for 1004 corners were used to perform the statistical analysis to validate the accuracy of the proposed approach. The location of the manual measurements is shown in Figure 4.33. Some areas with small primitives were not considered in the validation procedure because the extracted LiDAR boundaries of those small primitives were incorrect and the regularized boundaries would be also incorrect, as shown in Figure 4.34. There were two reasons to produce the incorrect boundaries. First, trees around the buildings could affect the extraction of segments. Then, derived LiDAR building boundaries could include points on trees. Moreover, the point spacing of this datasets was relatively large to derive high-quality small roof patches.

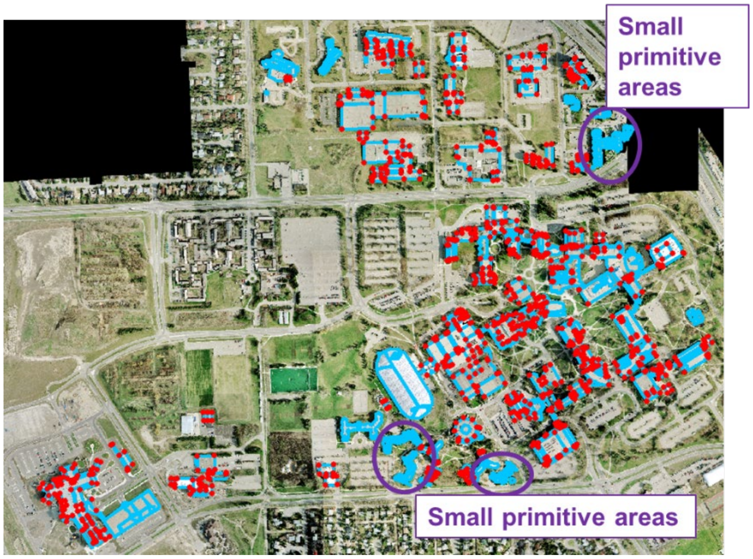
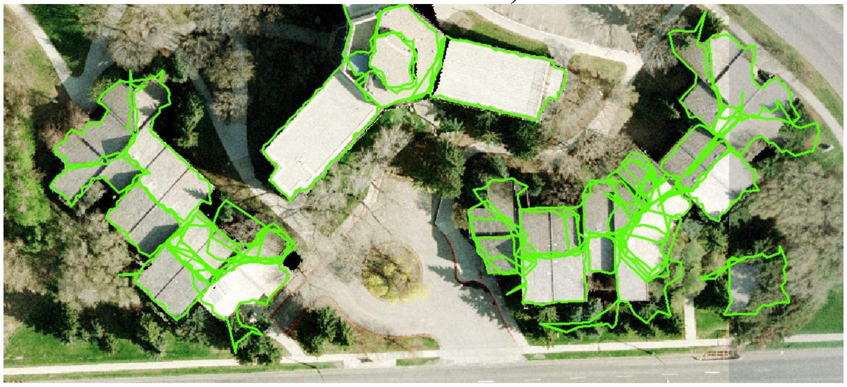
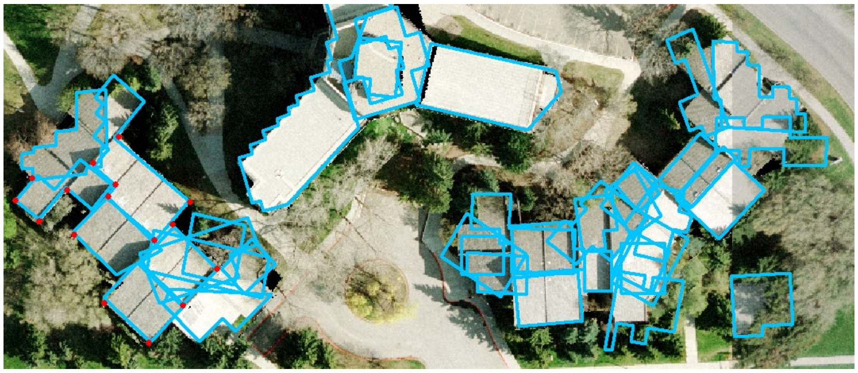


Figure 4.33. Reference data for validation (blue line: integrated models, red points: manual measurements)



(a)



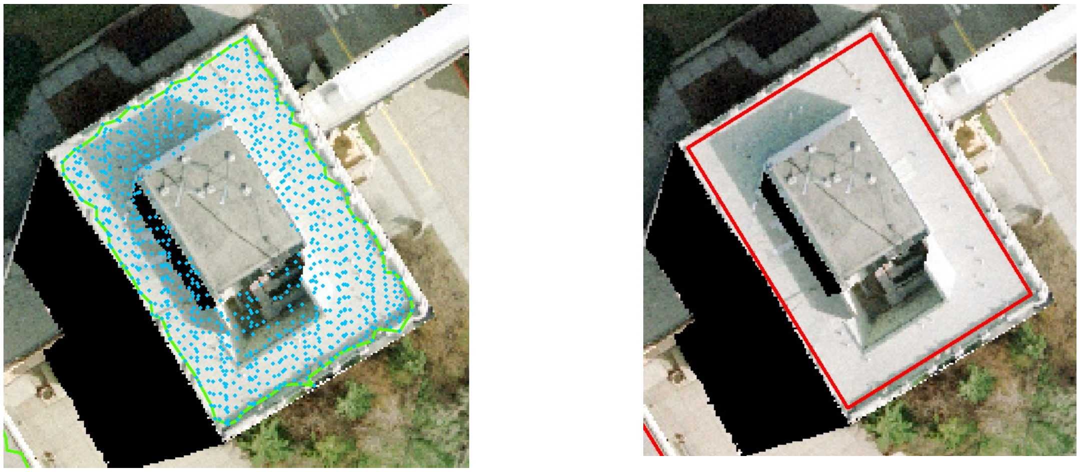
(b)

Figure 4.34 Incorrect results that were not included in the validation procedure (a) LiDAR boundaries, (b) integrated models

Table 4.2. Statistical evaluation of the deviation between the reference dataset and the integrated models

	X (m)	Y(m)
Absolute min	0.000	0.000
Absolute max	2.347	1.871
Absolute mean	0.364	0.399
Absolute standard deviation	0.294	0.343
Mean	-0.076	0.061
Standard deviation	0.462	0.521
RMSE	0.468	0.524

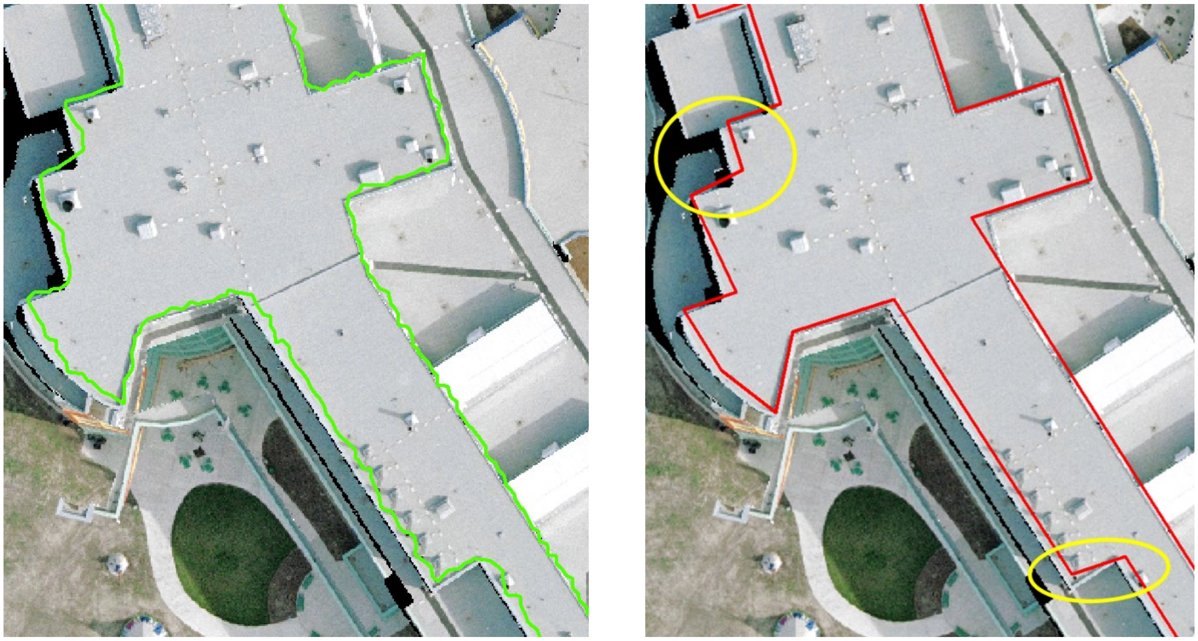
Table 4.2 shows the statistical evaluation of the deviation between the reference dataset and the integrated models. As tabulated, the RMSE in X-direction is 46.8 cm and in Y-direction is 52.4 cm. Apart from the accuracy of the boundary regularization using the proposed approach, there were several other sources of error which were inevitable in a test area. For instance, an extracted planar rooftop segment would not include the points on the parapet (a low protective wall along the edge of a roof) as the latter would have heights different from the height of points belonging to the rooftop. Figure 4.35 (a) shows a rooftop with parapet with the corresponding segmented points in blue and the LiDAR boundary in green (the segmented points did not include those representing the parapet). Therefore, the generated building models may deviate from the parapet, Figure 4.35 (b). Moreover, when an edge of a building boundary was short, i.e., as highlighted by the yellow circle in Figure 4.36 (b), the extracted boundary points may not be accurate enough to define the edge. So, the reconstructed segments in the regularized model could be inaccurate.



(a)

(b)

Figure 4.35. Building with a parapet: (a) segmented points (blue) and LiDAR boundary (green), and (b) building model in red



(a)

(b)

Figure 4.36. Building with short segments: (a) LiDAR boundary (green), and (b) building model in red

4.7.3 Water-tight DBM Results

From the results shown in Figure 4.32, it is obvious that there are gaps between the models associated with contiguous building primitives. In order to provide more realistic building models, the water-tight building model strategy is adopted to merge both the straight-line and curved segments. Figure 4.37 shows the comparison between the building primitives before and after applying the water-tight DBM generation strategy. Figure 4.37 (a) and (b) show an example of parallel straight-line segment merging, from which it is evident that right-angled buildings can be aligned and merged appropriately. The area in Figure 4.37 (c) and (d) mainly depicts curved segment to curved segment and straight-line segment merging. One should note that some primitives may not be modeled correctly, as highlighted by the yellow circle in Figure 4.37 (c), but the shape of models can be improved in the water-tight DBM generation strategy. Figure 4.37 (e) and (f) depict a complex building with both parallel and non-parallel straight-line segment merging. The reconstructed buildings can be displaced as 3D building models in ArcGIS software. Some samples of 3D building models corresponding to the areas in Figure 4.37 are shown in Figure 4.38.

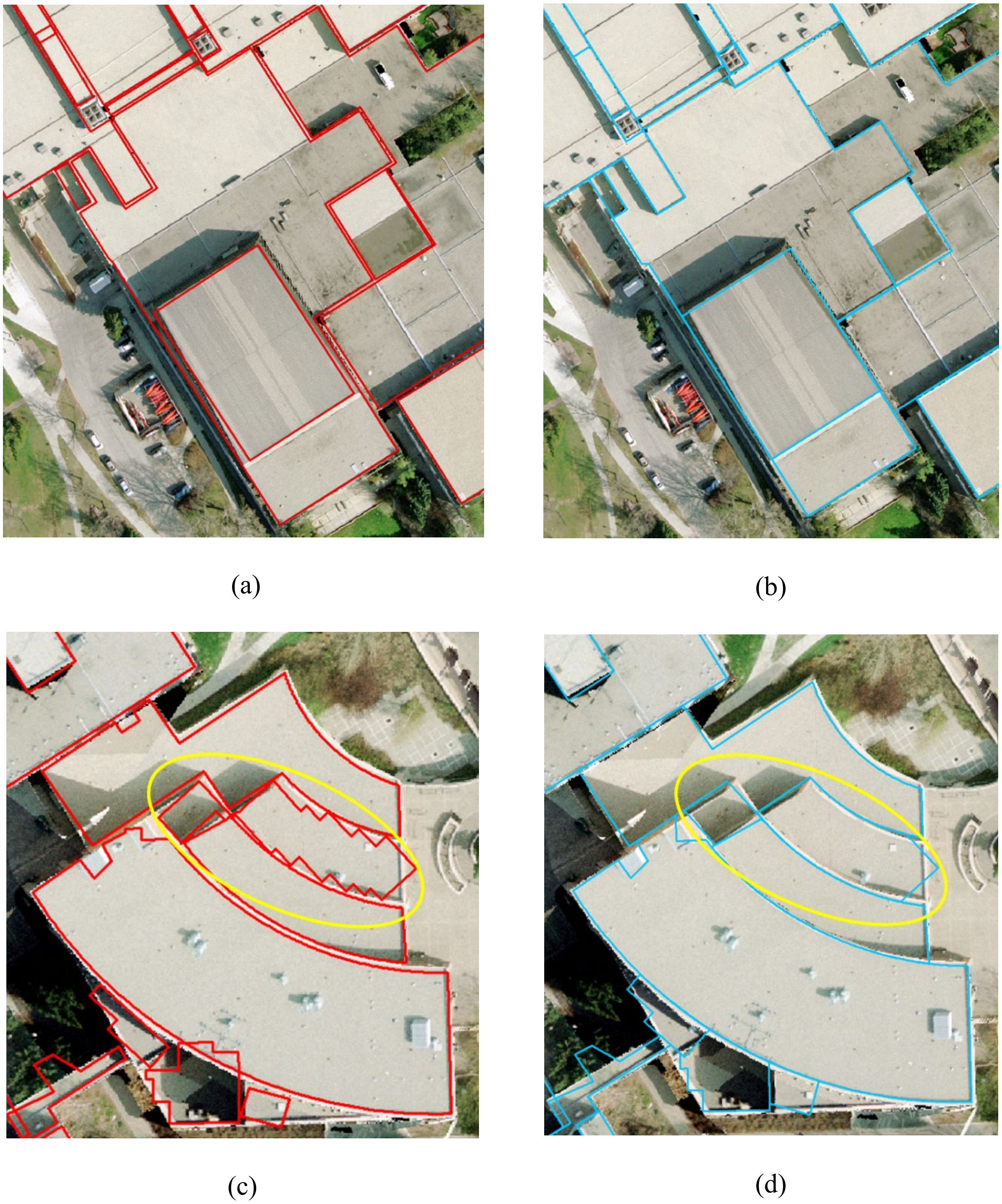
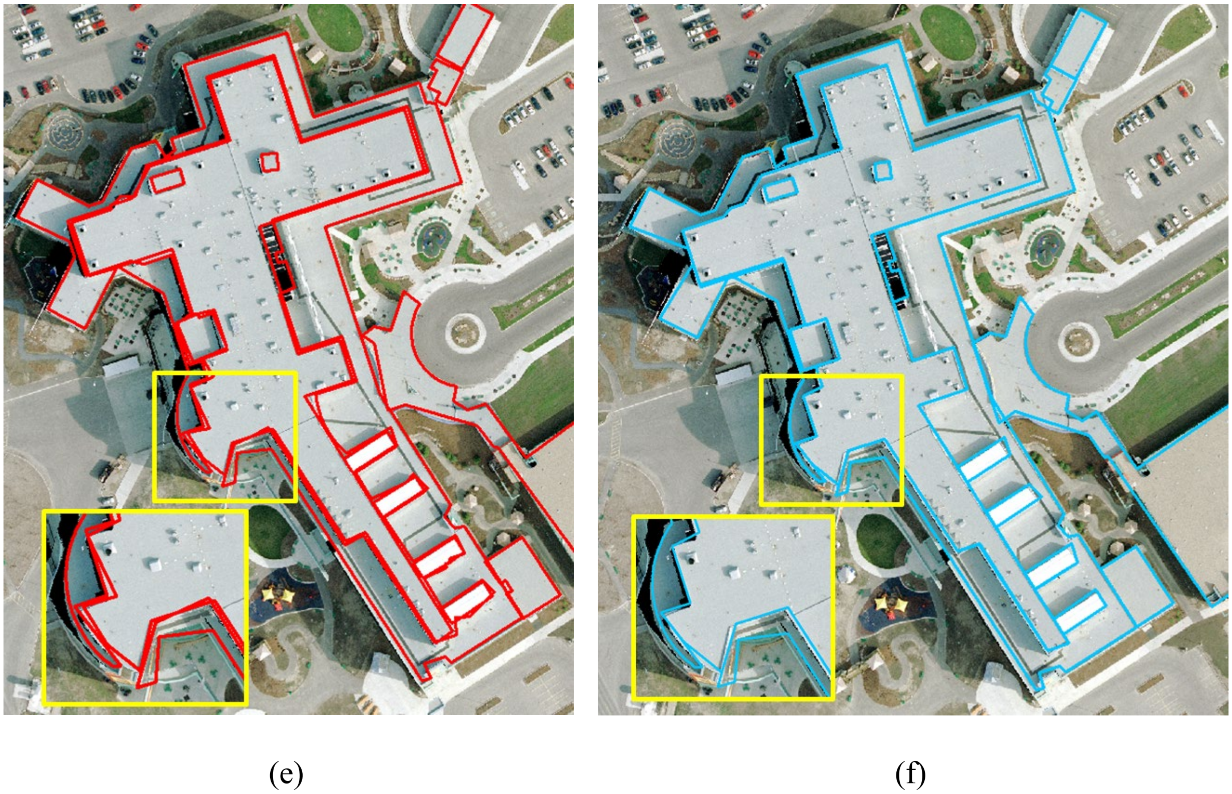


