AUTOMATIC URBAN MODELLING
USING MOBILE URBAN LIDAR DATA

by

YANI IOANNOU

A thesis submitted to the
School of Computing
in conformity with the requirements for
the degree of Master of Science (Computing)

Queen’s University
Kingston, Ontario, Canada
February 2010

Copyright © Yani Ioannou, 2010
Abstract

Recent advances in Light Detection and Ranging (LIDAR) technology and integration have resulted in vehicle-borne platforms for urban LIDAR scanning, such as Terra-point Inc.’s TITAN® system. Such technology has lead to an explosion in ground LIDAR data. The large size of such mobile urban LIDAR data sets, and the ease at which they may now be collected, has shifted the bottleneck of creating abstract urban models for Geographical Information Systems (GIS) from data collection to data processing. While turning such data into useful models has traditionally relied on human analysis, this is no longer practical.

This thesis outlines a methodology for automatically recovering the necessary information to create abstract urban models from mobile urban LIDAR data using computer vision methods. As an integral part of the methodology, a novel scale-based interest operator is introduced (Difference of Normals) that is efficient enough to process large datasets, while accurately isolating objects of interest in the scene according to real-world parameters.

Finally a novel localized object recognition algorithm is introduced (Local Potential Well Space Embedding), derived from a proven global method for object recognition (Potential Well Space Embedding). The object recognition phase of our methodology is discussed with these two algorithms as a focus.
Acknowledgements

I thank all my colleagues in the robotics and computer vision laboratory for their company and support, with special thanks to Dr. Limin Shang who was always available to bounce ideas off. Without the collaboration of my colleagues, and Limin’s in particular, the ideas presented in this thesis would never have developed beyond their formative stages. I’d also like to thank Michael Belshaw and Beverly Wong, who went out of their way to make living in Kingston an enjoyable experience, rather than a mere necessity, and succeeded. I thank my primary supervisor, Dr. Michael Greenspan, for his invaluable feedback on this thesis and the associated papers, and for accepting me as his student despite the exceptionally poor timing and perhaps his better judgement! I’d also like to thank my co-supervisor, Robin Harrap, for providing tremendous motivation and help, especially in writing this thesis, despite his specialization in a subject far from my own. I thank Terrapoint Inc. for their cooperation and the opportunity to work with such ground breaking technology. I also acknowledge the financial support of the Geoide Network Center of Excellence of NSERC, and the Ontario Centres of Excellence.

Finally, but most importantly, I emphatically thank my Mother for her enormous support. Funding may be fickle but, as she has proven time and again, and continues to exemplify, a Mother’s love and understanding never wavers.
Glossary

2.5D a depth image, strictly structured range data consisting of only x, y and depth coordinates. 4, 18

computer vision the study of algorithms and methods to allow computers or robots to perceive their environment using visual sensors, such as cameras or range data sensors. 2, 6, 9, 10, 12, 14, 15, 17, 19, 23, 29–32, 40, 83

object class recognition recognition of objects belonging to the same general class of objects, e.g. recognizing different types of cars as belonging to the car class. 15, 17, 82, 86

object recognition recognition of objects belonging to a specific set of identical objects, e.g. identifying cars with a specific model, make and year. 3, 9, 11, 12, 14, 15, 17, 19, 23, 24, 26–29, 31, 32, 40, 41, 54, 55, 71, 82, 84–86

point cloud data set consisting only of points. 6, 8, 9, 19–21, 24, 26, 29, 34–36, 44, 51, 52, 54, 55, 71, 79

range data a three-dimensional data set consisting of point positions in space, and obtained by range sensor. 2, 17, 24, 28, 30–32, 36, 38, 54, 85
segmentation to partition a data set into simpler groups, where the divisions themselves are more meaningful and/or easier to analyze. i, 8, 9, 12, 19, 23, 24, 26–29, 31, 32, 40, 54, 55, 68, 71, 84–86

self-occlusion the phenomenon of an object occluding itself in 2D projection or as seen from a specific view point. 18

street furniture a GIS term for fixed objects of interest found on the side of street, e.g. fire hydrants, signs, etc.. 9

unorganized range data range data with a large variance in point density and locations, highly unstructured. 10, 29, 33, 54, 84
Acronyms