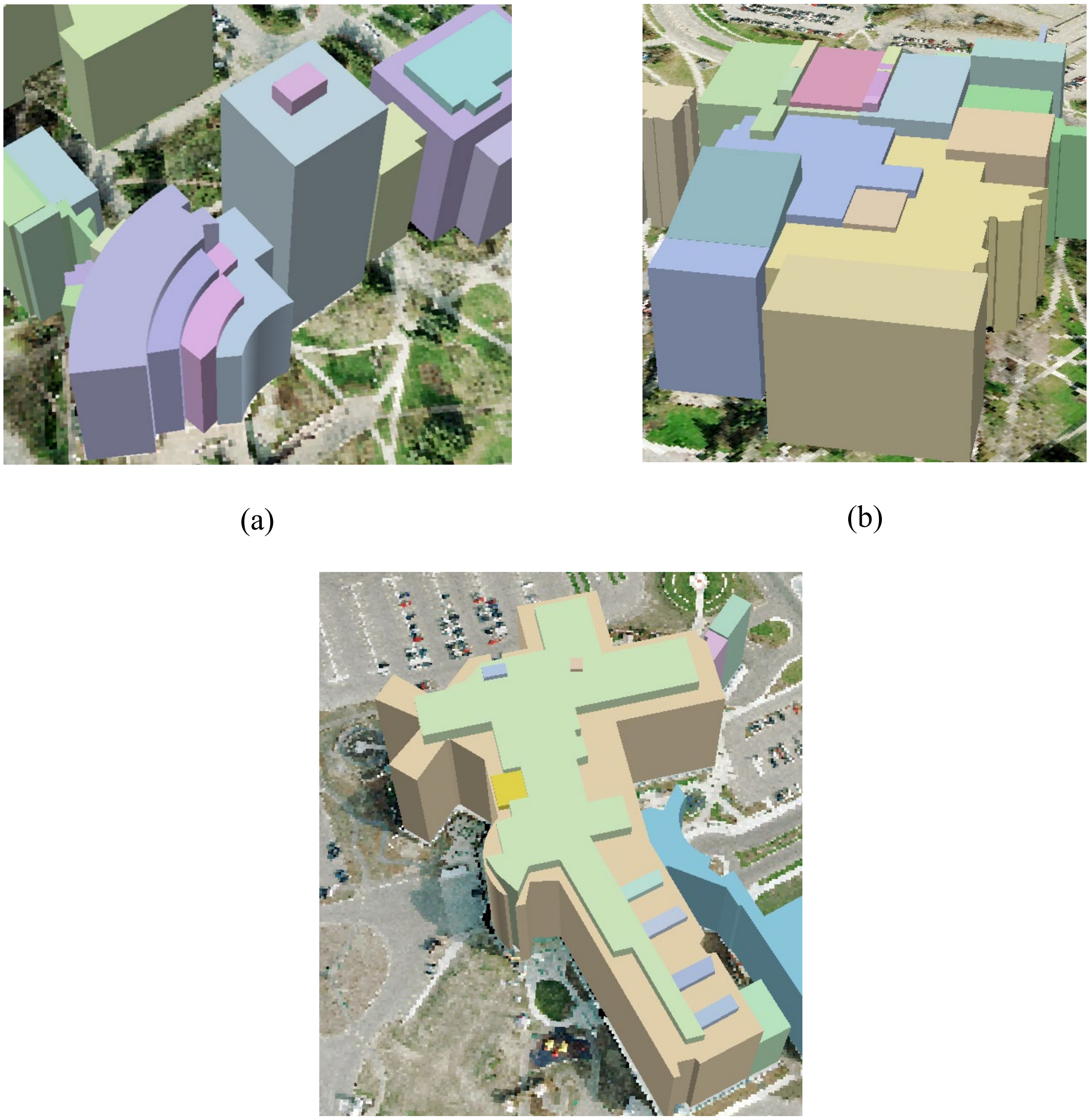
Figure 4.37. Comparison between the building primitives before and after applying water-tight building model generation strategy: (a) – (b) parallel straight-line segment merging, (c) – (d) non-parallel segment merging (curved-line to curved-line), and (e) – (h) parallel and non-parallel straight-line segment merging

Figure 4.37. Continued



(e)

(f)



(a)

(b)

(c)

Figure 4.38. Produced 3D seamless digital building models from the proposed methodology

Table 4.3. Statistical evaluation of the deviation between the reference dataset and the generated water-tight building models

	X (m)	Y(m)
Absolute min	0.001	0.001
Absolute max	1.957	1.858
Absolute mean	0.358	0.419
Absolute standard deviation	0.292	0.342
Mean	-0.075	0.047
Standard deviation	0.456	0.539
RMSE	0.462	0.541

The same reference data (i.e., 1004 corners) were used to evaluate the accuracy of the generated water-tight building models. Table 4.3 shows the statistical evaluation of the deviation between the reference dataset and the generated water-tight building models. As tabulated, the RMSE in X-direction is 46.2 cm and in Y-direction is 54.1 cm. The difference of RMSE before and after applying the water-tight DBM generation in both X and Y direction is within two centimeters. The change rate of corners from integrated building models to water-tight building model is 85.9%. It shows that the water-tight DBM generation strategy can improve the visualization results while maintaining the same level of accuracy.

4.8 Summary

This chapter proposed a boundary regularization strategy as well as a water-tight building model generation strategy that can automatically reconstruct seamless building models for right-angle, multi-orientation, and curved boundary primitives. This approach can generate robust building models while considering the characteristics of LiDAR data. Moreover, by means of simultaneous RMBR and LSA and the proposed stopping criteria, building models can be less sensitive to a highly irregular LiDAR boundary. With the proposed generation of generalized RMBR procedure, all the right-angle building can be reconstructed. For the proposed boundary regularization using the breakpoint detection approach, multi-orientation and curved boundary buildings can be reconstructed correctly. Furthermore, this thesis utilized the advantages of ERMBR and the BD approaches to generate more diverse and robust building models. This thesis also proposed a strategy for water-tight DBM generation to eliminate gaps between models associated with contiguous building primitives and providing better visualization of the outcome by analyzing the connectivity relationship of the extracted primitives.

5. LANE WIDTH ESTIMATION

5.1 Introduction

This chapter presents a new method to derive lane width estimates using point clouds acquired from a calibrated mobile mapping system. The calibration approach of the MMS is introduced first. After generating an accurate LiDAR point cloud, the road surface is extracted with the assistance of trajectory data. Lane markings are then identified based on the intensity data; and the lane marking centerline is derived for lane width estimation. Comprehensive testing was conducted to demonstrate the feasibility and performance of the proposed procedure, using six datasets collected in different seasons and different sensors.

5.2 System Architecture of the TMMS Used in this Thesis

The test datasets were captured by a designed terrestrial mobile mapping platform, which includes a Velodyne VLP-16 laser scanner, Velodyne HDL-32E laser scanner, FLIR Flea-2G camera, and SPAN-CPT GNSS/INS, as shown in Figure 5.1. The HDL-32E consists of 32 radially oriented laser rangefinders that are aligned from +10.67˚ to -30.67˚. In total, the vertical field of view (FOV) is 41.34˚. Also, the whole unit can rotate to achieve a 360˚ horizontal FOV. The point capture rate is around 700,000 points per second [94]. The VLP-16, which has 16 radially oriented laser rangefinders, is a lite version of HDL-32E. The vertical FOV is from -15˚ to +15˚ and the horizontal FOV is 360˚. The point capture rate is around 300,000 points per second [95]. The navigation system adopted in this thesis is the SPAN-CPT that combines GNSS and Inertial Measurement Unit (IMU) hardware inside a single enclosure. The GNSS collection rate is 20 Hz and the IMU measurement rate is 100 Hz [96]. The FLIR Flea-2G camera was used as an auxiliary sensor and was directly georeferenced by the SPAN-CPT unit. The FLIR Flea-2G camera has a maximum image resolution of 5 MP and has built-in ports for both triggering and strobe feedback signals.

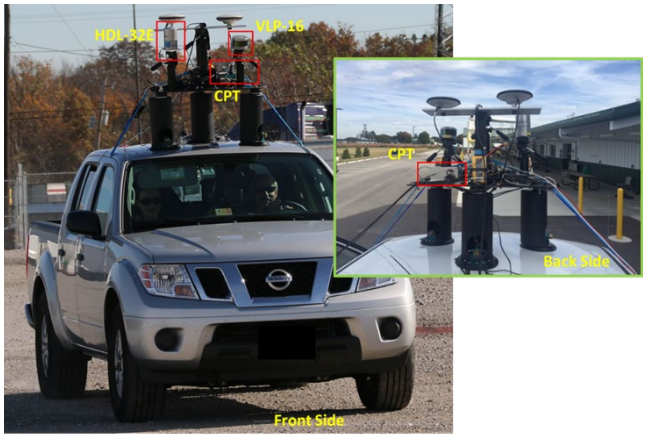


Figure 5.1. Terrestrial mobile mapping system used in this thesis

In order to derive a directly georeferenced LiDAR point cloud, the SPAN-CPT supplies sequentially precise time pulses, known as pulse-per-second (PPS) signals, which provide the ability to generate a time-tagged point cloud. Furthermore, the SPAN-CPT provides a navigation message, also known as the GPs Recommended Minimum specifiC - GPRMC message, including information related to position, rotation, and GPS time, which is transmitted over a dedicated RS-232 serial port and is received by the LiDAR unit via the Velodyne interface box in the form of serial data. This synchronization process is illustrated in Figure 5.2.

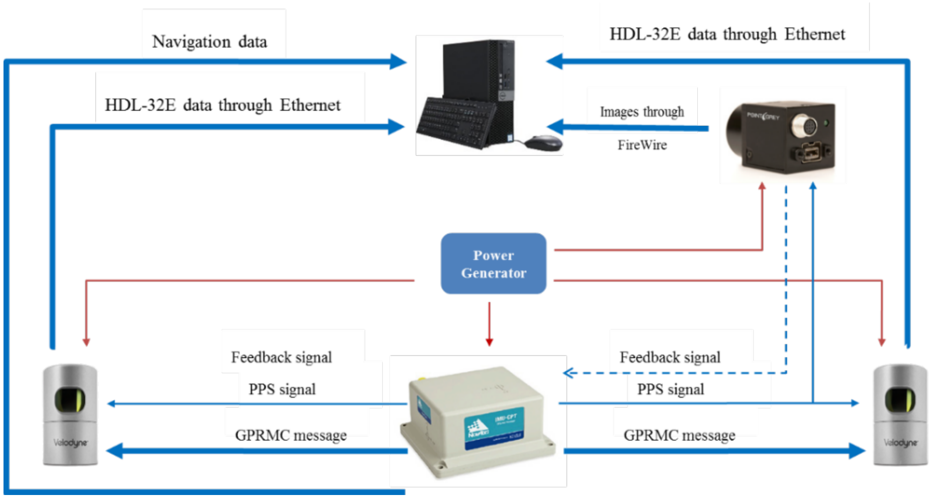


Figure 5.2. Synchronization process and data storage for the mobile mapping system

The accuracy of the derived LiDAR point cloud can be estimated from the utilized platform using error propagation. The SPAN-CPT can attain an accuracy of less than 2 cm in position and an accuracy of 0.008˚ and 0.035˚ in the roll/pitch and heading, respectively [96]. The Velodyne laser unit has a range accuracy of 2 cm. These values, along with the nominal standard deviation of the estimated system mounting parameters, are used to derive the expected accuracy in the computed mapping frame coordinates using the LiDAR Error Propagation calculator developed by Habib et al. [97]. The calculator specifications indicate an accuracy of around 2-3 cm at a range of 30 m. Since the accuracy of lane width estimates would directly depend on the LiDAR point cloud accuracy, the expected accuracy of the derived lane width should be around 3 cm.

5.3 Conceptual Basis of LiDAR Point Positioning and System Calibration

As illustrated in Figure 5.3, a typical directly geo-referenced multi-unit LiDAR system is comprised of three coordinate systems: the mapping frame, the GNSS/INS body frame, and the laser unit frame. These coordinate systems and their spatial/rotational relationships are used to define the mapping coordinates of a given point, $\boldsymbol{I}$, acquired from a mobile multi-unit LiDAR mapping system, as given by Equation 5.1, where a reference and slave sensors are considered. In the case of the spinning multi-beam laser scanners used in this thesis, the coordinates of a point relative to the laser unit coordinate system, $\boldsymbol{r_I^{luj}(t)}$, is defined by Equation 5.2 using the vertical angle, $\boldsymbol{\beta}$, determined by the fired laser beam ID; the horizontal angle, $\boldsymbol{\alpha}$, which depends on the rotation of the laser unit; and the range, $\boldsymbol{\rho}$, defined by the distance from the firing point to the footprint of the laser beam. The terms $\boldsymbol{r_{luj}^{lur}}$ and $\boldsymbol{R_{luj}^{lur}}$ are the rigidly defined lever arm and boresight matrix, respectively, relating the reference sensor, $\boldsymbol{lur}$, and the slave sensor, $\boldsymbol{luj}$. The lever arm, $\boldsymbol{r_{lur}^{b}}$, and boresight matrix, $\boldsymbol{R_{lur}^{b}}$, between the reference laser unit and IMU body frame coordinate systems are time-independent since the laser scanner and the IMU are rigidly fixed relative to each other. The lever arm components and boresight matrices can be derived from a system calibration process. The GNSS/INS integration provides the time-dependent position, $\boldsymbol{r_b^m(t)}$, and orientation, $\boldsymbol{R_b^m(t)}$, relating the body frame to the mapping frame.

$$r_I^m = r_b^m(t) + R_b^m(t)\, r_{lur}^b + R_b^m(t)\, R_{lur}^b\, r_{luj}^{lur} + R_b^m(t)\, R_{lur}^b\, R_{luj}^{lur}\, r_I^{luj}(t) \tag{5.1}$$

$$r_I^{luj}(t) = \begin{pmatrix} \rho(t)\cos\beta(t)\cos\alpha(t) \\ \rho(t)\cos\beta(t)\sin\alpha(t) \\ \rho(t)\ \sin\beta(t) \end{pmatrix} \tag{5.2}$$

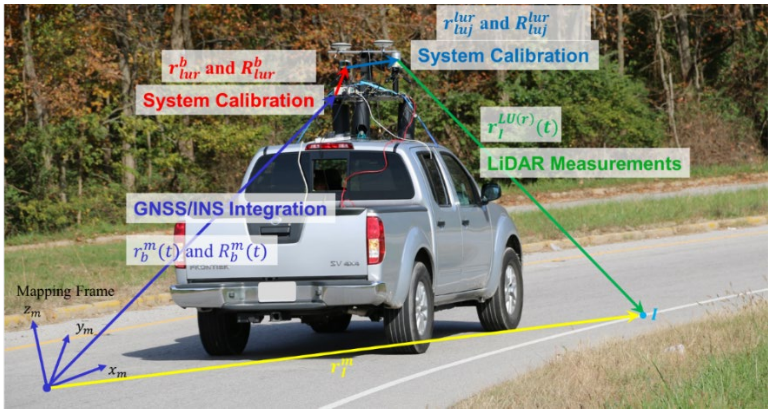


Figure 5.3. Illustration of point positioning of a directly geo-referenced multi-unit LiDAR system

The mounting parameters of each sensor are derived by minimizing the discrepancies among the conjugate features extracted from the point clouds obtained from different sensors in overlapping drive-runs [98]. The ideal mathematical condition for a conjugate point pair from two overlapping drive-runs is given by Equation 5.3. After reconstructing the point clouds from several drive-runs using the initial estimates for the mounting parameters, different types of geometric features (planar, and linear/cylindrical) can be used as conjugate features for the calibration procedure. Then, the mounting parameters for multiple sensors are derived simultaneously. In this thesis, several hut-shaped targets and highly reflective calibration boards are deployed and used as planar features for calibration, as shown in Figure 5.4. Moreover, outdoor objects, such as light poles and ground patches, are also used as linear/cylindrical and planar features, respectively.

$$r_I^m(\text{drive-run } 1) - r_I^m(\text{drive-run } 2) = 0 \tag{5.3}$$

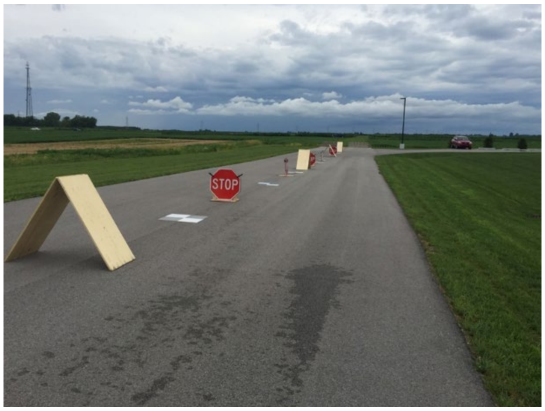


Figure 5.4. Calibration test field with hut-shaped targets and highly reflective boards

5.4 Lane Width Estimation Strategy

The proposed methodology for lane width estimation proceeds in three major steps: (1) road surface and lane marking extraction, (2) derivation of the lane marking centerline, and (3) lane width estimation. First, the road surface is extracted with the assistance of vehicle trajectory data. Next, the lane markings are extracted along the road surface by identifying high-intensity points. Since lane markings have a certain thickness, their centerline is derived for estimating the lane width. The flowchart of the proposed strategy is shown in Figure 5.5. The following subsections introduce the technical details of these steps.

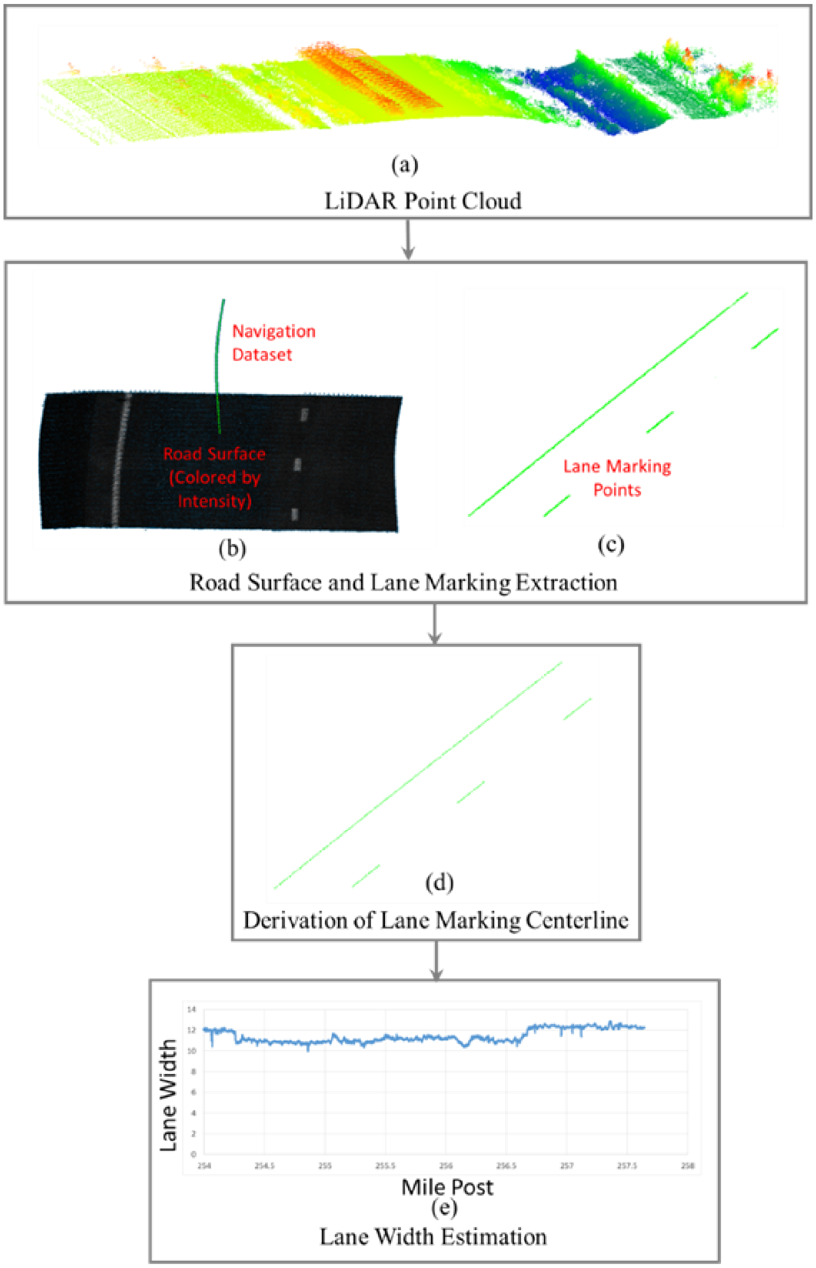


Figure 5.5. Flowchart of the lane width estimation strategy, (a) input LiDAR point cloud (colored by height), (b) extracted road surface, (c) candidate lane marking points, (d) derivation of lane marking centerline, and (e) lane width vs. longitudinal distance plot

5.4.1 Road Surface and Lane Marking Extraction

The proposed method focuses on lane width estimation along the driving lane. In this subsection, we introduce a method to extract the road surface and lane marking on the road surface along the driving lane. The trajectory data records the information of vehicle position and orientation, which are defined by the position and orientation of the IMU body frame of the GNSS/INS navigation system. Therefore, the LiDAR point cloud pertaining to the road surface can be derived with the help of the trajectory data and the IMU height above the road surface. First, the road surface points are extracted by setting a height threshold (h_{IMU}) and a threshold (d_w) for the lateral distance from the vehicle trajectory, as shown in Figure 5.6. The lateral distance threshold (d_w) is defined for both sides of the trajectory data across its direction and is used to only extract the road surface for the driving lane. The prescribed minimum lane width in highway areas is 12 ft [99]. So, in this thesis, the value for d_w is set as 3 m (9.843 ft) so the total lateral distance would be 6 m (19.686 ft), which is approximately 1.5 times the minimum designated lane width, thus ensuring extraction of the lane markings on both sides. The height threshold (h_{IMU}) is the expected normal distance from the IMU body frame to the road surface. It can be derived automatically by first randomly selecting a trajectory data point and searching the LiDAR point (P_i) with the closest (X, Y) coordinates and the least Z-coordinate. Then, a k-nearest neighbor search is applied to P_i for defining a road surface; and a plane-fitting is done to estimate the parameters of the road surface. Finally, the height threshold can be derived from the normal distance between the trajectory data point to the fitted plane. A pre-determined buffer (h_{buff}) for the height threshold (h_{IMU}) is necessary since the road surface may not be flat due to a pavement cross slope of 2% that is provided to ensure proper drainage [99]. The cross slope results in a height difference within the lane of up to 0.24 ft (0.07 m) for a 12 ft wide lane. Hence, the h_{buff} in this thesis was set as ± 0.2 m. The height buffer is illustrated in Figure 5.6. When the normal distance between a LiDAR point and the trajectory projection onto the road surface is smaller than d_w and its height is within the height buffer, the LiDAR point is regarded as a road surface point, as shown in Figure 5.5(b). After the road surface is extracted, the lane markings can be extracted. An intensity threshold (Th_I) is pre-defined to extract the points representing the lane markings. When the intensity of the extracted road surface points is larger than Th_I, they are regarded as potential candidate lane markings, as illustrated in Figure 5.5(c). One should note that a constant pre-defined value was used for the intensity threshold instead of an adaptive value because the laser beams contributing towards the lane markings along the driving direction are homogeneously distributed and the intensity value from lane markings hence are homogeneous as well along the entire trajectory.

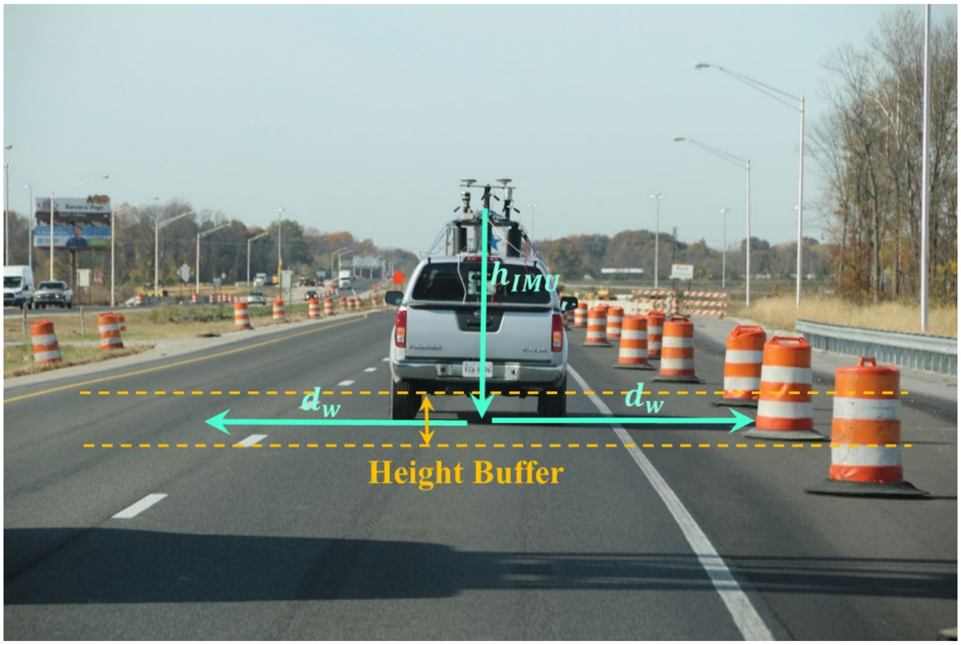


Figure 5.6. Illustration of the various thresholds for road surface extraction: the height threshold (h_{IMU}) and the lateral distance threshold (d_w) .

5.4.2 Derivation of the Lane Marking Centerline

Since lane markings are strips of a finite width, it is imperative to derive their centerline for lane width estimation. The points extracted as potential candidate lane markings in the previous step may also arise from other features, such as road markings, road surface pavement, and rubble within construction work zones. Therefore, these non-lane marking points must be removed before proceeding to derivation of the centerline. The derivation of centerline for lane marking segments from the extracted high intensity points is achieved using the following steps: (1) clustering potential candidate lane marking points using a distance-based region growing, (2) partitioning such clusters into subgroups, (3) removal of non-lane marking points, and (4) centerline generation and down-sampling to derive centerline segments.

First, a distance-based region growing with a distance threshold (Th_{dist}) is conducted to group neighboring high intensity points into clusters. Th_{dist} can be decided based on an analysis of the local point spacing along the road surface. When the number of points in a cluster is less than a threshold (Th_{pt}), which can be decided according to the local point density, the cluster is regarded as a non-lane marking and removed, as shown in Figure 5.7(a-b). Since road lanes may not be straight, each cluster is divided into small segments to represent curved lane markings by polylines. First, the main direction of each cluster is derived using PCA. Then, each cluster is partitioned into segments based on a fixed length ($Th_{partition}$) along its main direction. The standard curvature design for highways states a minimum radius of 2,040 ft for a driving speed of 70 miles/hr [99]. Based on this specification, we designate a value of 4 m for $Th_{partition,}$ which is not only larger than a dash line but also can be regarded as a straight line for this radius of curvature. The maximum distance between the arc (with 2,040 ft radius of curvature) and the corresponding chord of length 4 m is 0.32 cm, which is within the noise level of the mobile mapping system. These partitioned segments are assumed to be straight, as illustrated Figure 5.7(c). Next, the non-lane marking points of each segment are detected and removed using a RANSAC strategy [100], as depicted in Figure 5.7(d). The RANSAC algorithm randomly selects a minimal number of data points required to construct an estimate of a model and then checks the number of points from the entire dataset that are consistent with the estimated model. Here, the model estimate consists of line parameters. Therefore, two randomly selected points are used to define a straight line. Then, the normal distances of the remaining points within the partitioned segment to the line are calculated. If the normal distance of a point is larger than threshold (W_{lane}), it is regarded as an outlier. Threshold (W_{lane}) is defined according to the expected width of a lane marking. This procedure is repeated a certain number of times while keeping track of the estimated models and their corresponding consensus set size. The model with the largest consensus set is designated to be the best model. Finally, the largest consensus set is regarded as points belonging to lane markings and are utilized to derive the centerline parameters of the segment using a LSA model-fitting.

Note that RANSAC can only remove outliers within a segment but cannot deal with the case where the entire segment does not represent a lane marking. In this case, the trajectory data are utilized to identify the segments that do not belong to lane markings. In most cases, the lane markings are parallel to the vehicle trajectory. Therefore, this characteristic can be used to determine whether a segment represents a lane marking or not. When the direction of a segment is similar, (i.e., less than an angle threshold (Th_{angle})) to the trajectory, it is regarded as a lane marking segment, as shown in Figure 5.7(e). Considering scenarios where vehicles might change lanes, Th_{angle} is set as 5˚. According to a designated average work zone speed of 45 miles/hr and an average time duration of 4 sec during lane change [101], the slope of the trajectory relative to the lane alignment would be around 2.6˚. Therefore, a Th_{angle} of 5˚ can ensure that the lane marking points are not removed. The centerline parameters of a segment (a direction vector and a point along the vector) represent the direction of the segment. In order to generate a centerline segment, all the points in the inlier set of the segment are projected onto the LSA-based centerline, as illustrated in Figure 5.7(f). In this case, the projected points would be dense and a down-sampling strategy is applied to reduce the number of the projected points, as depicted in Figure 5.7(g). When applying the down-sampling strategy, the sample points are picked within a segment at a fixed distance interval ($Th_{interval}$), as shown in Figure 5.7(g). One should notice that the down-sampling strategy is mainly used to remove overlapping points and reduce the processing time. The value of $Th_{interval}$ would not affect the results since the points along the centerline segment represent a fitted straight line. Even if the number of points is reduced, those sampled points would still represent a fitted straight line.