**BHT** bounded Hough transform. 17, 42

**DARPA** Defense Advanced Research Projects Agency. 3

**DoG** difference of Gaussians. 12, 29, 30, 76, 86, 87

**DoN** difference of normals. 10–13, 26, 35, 36, 38, 39, 51, 52, 54, 55, 68, 71, 76, 84, 86, 87

**GIS** geographic information systems. 1, 2, 9, 14, 27, 28, 54, 83

**GISc** geographic information science. 1, 2, 10

**GPS** global positioning system. 1, 3, 6, 94, 95

**ICP** iterative closest point. 11, 17, 19–22, 41, 44, 46, 47

**IMU** inertial measurement unit. 5, 6, 94, 95

**LADAR** light detection and ranging. 2

**LIDAR** light detection and ranging. 2–6, 8–14, 17, 18, 24, 26–28, 31, 32, 36, 37, 40, 42, 51, 52, 54, 55, 68, 71, 76, 83–86, 94, 95
LPWSE local potential well space embedding. 10–13, 19, 42, 48, 50, 55, 78, 82, 85, 86

NASA the National Aeronautics and Space Administration. 3

NOAA the National Oceanic and Atmospheric Administration. 3

PCA principle component analysis. 17, 37

PWSE potential well space embedding. 10–12, 19, 41–43, 55, 78, 82, 85, 86

RANSAC random sample consensus. 50, 79, 82

SVD singular value decomposition. 21

UTM Universal Transverse Mercator. 8

VDLSD variable dimensional local shape descriptors. 15–17, 23
Table of Contents

Abstract i

Acknowledgements ii

Glossary iii

Acronyms v

Table of Contents vii

List of Tables xii

List of Figures xiii

Chapter 1:

Introduction 1

1.1 Geographic Information Systems 1

1.2 Light Detection and Ranging 2

1.2.1 Introduction to LIDAR 2

1.2.2 The Evolution of of LIDAR 3

1.2.3 TITAN® 5

1.3 The state of GIS and LIDAR 6
1.4 TITAN® and Computer Vision: Towards Automatically Generating GIS Models ................................................. 9
1.5 Open Problems in GIS .................................................. 10
1.6 Contributions ........................................................... 10
1.7 Thesis Outline ......................................................... 11

Chapter 2:

A Review of 3D Object Recognition ................................. 14
2.1 Object Recognition ..................................................... 14
2.1.1 Feature-based Object Recognition .............................. 15
2.1.2 Spin Images .......................................................... 15
2.1.3 Point Signatures ..................................................... 16
2.1.4 Variable Dimensional Local Shape Descriptors ............... 16
2.1.5 Potential Well Space Embedding .................................. 17
2.2 Object Recognition in LIDAR Data ............................... 17
2.3 Iterative Closest Point Algorithm ................................... 19
2.3.1 ICP Algorithm ...................................................... 20
2.3.2 Complexity .......................................................... 21

Chapter 3:

A Methodology for Object Recognition in Urban LIDAR Data ................................................................. 23
3.1 Motivation .............................................................. 23
3.2 A Different Approach ............................................... 26
### Chapter 4:
**Segmentation of Urban LIDAR Data**