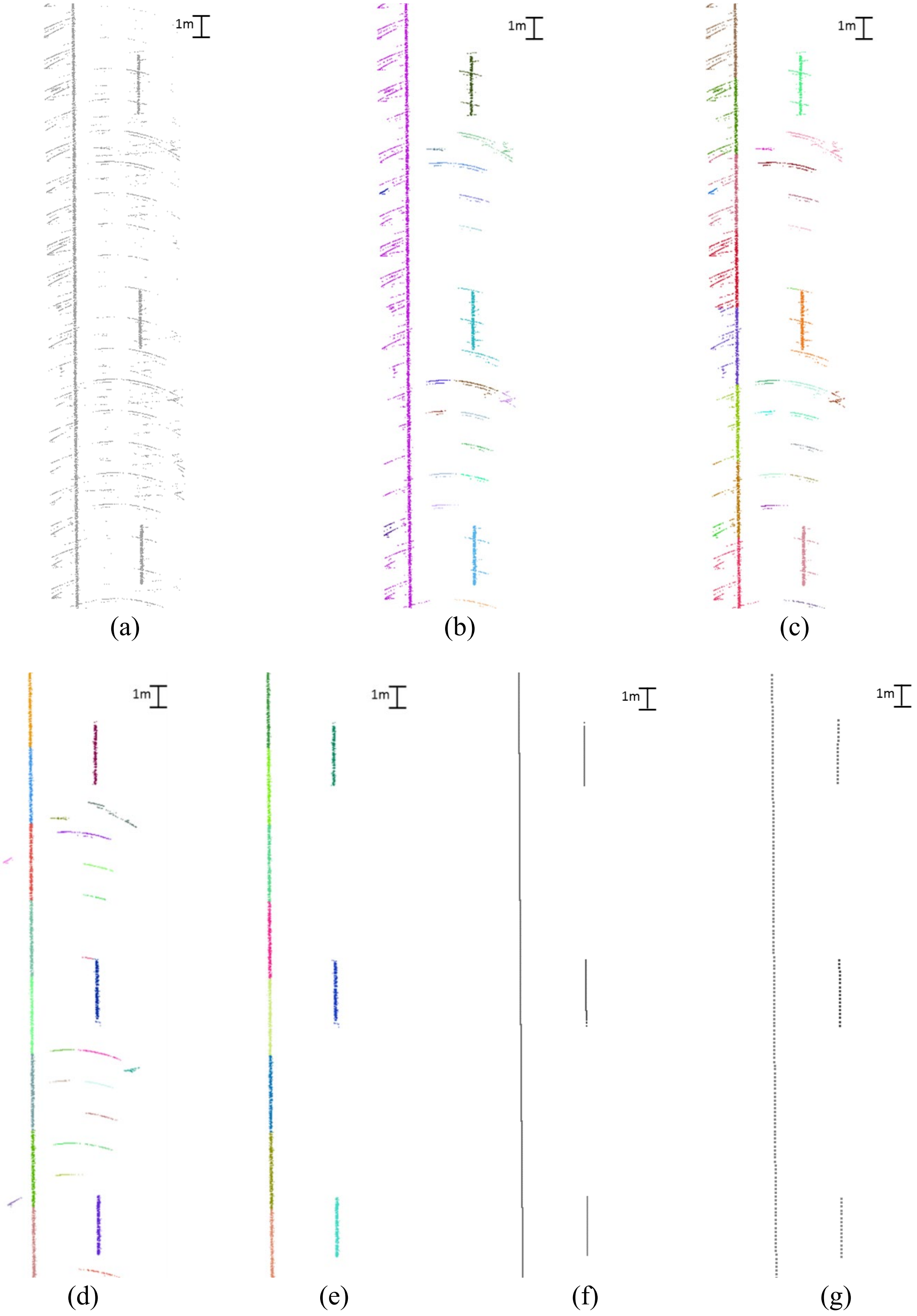


Figure 5.7. Estimation of lane marking centerline procedure (a) candidate lane marking points, (b) region growing-based clustering, (c) cluster partition, (d) outlier removal: RANSAC-based, (e) outlier removal: trajectory-based, (f) lane marking centerline, (g) down-sampled lane marking centerline

5.4.3 Lane Width Estimation

The lane width is derived as the normal distance between the centerlines of lane markings located on opposite sides of the trajectory along the driving lane. First, the two trajectory data points closest to the queried centerline point are searched to define a trajectory vector along the driving direction. Then, the queried centerline point can be determined to be located either to the left or to the right side of the trajectory vector. After identifying the left and right lane markings, as shown in Figure 5.8(a) by the blue dash line and red side line, the normal distance from a point on one side to the straight line defined by the two closest points on the other side can be derived and regarded as the lane width, as illustrated in Figure 5.8(b).

However, the lane markings may not be continuous (e.g., dash line or lane markings without strong reflection), as shown in Figure 5.9. In such cases, they could lead to inaccurate estimation of the normal distance. Hence, an interpolation between the centerline points is applied first to fill the gaps, as shown in Figure 5.8(c). In this step, the centerline points first are ordered along the driving direction based on the trajectory information. When the distance between two successive points is larger than a threshold ($Th_{interval}$), a linear interpolation is conducted between the two points. Figure 5.8(d) is an example of the inaccuracy in the derived normal distance when the interpolation is not applied. One should note that after the interpolation, the normal distance derived from a point on the left side to the straight line defined by the two closest points on the right side would be similar to the normal distance derived from a point on the right side to the straight line defined by the two closest points on the left side.

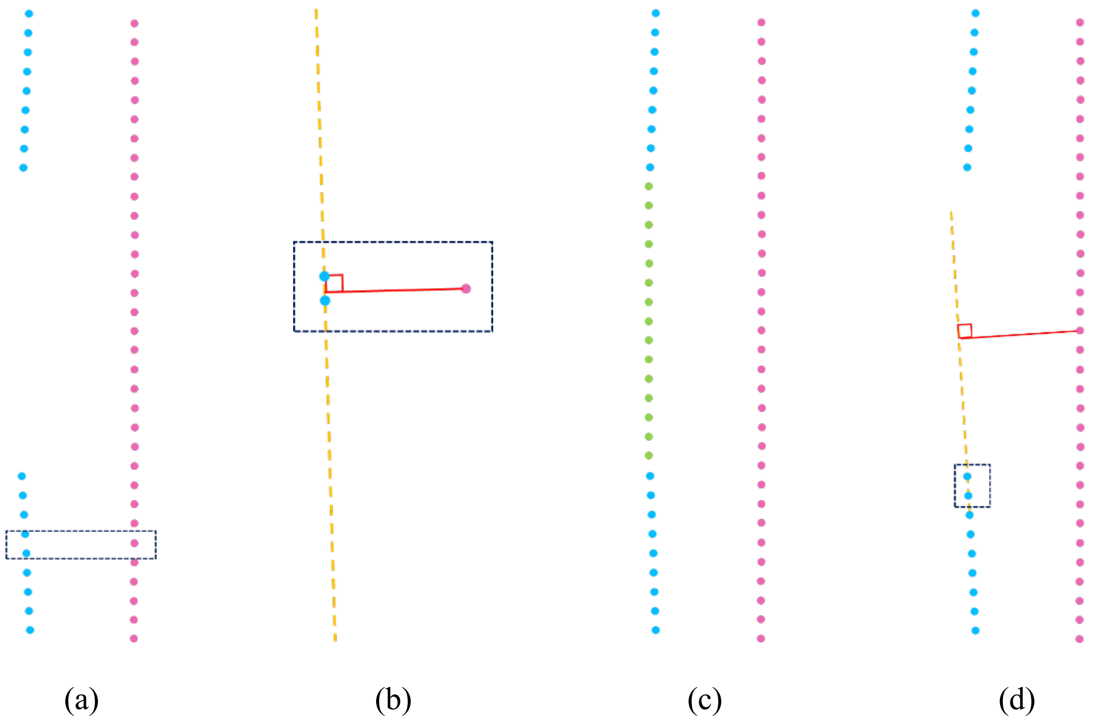


Figure 5.8. Lane width estimation procedure: (a) illustration of opposite-side lane marking centerlines, (b) lane width derivation, (c) interpolation among lane marking centerlines, and (d) inaccuracy in lane width estimation without centerline interpolation

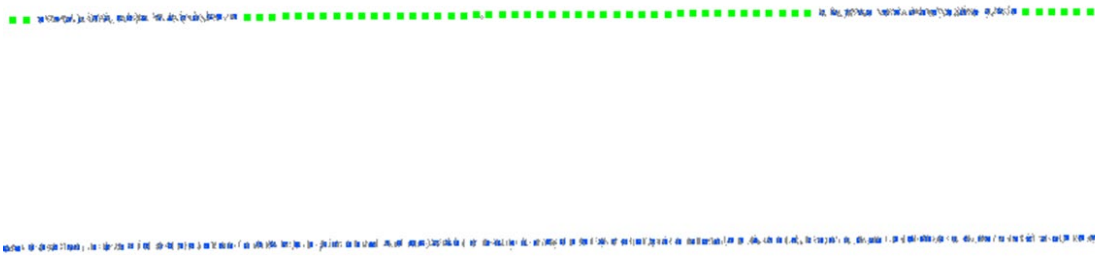
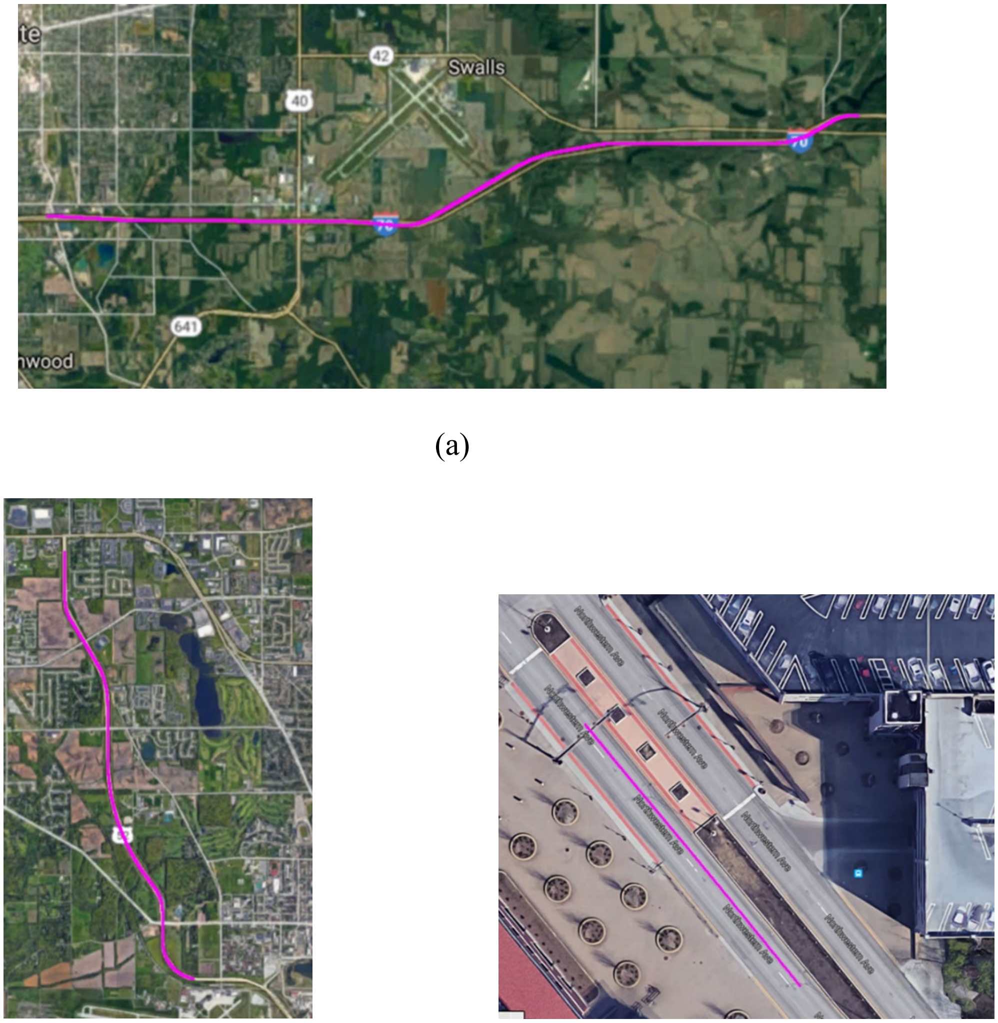


Figure 5.9. Non-continuous lane markings: dash line (gray points: lane marking points, blue points: lane marking centerline, green points: interpolated points)

5.5 Experimental Results

5.5.1 Data Description

To illustrate the performance and feasibility of the proposed calibration strategy and lane width estimation methodology, this section presents the experimental results from a total of six datasets collected for three road segments using different sensors in different seasons. The first road segment was surveyed on two different dates and is located in an interstate highway work zone area. Both datasets were collected while driving westbound from mile post 19 to mile post 10 (a total of 9 miles). This road segment is highlighted in Google Maps, as shown in Figure 5.10(a). The second road segment was surveyed on three different dates and is a 3.7 mile long segment located on a U.S. highway, as illustrated in Figure 5.10(b). The datasets for these two road segments were used for lane width estimation and repeatability analysis of the proposed strategy. The third road segment is located at a main arterial, which is a 0.12 mile long segment, as depicted in Figure 5.10(c). This dataset was used to compare the derived lane width values to the on-site manual measurements and thye manually digitized lane marking centerline in order to demonstrate the accuracy of the proposed method. The details of the test datasets are listed in Table 5.1.



(a)

(b)

(c)

Figure 5.10. Location of test datasets and their trajectory (red): (a) interstate highway, (b) U.S. highway, and (c) main arterial

Table 5.1. Details of test datasets collected in this thesis

Road Segment	**Collection Date**	**Used Sensors**	**Length**	**Driving Speed**
Interstate Highway	2016/11/05	HDL32E-2 VLP16-1	9 miles	40 miles/hr
	2017/05/02	HDL32E-2 HDL32E-3		40 miles/hr
U.S. Highway	2016/11/30	HDL32E-2 VLP16-1	3.7 miles	40 miles/hr
	2017/02/11	HDL32E-2 HDL32E-3		35 miles/hr
	2017/08/17	HDL32E-4 HDL32E-5		35 miles/hr
Main Arterial	2016/10/11	HDL32E-2 VLP16-1	0.12 miles	20 miles/hr

Note: HDL32E-2, HDL32E-3, HDL32E-4, and HDL32E-5 denote different LiDAR sensors of the same model

The mobile mapping system was mounted on different vehicles for these six data collections. To ensure the accuracy of the acquired point clouds, the calibration procedure was applied every time. The mounting parameters (i.e., boresight angles $\boldsymbol{\Delta\omega}, \boldsymbol{\Delta\varphi}, \boldsymbol{\Delta\kappa}$ and lever arm $\boldsymbol{\Delta X}, \boldsymbol{\Delta Y}, \boldsymbol{\Delta Z}$) derived from the multi-sensor calibration procedure are listed in Table 5.2. Based on the mounting parameters and navigation data, the point clouds were generated for lane width estimation.

Table 5.2. Mounting parameters of the MMS used to collect all datasets, as estimated from the multi-sensor system calibration procedure

Date	**Mounting Parameters**					
	$\boldsymbol{\Delta X}$ (m)	$\boldsymbol{\Delta Y}$ (m)	$\boldsymbol{\Delta Z}$ (m)	$\boldsymbol{\Delta\omega}$ (°)	$\boldsymbol{\Delta\varphi}$ (°)	$\boldsymbol{\Delta\kappa}$ (°)
Interstate Highway						
2016/11/05	HDL32E-2 Calibration Parameters $\boldsymbol{R}_{\boldsymbol{HDL32E2}}^{\boldsymbol{b}}$ $\boldsymbol{r}_{\boldsymbol{HDL32E2}}^{\boldsymbol{b}}$					
	0.471	0.2057	0.3000	-5.868	-0.1548	-54.0072
	VLP16-1 Calibration Parameters $\boldsymbol{R}_{\boldsymbol{VLP161}}^{\boldsymbol{HDL32E2}}$ $\boldsymbol{r}_{\boldsymbol{VLP161}}^{\boldsymbol{HDL32E2}}$					
	-0.271	-0.376	-0.073	-2.092	-4.161	22.014
2017/05/02	HDL32E-2 Calibration Parameters $\boldsymbol{R}_{\boldsymbol{HDL32E2}}^{\boldsymbol{b}}$ $\boldsymbol{r}_{\boldsymbol{HDL32E2}}^{\boldsymbol{b}}$					
	0.470	0.196	0.319	-6.110	0.488	-56.728
	HDL32E-3 Calibration Parameters $\boldsymbol{R}_{\boldsymbol{HDL32E3}}^{\boldsymbol{HDL32E2}}$ $\boldsymbol{r}_{\boldsymbol{HDL32E3}}^{\boldsymbol{HDL32E2}}$					
	-0.246	-0.384	-0.003	-0.734	0.1054	-111.320
U.S. Highway						
2016/11/30	HDL32E-2 Calibration Parameters $\boldsymbol{R}_{\boldsymbol{HDL32E2}}^{\boldsymbol{b}}$ $\boldsymbol{r}_{\boldsymbol{HDL32E2}}^{\boldsymbol{b}}$					
	0.476	0.201	0.300	-6.047	-0.095	-58.870
	VLP16-1 Calibration Parameters $\boldsymbol{R}_{\boldsymbol{VLP161}}^{\boldsymbol{HDL32E2}}$ $\boldsymbol{r}_{\boldsymbol{VLP161}}^{\boldsymbol{HDL32E2}}$					
	-0.246	-0.400	-0.089	-1.581	-4.274	26.810
2017/02/11	HDL32E-2 Calibration Parameters $\boldsymbol{R}_{\boldsymbol{HDL32E2}}^{\boldsymbol{b}}$ $\boldsymbol{r}_{\boldsymbol{HDL32E2}}^{\boldsymbol{b}}$					
	0.466	0.178	0.324	-6.049	0.456	-63.961
	HDL32E-3 Calibration Parameters $\boldsymbol{R}_{\boldsymbol{HDL32E3}}^{\boldsymbol{HDL32E2}}$ $\boldsymbol{r}_{\boldsymbol{HDL32E3}}^{\boldsymbol{HDL32E2}}$					
	-0.218	-0.378	-0.004	-1.464	-4.425	-105.495
2017/08/17	HDL32E-4 Calibration Parameters $\boldsymbol{R}_{\boldsymbol{HDL32E4}}^{\boldsymbol{b}}$ $\boldsymbol{r}_{\boldsymbol{HDL32E4}}^{\boldsymbol{b}}$					
	0.446	0.204	0.322	-5.692	0.941	-53.531
	HDL32E-5 Calibration Parameters $\boldsymbol{R}_{\boldsymbol{HDL32E5}}^{\boldsymbol{HDL32E4}}$ $\boldsymbol{r}_{\boldsymbol{HDL32E5}}^{\boldsymbol{HDL32E4}}$					
	-0.286	-0.375	-0.009	-0.696	-0.478	-12.616
Main Arterial						
2016/10/11	HDL32E-2 Calibration Parameters $\boldsymbol{R}_{\boldsymbol{HDL32E2}}^{\boldsymbol{b}}$ $\boldsymbol{r}_{\boldsymbol{HDL32E2}}^{\boldsymbol{b}}$					
	0.425	0.191	0.300	-5.839	-1.460	-47.953
	VLP16-1 Calibration Parameters $\boldsymbol{R}_{\boldsymbol{VLP161}}^{\boldsymbol{HDL32E2}}$ $\boldsymbol{r}_{\boldsymbol{VLP161}}^{\boldsymbol{HDL32E2}}$					
	-0.298	-0.336	-0.009	-3.537	-3.147	15.996

5.5.2 Experimental Results of Lane Width Estimation

In the following four sub-sections, four experiments are discussed. First, the lane width from a calibrated dataset was estimated and these results then were compared to the results obtained from the same dataset generated after adding a bias to the system mounting parameters. This comparison showed the importance of accurate system calibration to derive accurate lane width estimates. In the second experiment, the 2016/11/30 dataset was used, which included two different types of spinning multi-beam laser scanners (VLP16 and HDL32E) to compare their performance in lane width estimation. Moreover, the similarity of the derived lane width from the two sensors demonstrated the accuracy of the derived mounting parameters from the introduced multi-sensor system calibration procedure. The third experiment aimed to confirm the precision of the lane width estimation and calibration strategies by comparing the lane width vs the mile post plots obtained from five datasets for an interstate and a U.S. highway (with a total length of approximately 25 miles) scanned by different sensors in different seasons. Finally, the last experiment was conducted to show the accuracy of the derived lane width by comparing the results obtained for the dataset collected over the main arterial with that from the manually digitized lane marking centerline and the on-site manual measurements. The various thresholds used for these experiments, which are listed in Table 5.3, were tested for a road segment of over 25 miles using different sensor, seasons, and locations to demonstrate that the derived lane width estimates were insensitive to the value of these thresholds.

Table 5.3. Pre-defined thresholds for lane width estimation

Name	Value
Lateral distance threshold for road surface extraction (d_w)	3.0 m
Height buffer for road surface extraction (h_{buff})	±0.2 m
Intensity threshold (Th_I)	30-40
Distance threshold for distance-based region growing (Th_{dist})	0.3 m & 0.4 m*
Minimum number of points to define candidate lane marking (Th_{pt})	50 pts & 25 pts*
Length for the partitioned segments ($Th_{partition}$)	4 m
Lane marking width threshold (W_{lane})	0.12 m
Angle threshold for rejecting non-lane markings based on their orientation relative to the trajectory (Th_{angle})	5°
A distance interval threshold for down-sampling and interpolation ($Th_{interval}$)	0.2 m

* These values are used while dealing with point clouds obtained from a single VLP-16 scanner.

The proposed method was implemented using a PC with an Intel® Core™ i7-4790 CPU at 3.60 GHz. In the case of a point cloud with 0.8 billion points over a stretch of five miles captured from two HDL32E sensors, the processing time for each step is listed in Table 5.4. Note that the processing time varied according to the performance of the computer and can be improved using parallel processing. The proposed strategy mainly focuses on the estimation of the lane width in work zones for the inspection process of the delineated lane markings. Hence, real-time processing was not critical in this thesis. The computation performance for the proposed strategy demonstrated that the lane width can be derived in a short enough period to provide the construction project manager with timely information.

Table 5.4. Data processing time for each step of lane width estimation

Item	Time
System calibration	2 hrs
Point cloud reconstruction	0.5 hr
Road surface extraction	1.5 hr
Lane marking extraction and lane width estimation	5 mins

5.5.2.1 Impact of Mounting Parameters on Lane Width Estimation

In this section, the point clouds generated from the HDL32E-2 and HDL32E-3 sensors on 2017/02/11 were used to demonstrate that deviations in the mounting parameters can impact the derived lane width estimates. In this experiment, a +2˚ and -2˚ bias was added to the boresight parameters ($\boldsymbol{\Delta\omega}, \boldsymbol{\Delta\varphi}, \boldsymbol{\Delta\kappa}$) of the HDL32E-2 and HDL32E-3 sensors, respectively, to evaluate the effect on lane width estimation from the dataset generated using different estimates of the mounting parameters. Note that the two cases are characterized only by the difference in their mounting parameters and all the other environmental variables, such as the sensors, trajectory, time and date of data collection, are identical. Figure 5.11(a) shows the extracted lane marking points for the dataset generated using accurate and inaccurate estimates for the mounting parameters in green and orange colors, respectively. It can be seen that the lane marking points extracted using inaccurate mounting parameters are highly distorted. Figure 5.11(b-c) shows the derived centerline (in blue) for the two different cases.

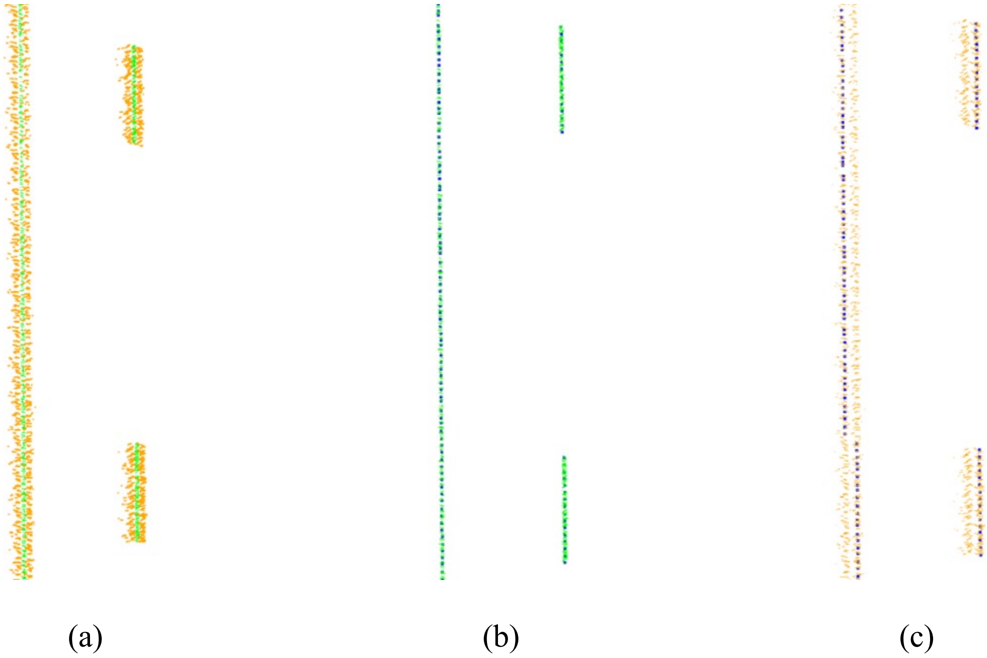


Figure 5.11. Lane marking points and derived lane marking centerline: (a) lane marking points (green: using accurate mounting parameters and orange: using inaccurate mounting parameters);(b) lane marking points (green) and derived centerline (blue) using accurate mounting parameters; and (c) lane marking points (orange) and derived centerline (blue) using inaccurate mounting parameters

The comparison between lane width estimates obtained using the two different sets of mounting parameters is shown in Figure 5.12. The mean, standard deviation, and RMSE values listed in Table 5.5 quantify the effect of inaccurate mounting parameters on the lane width estimation. Table 5 shows that when compared with the results obtained using accurate estimates of mounting parameters, a change of 2˚ in the boresight parameters of both sensors resulted in an RMSE of 27.91 cm. Although the lane width estimation strategy includes outlier removal and LSA-based line fitting that can minimize the effect of inaccurate mounting parameters, it cannot mitigate the effects altogether.

Table 5.5. Statistics of comparison between lane width estimates using accurate and inaccurate values of mounting parameters

Compared Datasets	**Mean**	**Standard Deviation**	**RMSE**
2017/02/11 dataset with different estimates of mounting parameters	23.28 cm (0.76 ft)	15.40 cm (0.51 ft)	27.91 cm (0.92 ft)

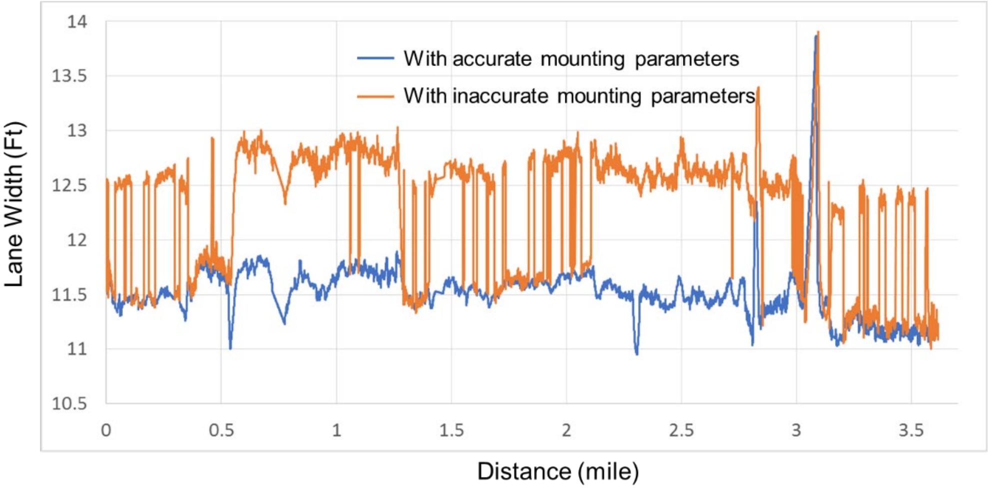


Figure 5.12. Comparison of lane width estimates using accurate and inaccurate values of the mounting parameters for the 2017/02/11 dataset from U.S. Highway

5.5.2.2 Performance Evaluation of Different Sensors

The lane width estimates derived from point clouds acquired by the VLP16 and HDL32E sensors on 2016/11/30 for the U.S. Highway dataset were used to compare the performance of these units for lane width estimation and to analyze the accuracy of the derived mounting parameters from the introduced multi-sensor calibration procedure. In this experiment, the distance threshold for distance-based region growing (Th_{dist}) for the VLP16 dataset was increased to 0.4 m since the point cloud captured using VLP16 was much sparser than that from HDL32. The minimum number of points to define a candidate lane marking (*Th_{pt}*) for the VLP 16 dataset is 25 points, which is half of the one for the HDL32E dataset due to the difference in the number of acquired pulses per second. The lane width comparison analysis is shown in Figure 5.13 and the quantitative evaluation is shown in Table 5.6.

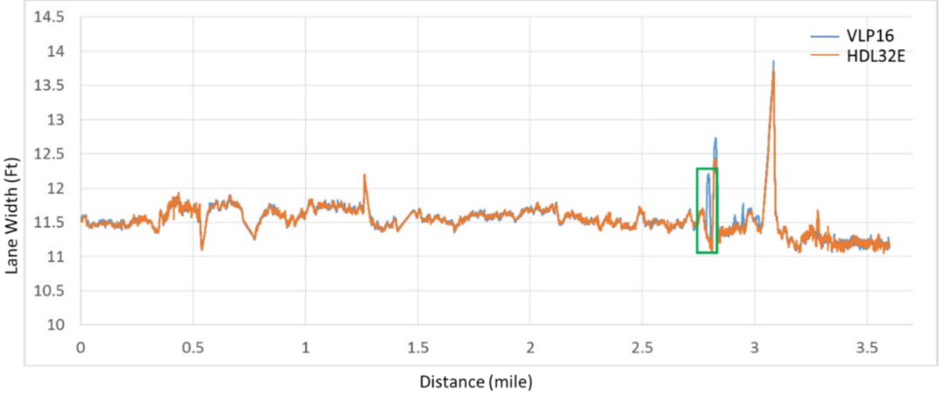


Figure 5.13. Comparison of estimated lane width from VLP16 and HDL32E for the 2016/11/30 dataset along the U.S. highway

Table 5.6. Statistics of comparison between the lane width estimates from VLP16 and HDL32E sensors

Compared Datasets	Mean	Standard Deviation	RMSE
VLP16 & HDL32E from 2016/11/30	0.007 cm (0.023 ft)	1.53cm (0.050 ft)	1.68cm (0.055 ft)

The quantitative evaluation in Table 5.6 indicates that the difference between the derived lane width estimates from VLP16 and HDL32E was small, which shows that both sensors are equally capable of accurate lane width estimation. It also implied that the estimated mounting parameters of the two laser scanners are accurate. However, there was a 0.86ft (26.21 cm) difference in the location highlighted by the green box in Figure 5.13. This difference was attributed to the sparse nature of the VLP16 point cloud, for which the clustered lane marking points were less than the Th_{pt} threshold so centerline extraction at that location was incomplete and the interpolated centerline deviated from the lane markings. This problem was exacerbated by a curved road segment at this location, thus resulting in inaccuracy in centerline interpolation. Figure 5.14(a-b) shows the lane marking points (gray) and derived centerline (green and red) from both laser scanners. After interpolation of the derived centerline, as illustrated in Figure 5.14(c), the maximum deviation in that portion was around 0.86 ft. Since most lane width estimates from the VLP16 and HDL32E dataset along this road segment were compatible, the effect of this bias was insignificant. However, such a discrepancy would not arise in the case of a straight portion, as illustrated in Figure 5.15. This analysis led to the conclusion that the lane width estimates from the two sensors were compatible, but in order to avoid the discrepancies caused by the sparse nature of the point cloud acquired from VLP16, a lower speed for data collection is recommended when using this sensor.

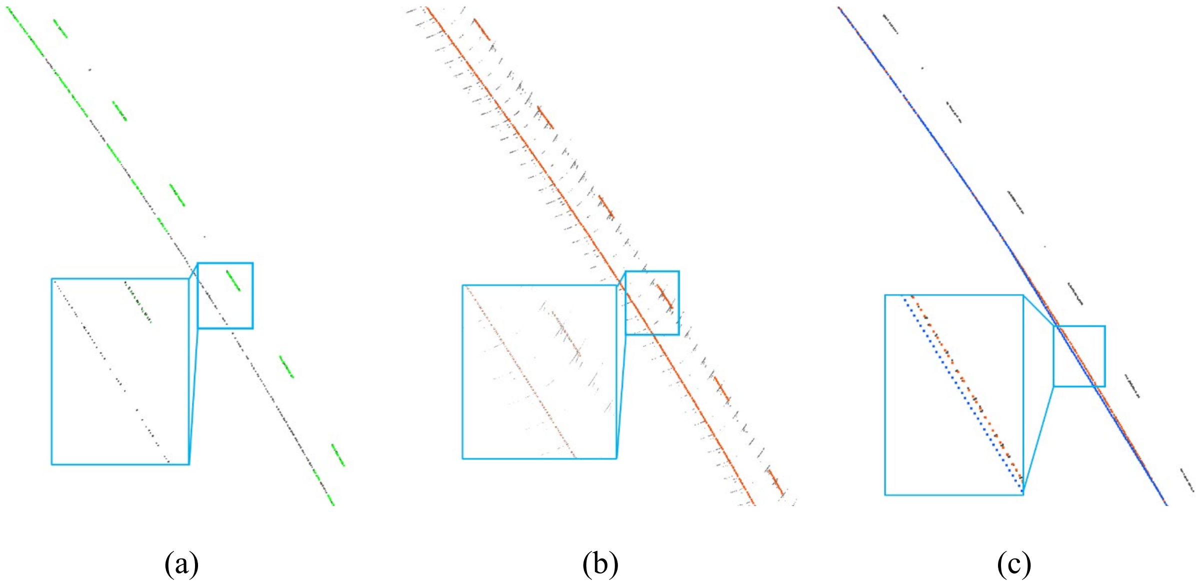


Figure 5.14. Anomalies in the marked area of Figure 5.11: (a) candidate lane marking points (gray) and the derived centerlines (green) from VLP16; (b) candidate lane marking points (gray) and the derived centerlines (red) from HDL32E; and (c) candidate lane marking points from VLP16 (gray), the interpolated centerline from VLP16 (blue), and the derived centerline from HDL32 (red)

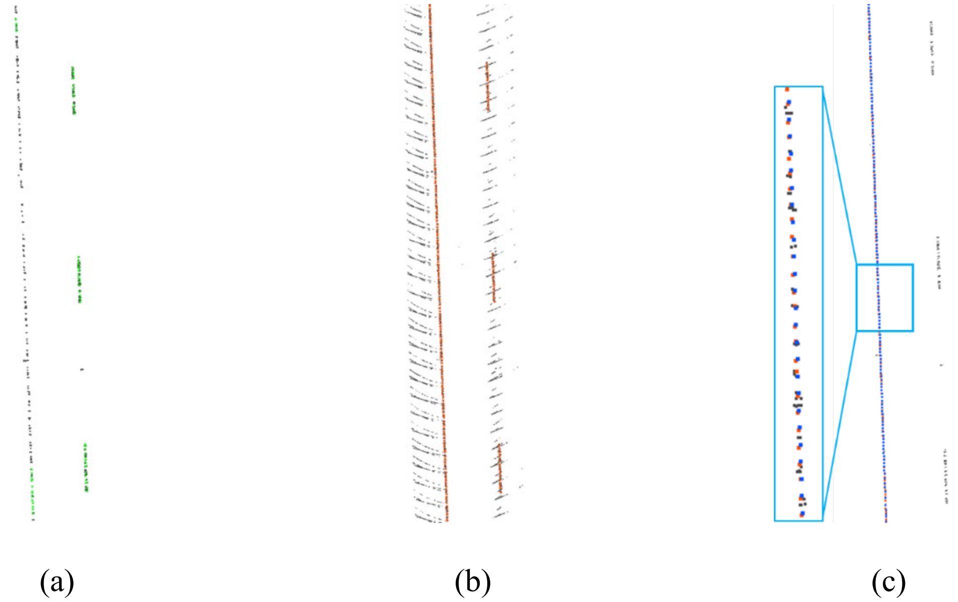


Figure 5.15. Interpolation of straight portion: (a) candidate lane marking points (gray) and the derived centerlines (green) from VLP16; (b) candidate lane marking points (gray) and the derived centerlines (red) from HDL32E; and (c) candidate lane marking points from VLP16 (gray), the interpolated centerline from VLP16 (blue), and the derived centerline from HDL32 (red)

5.5.2.3 Repeatability Analysis of Lane Width Estimation

This section provides an evaluation of the lane width estimation strategy using the datasets collected for two different road segments – an interstate highway work zone area and a U.S. highway segment. The intensity of the lane markings can be affected by the sensor units used, the weather, the incidence angle of the laser beams, and the quality of the lane markings. Therefore, the Th_I threshold values used in the processing of these six datasets varied from 30 to 40, depending on the above factors for each dataset. For the interstate highway work zone area, two datasets were used to derive the lane width; and the obtained values were compared by plotting the lane width vs the mile post, as shown in Figure 5.16.

Figure 5.16 shows a spike in the 2016/11/05 dataset, which is indicated by green box 1. This spike was attributed to the poor condition of the lane markings in the work zone, as shown in Figure 5.17(a), which in turn resulted in an erroneous lane marking extraction, as illustrated in

Figure 5.17(b). However, this anomaly does not appear in the 2017/05/02 dataset, as shown in Figure 5.18. Since the poor lane markings faded away after six months, the intensity of the LiDAR points at that area was not as strong as the one from the 2016/11/05 dataset.

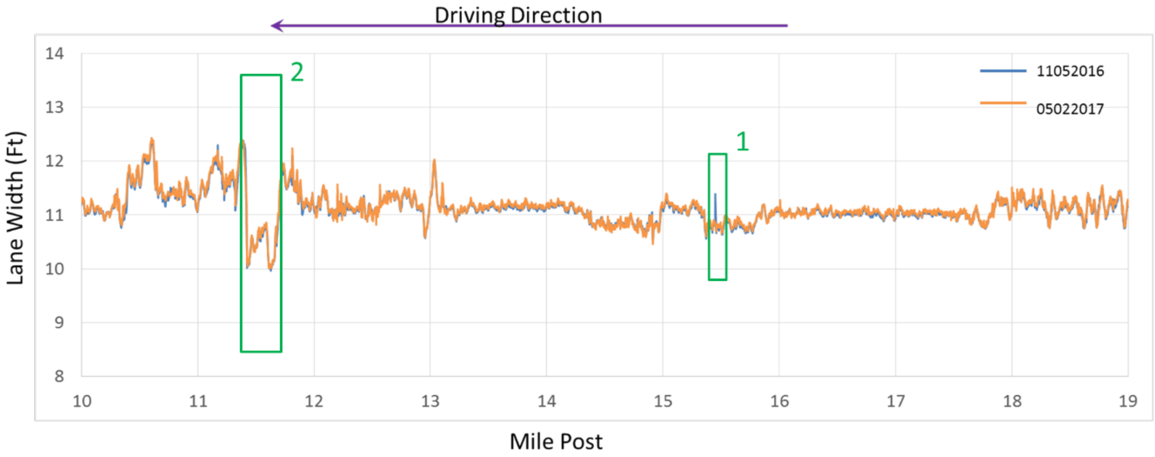
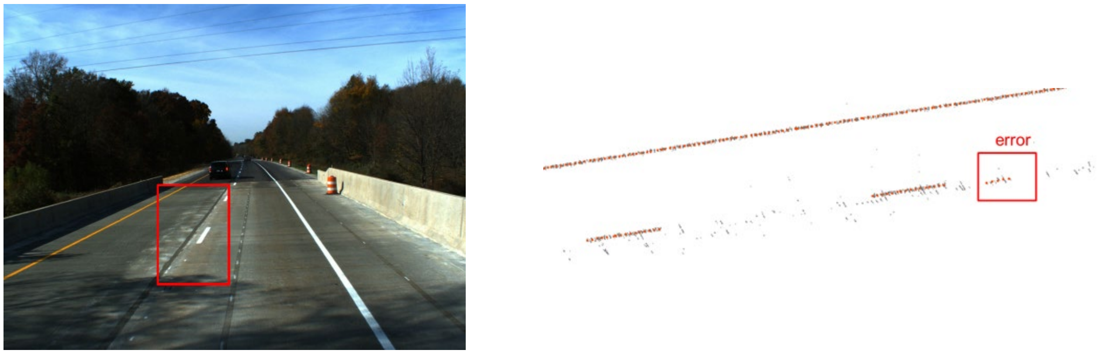


Figure 5.16. Comparison of estimated lane width from two datasets for the interstate highway work zone area



(a)

(b)

Figure 5.17. Anomalies in the 2016/11/05 dataset: (a) RGB image and (b) extracted lane marking points (gray) and derived lane marking centerline (red)

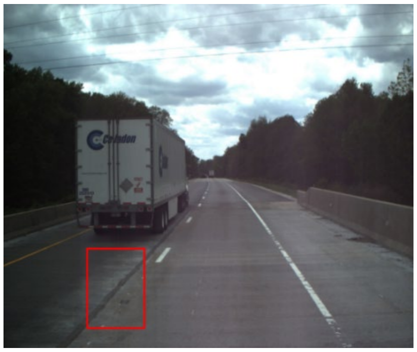


Figure 5.18. Image from the 2017/05/02 dataset for the area that had poor lane markings in the 2016/11/05 dataset (Figure 16(a))

For the U.S. highway segment, the comparisons of lane width estimates from three datasets are shown in Figure 5.19. The quantitative evaluation of the repeatability analysis for the interstate highway and U.S. highway is shown in Table 5.7, which indicates that the RMSEs from the comparison results ranged from 1.49 cm to 3.01 cm, which is acceptable keeping in mind the LiDAR point cloud accuracy obtained from error propagation. The statistical results demonstrate the repeatability of the proposed strategy when using different sensors to collect data in different seasons and the precision of the estimated mounting parameters from the system calibration.

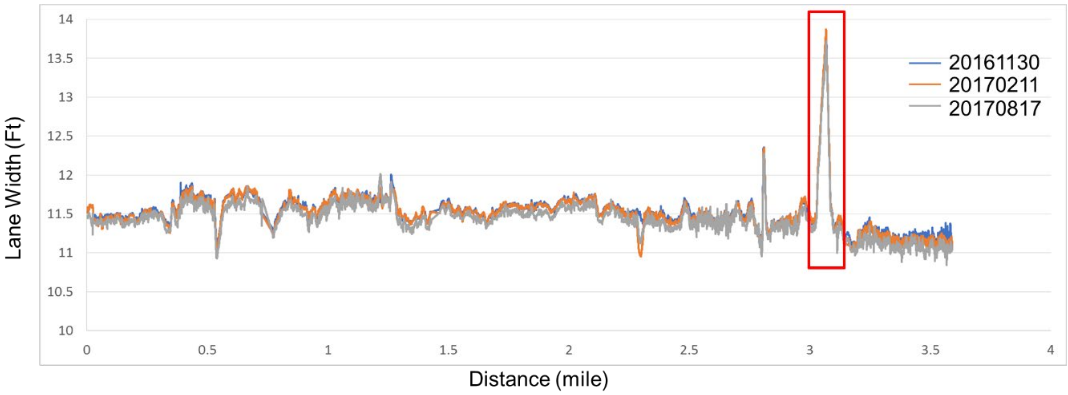
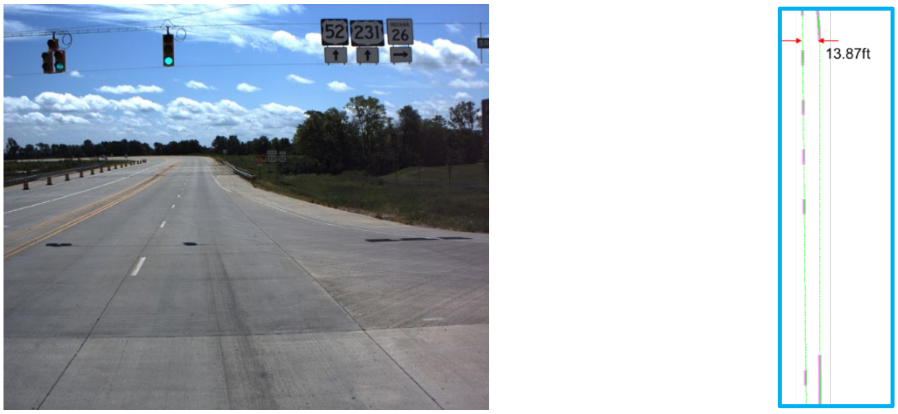


Figure 5.19. Comparison of estimated lane width from three datasets (collected on 2016/11/30, 2017/02/11, and 2017/08/17) for the U.S. highway segment

Table 5.7. Statistics of lane width comparison from interstate highway datasets and U.S. highway datasets

Compared Datasets	**Mean**	**Standard Deviation**	**RMSE**
Interstate Highway			
2016/11/05 & 2017/05/02	1.22 cm (0.040 ft)	1.01 cm (0.036 ft)	1.65 cm (0.054 ft)
U.S. Highway			
2016/11/30 & 2017/02/11	0.85 cm (0.028 ft)	1.22 cm (0.040 ft)	1.49 cm (0.049 ft)
2016/11/30 & 2017/08/17	2.74 cm (0.090 ft)	1.25 cm (0.041 ft)	3.02 cm (0.099 ft)

One should note that usually in the case of an area corresponding to an intersection, there are no lane markings and hence, the lane width is not defined. Moreover, the lane markings when approaching an intersection area would be curved as they are turning to the other road. Therefore, the interpolated centerlines would be erroneous, thus leading to larger lane width estimates. Figure 5.17 depicts such a case, where the spike within the area marked by the red box is taking place at an intersection and this spike occurs for each of the three datasets. Figure 5.20(a) shows the corresponding image and Figure 5.20(b) shows the extracted lane marking points and lane marking centerline after interpolation.



(a)

(b)

Figure 5.20. Road intersection area resulting in the spike in Figure 5.19: (a) the captured image at the intersection, and (b) the extracted lane marking points (pink) and interpolated lane marking centerline (green)

5.5.2.4 Accuracy Analysis of Lane Width Estimation

The main arterial segment was selected for comparing the lane width obtained from the proposed strategy to that from a manually digitized centerline and on-site manual measurements of the corresponding lane width. Figure 5.21 shows the extracted lane markings, their derived centerline, and the average lane width estimate for each dash line using the proposed strategy. To evaluate the accuracy of the derived lane width, the corresponding values from on-site manual measurements was derived by averaging four values: distance between inner bounds, outer bounds, inner and outer bounds, and outer and inner bounds of the dash line and side lane markings, as shown in Figure 5.22. Moreover, the lane width derived from a manually digitized centerline was provided to analyze its difference from the lane width estimates obtained from the proposed strategy as well as the ground truth. The comparison of the estimated lane width from the proposed strategy and the corresponding value from the manually digitized centerline as well as the ground truth for each of the three dash lines is shown in Figure 5.23. The difference among the derived lane widths from the proposed strategy and the ground truth was around 0.1 ft (3.04 cm), which validated the accuracy of the lane width estimates from the proposed strategy and also indicated the accuracy of the mounting parameter estimates from the system calibration. Also, the difference between the lane width obtained from the manually digitized centerline and the ground truth was around 0.04 ft (1.31 cm), which again illustrated the accuracy of the mounting parameter estimates.

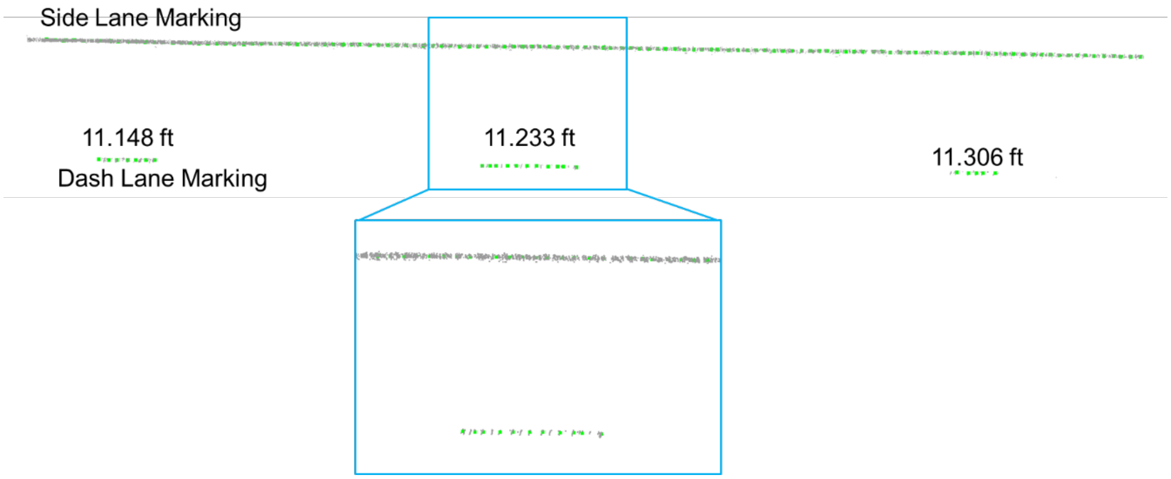


Figure 5.21. Extracted lane marking points (gray) and derived lane marking centerline (green), as well as estimated lane width (unit: ft) using the proposed strategy

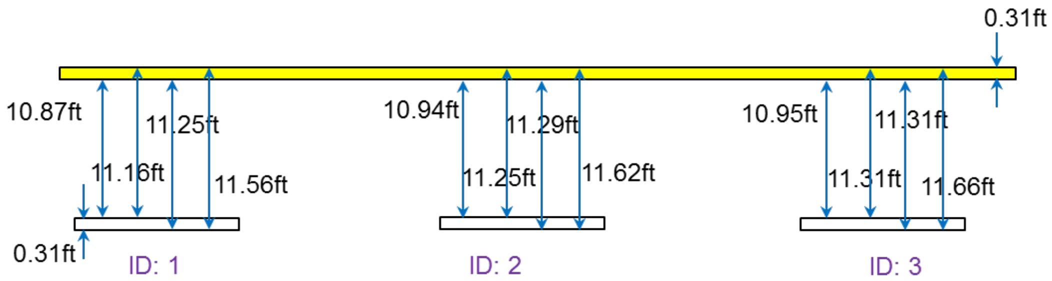


Figure 5.22. On-site manual measurements of lane width for the main arterial segment

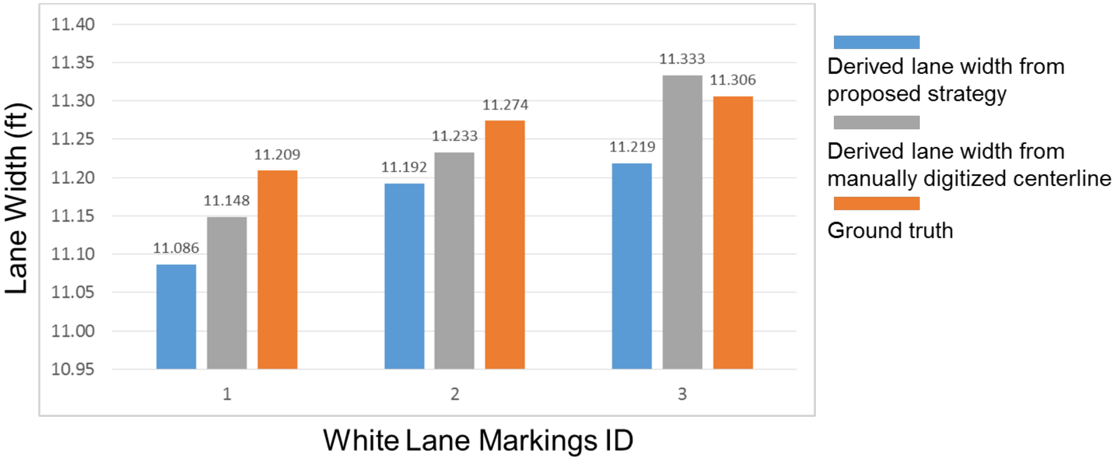


Figure 5.23. Comparison of a derived lane width from the proposed strategy and a manually digitized centerline and a ground truth lane width

5.6 Summary

This chapter introduced a new approach for estimating the lane width along a driving lane from dense point clouds acquired by a LiDAR-based MMS. First presented was the system architecture of an MMS, which consists of several spinning multi-beam laser scanners, a camera, and a GNSS/INS. A LiDAR calibration strategy for deriving the system mounting parameters then was introduced, which was used to extract the road surface along the driving lane with the assistance of vehicle trajectory data. Next, lane marking points were extracted based on the assumption that their LiDAR intensity was higher than those from pavement. A distance-based region growing was applied to cluster the potential candidate lane marking points. The clusters then were partitioned into subgroups and the RANSAC algorithm was used along with the vehicle trajectory information to remove non-lane marking points from the subgroups. Finally, the lane marking centerline was derived from each subgroup for lane width estimation.

To illustrate the performance of the proposed lane width estimation strategy, the results from four different sets of experimental results, which used a total of six datasets for three different road segments surveyed on different dates using different sensors, were analyzed. The first experiment emphasized the importance of accurate system calibration for deriving accurate lane width estimates by observing the impact of variations in the mounting parameters on lane marking extraction and lane width estimation. The second experiment suggested that the VLP16 and HDL32E laser scanners produced compatible estimates for lane width, which also confirmed the accuracy of the mounting parameters derived using the proposed multi-sensor calibration strategy. The results from the third experiment demonstrated the repeatability of the proposed lane width estimation strategy by analyzing the precision of the results from multiple datasets for two road segments. Finally, the fourth experiment validated the accuracy of the lane width estimates obtained using the proposed strategy. The above experiments also indicated the following two scenarios where the proposed strategy results were inaccurate or unexpected. 1) The performance of the proposed lane width estimation strategy relies on an accurate calibration of the MMS used to capture the LiDAR data. This aspect was demonstrated by comparing the lane width estimates obtained from a point cloud reconstructed using calibrated mounting parameters and biased mounting parameters. 2) The proposed strategy was found to result in unexpected lane width estimates in regions where the quality of lane markings was poor or when debris covered the lane markings. However, these unexpected results are favorable for work zone monitoring as they could aid in identifying issues related to low quality lane markings and reporting them to the construction project manager. The precision and accuracy of the results using the proposed strategy were in the range of 3.5 cm, which is acceptable keeping in mind the LiDAR point cloud accuracy obtained from error propagation.

6. CONCLUSION AND RECOMMENDATION FOR FUTURE RESEARCH

6.1 Summary of Contributions

This thesis focused on developing a generic framework for point cloud processing, which includes the initial processing procedures (an adaptive down-sampling strategy and a multi-class segmentation approach) and two applications. One of the most important objectives for the initial processing procedures is to confirm that the proposed approaches are suitable for point clouds from various platforms/resources by considering internal characteristics.

In addition, two applications were carried out in high human activity areas. First, in an urban scene, a building model generation strategy using a point cloud from an ALS dataset. The second application was in a transportation corridors, where a new lane width estimation framework was introduced using the point clouds collected from a homemade mobile mapping system. The contributions of each proposed strategy are summarized below:

First Tier of Data Processing:

An adaptive down-sampling strategy was proposed, which can accomplish the following:

- Removes redundant points from high-density planar regions while retaining points in planar areas with sparse points as well as all the points within the linear/cylindrical and rough neighborhoods.
- Considers the different local characteristics (noise level, LPD) and feature categories (planar, linear/cylindrical, and rough) of a point cloud.
- Introduces two PCA approaches which can identify the characteristics of planar, linear/cylindrical, and rough neighborhoods while considering the noise level within the constituents of a point cloud.

- Estimates accurate LPDs based on the nature of the local surface.
- Utilizes a probability-based test to remove points using an evaluated LPD within the planar areas.
- Can be applied to both laser scanning from various platforms and image-based point clouds.
- Enhances the linear/cylindrical features, especially datasets that have large *noise-level-to-area ratios.*

A new spatial-domain region growing approach for the segmentation of planar, linear/cylindrical, and rough features was proposed, which can perform the following functions:

- Simultaneously partitions a given point cloud into planar, linear, varying-radii cylindrical, and rough segments.
- Can be suitable for multi-platforms/resources and multi-resolution point clouds.
- Proposes a concept of seed region evaluation to reduce the sensitivity of the segmentation outcome to the choice of the seed location.
- Considers the variations in the internal characteristics (noise level, LPD) and feature categories (planar, linear/cylindrical, and rough) of point clouds.

A sequence of QC procedures was proposed to improve and measure the segmentation outcomes by doing the following:

- Identifies several problems that affect the quality of the segmentation outcomes, such as competition among neighboring planar and linear/cylindrical features.
- Proposes practical solutions to improve the segmentation results and provides quantitative measures that reflect the frequency of these problems.

Second Tier of Data Processing:

For urban environments, a building model generation approach was proposed to reconstruct building models from ALS data by doing the following:

- Reconstructs complex building models with several characteristics, which includes right-angle, multi-orientation, and curved boundaries using a single data source (ALS data) without any ancillary data.
- Introduces an ERMBR procedure, which includes RMBR and LSA integration as well as generalized RMBR procedure, to generate more robust right-angle DBMs.
- Proposes a new boundary regularization approach using a breakpoint detection strategy, which reconstructs building models with multi-orientation/curved boundaries.
- Integrates the outcome of GRMBR and BD strategies to generate DBM with right-angle, multi-orientation, and curved boundary.
- Proposes a water-tight strategy, which considers the topology between connected primitives and generates seamless building models

For transportation corridors, an accurate lane width estimation framework was proposed using an in-house developed mobile mapping system, which:

- Includes a comprehensive framework for system development, system calibration, road surface extraction, lane marking identification, lane width estimation, and validation).
- Has no limitations as far as the sensor model used and does not require raw measurements from a laser scanner.
- Has been tested for tens of miles of datasets to demonstrate the robustness of the proposed lane width estimation strategy.

- Has been validated using different sensor models, repeatability tests, and a ground truth dataset to prove its feasibility.

6.2 Recommendations for Future Work

Recommendations/suggestion for future work related to the proposed strategies are as follows:

- The future work of the proposed adaptive down-sampling approach should focus on investigating the performance of other classification approaches. In other words, rather than relying on PCA-based classifications to identify the planar, linear/cylindrical, and rough local neighborhoods of the individual points, other approaches should be investigated that are more able to correctly classify local neighborhoods even in the presence of relatively high noise, which is the case for DIM datasets. Another research effort could be to define appropriate local point density values for the extraction and modeling of planar, linear/cylindrical, and rough features.
- Future research regarding the proposed multi-class simultaneous segmentation and quality control procedures could focus on establishing additional constraints to ensure an even more reliable selection of seed regions. In addition, investigating the application of color/intensity information after accurate geometric and radiometric sensor calibration could improve the segmentation results. Addressing other segmentation problems that could be mitigated through improved QC procedures is another avenue to explore. Finally, the outcomes from the segmentation and QC procedures could be used to make hypotheses regarding the generated segments (e.g., building rooftops, building façades, light poles, road surfaces, trees, and bushes).

- Future work regarding building model generation should consider an adjustment approach for 2D boundary breakpoint detection. Currently, breakpoints of building boundaries are derived from two 1D profiles. In this way, geometry of 2D boundaries are not fully considered in the adjustment procedure so detected breakpoints may not represent real corners of buildings. Moreover, integration of LiDAR data and imagery for building model generation should be focused to improve both the accuracy and the visualization results. Although the proposed strategy considers two building model generation approaches and combined two building models into one, some building models still cannot be generated. The clear edges on the imagery can be used to refine the current results by identifying accurate boundaries. Furthermore, edges composed of small edge segments also could be fitted to the visible boundaries on the imagery.
- Future work in lane width estimation should concentrate on deriving lane widths from the driving lane as well as the neighboring lanes. Currently, there is a requirement to drive in each lane for which the lane width is being estimated. If the proposed strategy is extended to include lane width estimation for neighboring lanes, it would significantly reduce the time and effort required for dataset collection. LiDAR-based lane width estimation can be expanded by incorporating image datasets to make data analysis more efficient as it would aid in relaxing the sensitivity of the proposed strategy on the intensity value threshold. Moreover, the non-road surface point clouds can be used to extract other important work zone objects, such as construction barrels, guard rail, and traffic signs.

REFERENCES

[1] D. Gonzalez-Aguilera, E. Crespo-Matellan, D. Hernandez-Lopez, and P. Rodriguez-Gonzalvez, “Automated urban analysis based on LiDAR-derived building models,” *IEEE Trans. Geosci. Remote Sens.*, vol. 51, no. 3, pp. 1844–1851, 2013.

[2] Y. Alshawabkeh and N. Haala, “Integration of digital photogrammetry and laser scanning for heritage documentation,” *Int. Arch. Photogramm. Remote Sens.*, vol. 35, p. B5, 2004.

[3] Z. Lari and A. Habib, “New approaches for estimating the local point density and its impact on LiDAR data segmentation,” *Photogramm. Eng. Remote Sens.*, vol. 79, no. 2, pp. 195–207, 2013.

[4] N. E. Klepeis *et al.*, “The National Human Activity Pattern Survey (NHAPS): a resource for assessing exposure to environmental pollutants,” *J. Expo. Sci. Environ. Epidemiol.*, vol. 11, no. 3, p. 231, 2001.

[5] C. Fuchs, *Oeepe Survey on 3D City Models. In prep.* for OEEPE publications, 1997.

[6] H. Huang, C. Brenner, and M. Sester, “A generative statistical approach to automatic 3D building roof reconstruction from laser scanning data,” *ISPRS J. Photogramm. Remote Sens.*, vol. 79, pp. 29–43, 2013.

[7] E. Kwak and A. Habib, “Automatic representation and reconstruction of DBM from LiDAR data using Recursive Minimum Bounding Rectangle,” *ISPRS J. Photogramm. Remote Sens.*, vol. 93, pp. 171–191, 2014.

[8] J. Yan, K. Zhang, C. Zhang, S.-C. Chen, and G. Narasimhan, “Automatic construction of 3-D building model from airborne LiDAR data through 2-D snake algorithm,” *IEEE Trans. Geosci. Remote Sens.*, vol. 53, no. 1, pp. 3–14, 2015.

[9] O. Ozturk, K. Ozbay, and H. Yang, “Estimating the impact of work zones on highway safety,” 2014.

[10] FHWA, “Work Zone Facts and Statistics.” [Online]. Available: https://ops.fhwa.dot.gov/wz/resources/facts_stats/safety.htm. [Accessed: 18-Oct-2017].

[11] J. Wang and J. Shan, “Segmentation of LiDAR point clouds for building extraction,” in *American Society for Photogramm. Remote Sens. Annual Conference, Baltimore, MD*, 2009, pp. 9–13.

[12] S. Pu and G. Vosselman, “Automatic extraction of building features from terrestrial laser scanning,” *Int. Arch. Photogramm. Remote Sens. Spat. Inf. Sci.*, vol. 36, no. 5, pp. 25–27, 2006.

[13] A. Holgado-Barco, B. Riveiro, D. González-Aguilera, and P. Arias, “Automatic inventory of road cross-sections from mobile laser scanning system,” *Comput.-Aided Civ. Infrastruct. Eng.*, vol. 32, no. 1, pp. 3–17, 2017.

[14] A. Gruen, “Development and status of image matching in photogrammetry,” *Photogramm. Rec.*, vol. 27, no. 137, pp. 36–57, 2012.

[15] W. K. Pratt, *Introduction to digital image processing*. CRC Press, 2013.

[16] E. Rosten and T. Drummond, “Machine learning for high-speed corner detection,” in *European conference on computer vision*, 2006, pp. 430–443.

[17] D. G. Lowe, “Distinctive image features from scale-invariant keypoints,” *Int. J. Comput. Vis.*, vol. 60, no. 2, pp. 91–110, 2004.

[18] J. Canny, “A computational approach to edge detection,” *IEEE Trans. Pattern Anal. Mach. Intell.*, no. 6, pp. 679–698, 1986.

[19] D. H. Ballard, “Generalizing the Hough transform to detect arbitrary shapes,” *Pattern Recognit.*, vol. 13, no. 2, pp. 111–122, 1981.

[20] I. Suveg and G. Vosselman, “Reconstruction of 3D building models from aerial images and maps,” *ISPRS J. Photogramm. Remote Sens.*, vol. 58, no. 3–4, pp. 202–224, 2004.

[21] Y. Furukawa and J. Ponce, “Accurate, dense, and robust multiview stereopsis,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 32, no. 8, pp. 1362–1376, 2010.

[22] H. Hirschmuller, “Stereo processing by semiglobal matching and mutual information,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 30, no. 2, pp. 328–341, 2008.

[23] N. Haala, “The landscape of dense image matching algorithms,” 2013.

[24] J. Hyyppä, W. Wagner, M. Hollaus, and H. Hyyppä, *Airborne laser scanning*. SAGE Publications Ltd.: London, UK, 2009.

[25] A. F. Habib, R. Zhai, and C. Kim, “Generation of complex polyhedral building models by integrating stereo-aerial imagery and lidar data,” *Photogramm. Eng. Remote Sens.*, vol. 76, no. 5, pp. 609–623, 2010.

[26] K. Novak and J. D. Bossler, “Development and application of the highway mapping system of Ohio State University,” *Photogramm. Rec.*, vol. 15, no. 85, pp. 123–134, 1995.

[27] C. Ellum and N. El-Sheimy, “Land-based mobile mapping systems,” *Photogramm. Eng. Remote Sens.*, vol. 68, no. 1, pp. 13–17, 2002.

[28] I. Puente, H. González-Jorge, J. Martínez-Sánchez, and P. Arias, “Review of mobile mapping and surveying technologies,” *Measurement*, vol. 46, no. 7, pp. 2127–2145, 2013.

[29] K. P. Schwarz and N. Sheimy, “Digital Mobile Mapping Systems—State of the Art and Future Trends,” *Adv. Mob. Mapp. Technol.*, pp. 3–18, 2007.

[30] Z. Ji, M. Song, H. Guan, and Y. Yu, “Accurate and robust registration of high-speed railway viaduct point clouds using closing conditions and external geometric constraints,” *ISPRS J. Photogramm. Remote Sens.*, vol. 106, pp. 55–67, 2015.

[31] Z. Lari and A. Habib, “An adaptive approach for the segmentation and extraction of planar and linear/cylindrical features from laser scanning data,” *ISPRS J. Photogramm. Remote Sens.*, vol. 93, pp. 192–212, 2014.

[32] F. Rottensteiner, G. Sohn, M. Gerke, J. D. Wegner, U. Breitkopf, and J. Jung, “Results of the ISPRS benchmark on urban object detection and 3D building reconstruction,” *ISPRS J. Photogramm. Remote Sens.*, vol. 93, pp. 256–271, 2014.

[33] D. Girardeau-Montaut, “Cloudcompare-open source project,” *OpenSource Proj.*, 2011.

[34] C. Cabo, C. Ordoñez, S. García-Cortés, and J. Martínez, “An algorithm for automatic detection of pole-like street furniture objects from Mobile Laser Scanner point clouds,” *ISPRS J. Photogramm. Remote Sens.*, vol. 87, pp. 47–56, 2014.

[35] R. Schnabel and R. Klein, “Octree-based Point-Cloud Compression.,” *Spbg*, vol. 6, pp. 111–120, 2006.

[36] A. Mandow, J. L. Martínez, A. J. Reina, and J. Morales, “Fast range-independent spherical subsampling of 3D laser scanner points and data reduction performance evaluation for scene registration,” *Pattern Recognit. Lett.*, vol. 31, no. 11, pp. 1239–1250, 2010.

[37] E. Puttonen, M. Lehtomäki, H. Kaartinen, L. Zhu, A. Kukko, and A. Jaakkola, “Improved sampling for terrestrial and mobile laser scanner point cloud data,” *Remote Sens.*, vol. 5, no. 4, pp. 1754–1773, 2013.

[38] F. Rottensteiner and C. Briese, “A new method for building extraction in urban areas from high-resolution LIDAR data,” *Int. Arch. Photogramm. Remote Sens. Spat. Inf. Sci.*, vol. 34, no. 3/A, pp. 295–301, 2002.

[39] G. Forlani, C. Nardinocchi, M. Scaioni, and P. Zingaretti, "Complete classification of raw LIDAR data and 3D reconstruction of buildings," *Pattern Anal. Appl.*, vol. 8, no. 4, pp. 357–374, 2006.

[40] G. Vosselman, B. G. Gorte, G. Sithole, and T. Rabbani, "Recognising structure in laser scanner point clouds," *Int. Arch. Photogramm. Remote Sens. Spat. Inf. Sci.*, vol. 46, no. 8, pp. 33–38, 2004.

[41] M. Al-Durgham and A. Habib, "A framework for the registration and segmentation of heterogeneous LiDAR data," *Photogramm. Eng. Remote Sens.*, vol. 79, no. 2, pp. 135–145, 2013.

[42] T. Rabbani, F. Van Den Heuvel, and G. Vosselmann, "Segmentation of point clouds using smoothness constraint," *Int. Arch. Photogramm. Remote Sens. Spat. Inf. Sci.*, vol. 36, no. 5, pp. 248–253, 2006.

[43] B. Yang and Z. Dong, "A shape-based segmentation method for mobile laser scanning point clouds," *ISPRS J. Photogramm. Remote Sens.*, vol. 81, pp. 19–30, 2013.

[44] T. M. Awwad, Q. Zhu, Z. Du, and Y. Zhang, "An improved segmentation approach for planar surfaces from unstructured 3D point clouds," *Photogramm. Rec.*, vol. 25, no. 129, pp. 5–23, 2010.

[45] S. Filin and N. Pfeifer, "Segmentation of airborne laser scanning data using a slope adaptive neighborhood," *ISPRS J. Photogramm. Remote Sens.*, vol. 60, no. 2, pp. 71–80, 2006.

[46] R. M. Haralick and L. G. Shapiro, *Computer and robot vision*. Addison-wesley, 1992.

[47] J. M. Biosca and J. L. Lerma, "Unsupervised robust planar segmentation of terrestrial laser scanner point clouds based on fuzzy clustering methods," *ISPRS J. Photogramm. Remote Sens.*, vol. 63, no. 1, pp. 84–98, 2008.

[48] C. Heipke, H. Mayer, C. Wiedemann, and O. Jamet, "Evaluation of automatic road extraction," *Int. Arch. Photogramm. Remote Sens.*, vol. 32, no. 3 SECT 4W2, pp. 151–160, 1997.

[49] M. Rutzinger, F. Rottensteiner, and N. Pfeifer, "A comparison of evaluation techniques for building extraction from airborne laser scanning," *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.*, vol. 2, no. 1, pp. 11–20, 2009.

[50] D. Belton, *Classification and Segmentation of 3D Terrestrial Laser Scanner Point Clouds*. Curtin University of Technology., 2008.

[51] A. Nurunnabi, D. Belton, and G. West, "Robust segmentation for multiple planar surface extraction in laser scanning 3D point cloud data," in *Pattern Recognition (ICPR), 2012 21st International Conference on*, 2012, pp. 1367–1370.

[52] Y. Zhang and Z. Zou, "Multiple facade images matching for 3D building modeling," in *Audio, Language and Image Processing (ICALIP), 2012 International Conference on*, 2012, pp. 49–54.

[53] K. Mikolajczyk and C. Schmid, "A performance evaluation of local descriptors," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 27, no. 10, pp. 1615–1630, 2005.

[54] C. Kim, A. Habib, and Y.-C. Chang, "Automatic generation of digital building models for complex structures from LiDAR data," *Int. Arch. Photogramm. Remote Sens.*, vol. 37, no. B4, pp. 456–462, 2008.

[55] K. Zhang, J. Yan, and S.-C. Chen, "Automatic construction of building footprints from airborne LIDAR data," *IEEE Trans. Geosci. Remote Sens.*, vol. 44, no. 9, pp. 2523–2533, 2006.

[56] M. Awrangjeb, C. Zhang, and C. S. Fraser, "Automatic extraction of building roofs using LIDAR data and multispectral imagery," *ISPRS J. Photogramm. Remote Sens.*, vol. 83, pp. 1–18, 2013.

[57] G. Sohn and I. Dowman, "Data fusion of high-resolution satellite imagery and LiDAR data for automatic building extraction," *ISPRS J. Photogramm. Remote Sens.*, vol. 62, no. 1, pp. 43–63, 2007.

[58] L. Cheng *et al.*, "Integration of LiDAR data and optical multi-view images for 3D reconstruction of building roofs," *Opt. Lasers Eng.*, vol. 51, no. 4, pp. 493–502, 2013.

[59] H.-G. Maas and G. Vosselman, "Two algorithms for extracting building models from raw laser altimetry data," *ISPRS J. Photogramm. Remote Sens.*, vol. 54, no. 2–3, pp. 153–163, 1999.

[60] F. Tarsha-Kurdi, T. Landes, P. Grussenmeyer, and M. Koehl, "Model-driven and data-driven approaches using LIDAR data: Analysis and comparison," in *ISPRS Workshop, Photogrammetric Image Analysis (PIA07)*, 2007, pp. 87–92.

[61] A. Henn, G. Gröger, V. Stroh, and L. Plümer, "Model driven reconstruction of roofs from sparse LIDAR point clouds," *ISPRS J. Photogramm. Remote Sens.*, vol. 76, pp. 17–29, 2013.

[62] G. Vosselman, "Building reconstruction using planar faces in very high density height data," *Int. Arch. Photogramm. Remote Sens.*, vol. 32, no. 3; SECT 2W5, pp. 87–94, 1999.

[63] G. Sohn, X. Huang, and V. Tao, "Using a binary space partitioning tree for reconstructing polyhedral building models from airborne lidar data," *Photogramm. Eng. Remote Sens.*, vol. 74, no. 11, pp. 1425–1438, 2008.

[64] W. Faig and T. Widmer, "Automatic building extraction from aerial images," *Proc IAPRS*, vol. 33, p. B7, 2000.

[65] G. Vosselman and H.-G. Maas, *Airborne and terrestrial laser scanning*. CRC, 2010.

[66] N. Demir and E. Baltsavias, "Automated modeling of 3D building roofs using image and LiDAR data," in *Proceedings of the XXII Congress of the International Society for Photogrammetry, Remote Sensing, Melbourne, Australia*, 2012, vol. 25.

[67] Q. Zeng, J. Lai, X. Li, J. Mao, and X. Liu, "Simple building reconstruction from LIDAR point cloud," in *Audio, Language and Image Processing, 2008. ICALIP 2008. International Conference on*, 2008, pp. 1040–1044.

[68] S. Gargoum and K. El-Basyouny, "Automated extraction of road features using LiDAR data: A review of LiDAR applications in transportation," in *Transportation Information and Safety (ICTIS), 2017 4th International Conference on*, 2017, pp. 563–574.

[69] H. Guan, J. Li, S. Cao, and Y. Yu, "Use of mobile LiDAR in road information inventory: A review," *Int. J. Image Data Fusion*, vol. 7, no. 3, pp. 219–242, 2016.

[70] J. Han, D. Kim, M. Lee, and M. Sunwoo, "Enhanced road boundary and obstacle detection using a downward-looking LIDAR sensor," *IEEE Trans. Veh. Technol.*, vol. 61, no. 3, pp. 971–985, 2012.

[71] Y. Kang, C. Roh, S.-B. Suh, and B. Song, "A lidar-based decision-making method for road boundary detection using multiple kalman filters," *IEEE Trans. Ind. Electron.*, vol. 59, no. 11, pp. 4360–4368, 2012.

[72] W. Zhang, "Lidar-based road and road-edge detection," in *Intelligent Vehicles Symposium (IV), 2010 IEEE*, 2010, pp. 845–848.

[73] P. Kumar, C. P. McElhinney, P. Lewis, and T. McCarthy, “An automated algorithm for extracting road edges from terrestrial mobile LiDAR data,” *ISPRS J. Photogramm. Remote Sens.*, vol. 85, pp. 44–55, 2013.

[74] P. Kumar, C. P. McElhinney, P. Lewis, and T. McCarthy, “Automated road markings extraction from mobile laser scanning data,” *Int. J. Appl. Earth Obs. Geoinformation*, vol. 32, pp. 125–137, 2014.

[75] Y. Yu, J. Li, H. Guan, F. Jia, and C. Wang, “Learning hierarchical features for automated extraction of road markings from 3-D mobile LiDAR point clouds,” *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.*, vol. 8, no. 2, pp. 709–726, 2015.

[76] H. Guan, J. Li, Y. Yu, C. Wang, M. Chapman, and B. Yang, “Using mobile laser scanning data for automated extraction of road markings,” *ISPRS J. Photogramm. Remote Sens.*, vol. 87, pp. 93–107, 2014.

[77] M. Cheng, H. Zhang, C. Wang, and J. Li, “Extraction and classification of road markings using mobile laser scanning point clouds,” *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.*, vol. 10, no. 3, pp. 1182–1196, 2017.

[78] B. Yang, Y. Liu, Z. Dong, F. Liang, B. Li, and X. Peng, “3D local feature BKD to extract road information from mobile laser scanning point clouds,” *ISPRS J. Photogramm. Remote Sens.*, vol. 130, pp. 329–343, 2017.

[79] H. Cai and W. Rasdorf, “Modeling road centerlines and predicting lengths in 3-D using LIDAR point cloud and planimetric road centerline data,” *Comput.-Aided Civ. Infrastruct. Eng.*, vol. 23, no. 3, pp. 157–173, 2008.

[80] A. Holgado-Barco, D. Gonzalez-Aguilera, P. Arias-Sanchez, and J. Martinez-Sanchez, “An automated approach to vertical road characterisation using mobile LiDAR systems: Longitudinal profiles and cross-sections,” *ISPRS J. Photogramm. Remote Sens.*, vol. 96, pp. 28–37, 2014.

[81] A. Holgado-Barco, D. González-Aguilera, P. Arias-Sanchez, and J. Martinez-Sanchez, “Semiautomatic extraction of road horizontal alignment from a mobile LiDAR system,” *Comput.-Aided Civ. Infrastruct. Eng.*, vol. 30, no. 3, pp. 217–228, 2015.

[82] J. Wang, Z. Hu, Y. Chen, and Z. Zhang, “Automatic Estimation of Road Slopes and Superelevations Using Point Clouds,” *Photogramm. Eng. Remote Sens.*, vol. 83, no. 3, pp. 217–223, 2017.

[83] J. H. Friedman, J. L. Bentley, and R. A. Finkel, “An algorithm for finding best matches in logarithmic expected time,” *ACM Trans. Math. Softw. TOMS*, vol. 3, no. 3, pp. 209–226, 1977.

[84] J. Demantké, C. Mallet, N. David, and B. Vallet, “Dimensionality based scale selection in 3D lidar point clouds,” *Int. Arch. Photogramm. Remote Sens. Spat. Inf. Sci.*, vol. 38, no. Part 5, p. W12, 2011.

[85] M. M. Al-Durgham, “The Registration and Segmentation of Heterogeneous Laser Scanning Data,” University of Toronto, 2014.

[86] A. Sampath and J. Shan, “Building boundary tracing and regularization from airborne LiDAR point clouds,” *Photogramm. Eng. Remote Sens.*, vol. 73, no. 7, pp. 805–812, 2007.

[87] F. He and A. Habib, “Linear approach for initial recovery of the exterior orientation parameters of randomly captured images by low-cost mobile mapping systems,” *Int. Arch. Photogramm. Remote Sens. Spat. Inf. Sci.*, vol. 40, no. 1, p. 149, 2014.

[88] Z. Lari and A. Habib, “Segmentation-based classification of 3D laser data,” in *Proceedings of American Society of Photogrammetry and Remote Sensing Annual Conference 2012*, 2012.

[89] H. Freeman and R. Shapira, “Determining the minimum-area encasing rectangle for an arbitrary closed curve,” *Commun. ACM*, vol. 18, no. 7, pp. 409–413, 1975.

[90] G. T. Toussaint, “Solving geometric problems with the rotating calipers,” in *Proc. IEEE Melecon*, 1983, vol. 83, p. A10.

[91] W. Förstner and B. P. Wrobel, *Photogrammetric computer vision*. Springer, 2016.

[92] A. F. Habib, E.-M. Kim, and C.-J. Kim, “New methodologies for true orthophoto generation,” *Photogramm. Eng. Remote Sens.*, vol. 73, no. 1, pp. 25–36, 2007.

[93] A. F. Habib, R. Zhai, and C. Kim, “Generation of complex polyhedral building models by integrating stereo-aerial imagery and lidar data,” *Photogramm. Eng. Remote Sens.*, vol. 76, no. 5, pp. 609–623, 2010.

[94] “Velodyne HDL-32E User Manual.” [Online]. Available: http://velodynelidar.com/lidar/products/manual/63-9113%20HDL-32E%20manual_Rev%20E_NOV2012.pdf. [Accessed: 09-Jul-2018].

[95] “Velodyne VLP-16 User Manual.” [Online]. Available: https://usermanual.wiki/Pdf/VLP1620User20Manual20and20Programming20Guide2063924320Rev20A.1338722902.pdf. [Accessed: 09-Jul-2018].

[96] “SPAN-CPT Receiver User Manual.” [Online]. Available: https://www.novatel.com/assets/Documents/Manuals/om-20000122.pdf. [Accessed: 09-Jul-2018].

[97] A. F. Habib, J. Lay, and C. Wong, “Specifications for the Quality Assurance and Quality Control of LIDAR Systems.” [Online]. Available: ttps://engineering.purdue.edu/CE/Academics/Groups/Geomatics/DPRG/files/LIDARErrorPropagation.z601 ip.

[98] R. Ravi, Y.-J. Lin, M. Elbahnasawy, T. Shamseldin, and A. Habib, “Bias Impact Analysis and Calibration of Terrestrial Mobile LiDAR System With Several Spinning Multibeam Laser Scanners,” *IEEE Trans. Geosci. Remote Sens.*, 2018.

[99] A. Aashto, “Policy on geometric design of highways and streets,” *Am. Assoc. State Highw. Transp. Off. Wash. DC*, vol. 1, no. 990, p. 158, 2001.

[100] M. A. Fischler and R. C. Bolles, “Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography,” *Commun. ACM*, vol. 24, no. 6, pp. 381–395, 1981.

[101] T. Toledo and D. Zohar, “Modeling duration of lane changes,” *Transp. Res. Rec. J. Transp. Res. Board*, no. 1999, pp. 71–78, 2007.