4.1 Interest Operator Background .......................... 29

4.1.1 Difference of Gaussians .................................. 29

4.1.2 Surface Processing ........................................ 30

4.1.3 Scale Space ............................................. 31

4.2 Difference of Normals Operator .......................... 33

4.2.1 Normals as a Scale Space for Surfaces ................. 33

4.2.2 Difference of Normals .................................... 35

4.2.3 Approximating Normals in Range Data .................. 36

4.2.4 Resolving Normal Ambiguity ............................ 38

4.2.5 Segmentation with Difference of Normals ............... 38

### Chapter 5:
**Object Recognition in Segmented Urban LIDAR Data**

5.1 Global Potential Well Space Embedding .................. 41

5.2 Local Potential Well Space Embedding .................... 42

5.2.1 LPWSE Definition ...................................... 44

5.2.2 LPWSE Algorithm ....................................... 46

Feature Generation ............................................ 46

Offline ...................................................... 47

Online ..................................................... 47

5.2.3 Oriented Registration Model ............................ 48

5.3 Classification .............................................. 48

5.3.1 Normalized Distance Vote .............................. 48
Chapter 6:

Experimental Results ........................................ 51
6.1 Segmentation Experiments ............................. 51
   6.1.1 Complete 3D Point Cloud ......................... 52
   6.1.2 TITAN® Urban Mobile LIDAR Data .............. 52
      Multi-Scale Isolation ................................. 55
      Isolation of Street Furniture ....................... 68
      Isolation of Non-Ground Points ..................... 71
      Curb Isolation ...................................... 76
6.2 Object Recognition Experiments ..................... 78
   6.2.1 Motivation ........................................ 78
   6.2.2 Experimental Setup ............................ 78
   6.2.3 Experimental Results ............................ 79
   6.2.4 Applying LPWSE to Mobile Urban LIDAR Data ... 82

Chapter 7:

Conclusion .................................................... 83
7.1 Segmentation ............................................ 84
7.2 Object Recognition .................................... 84
7.3 Methodology .......................................... 85
7.4 Future Work ........................................... 86

Bibliography .................................................. 88
Appendix A:

TITAN® .......................................................... 93
A.0.1 Introduction to TITAN® .............................. 93
A.0.2 TITAN® Post-processing ............................. 95
A.0.3 TITAN® Specification Overview .................. 95

Appendix B:

LIDAR Scenes Used in Experiments ...................... 96

Appendix C:

Supplemental Difference of Normal Segmentation Re-
results .......................................................... 102

Appendix D:

Supplemental Non-Ground Segmentation Results .... 138

Appendix E:

Supplemental Curb Segmentation Results .............. 144
List of Tables

6.1 Object Dimensions .................................................. 80
6.2 Confusion matrix for classification of 50 views after 500 RANSAC verification with 5000 points. .................................................. 81
6.3 Confusion matrix for classification of 50 views after 500 iterations of RANSAC verification with 1250 points. .............................. 81
6.4 Confusion matrix for classification of 50 views after 500 RANSAC verification with 500 points. .................................................. 81
## List of Figures

1.1 Illustration of an airborne LIDAR scanner in operation. Note that a helicopter system provides an accurate 3D scan of objects by measuring distance from the known position of the helicopter, as determined by precise on-board navigation systems. .................................................. 4

1.2 Illustration of the TITAN® terrestrial LIDAR scanner scanning an urban scene. ................................................................. 5

1.3 LIDAR point cloud for King St. East and Princess St, Kingston, ON, Canada: 3606311 points. .................................................. 7

2.1 The nearest neighbour algorithm applied to a simple point set, as in the first step of ICP. .............................................................. 21

3.1 Object Recognition Pipeline for Urban LIDAR Data. .................. 25

4.1 Gaussian Pyramid of a satellite image of an urban scene (©2009 Google Imagery, ©2009 Digital Globe, GeoEye, used in accordance with terms of service). ................................................................. 32
4.2 Illustration of the effects of the support radius on the estimated normal in an urban scene. A larger support radius ($r_3$) averages out small scale variations (high frequency) in the signal, vs. a smaller support radius ($r_1$) which is more effected by such variations. .................. 34

5.1 Illustration of the Local PWSE features, when using 3 spherical shells as registration models, on an example model point cloud. ............. 45

6.2 DoN segmentation results on King St. East and Princess St, Kingston, ON, Canada (3606311 points). ......................... 56

6.4 DoN segmentation results on TITAN scene fragment containing a fire hydrant. .......................................................... 69

6.5 Illustration of the complete process of segmentation of an object from a scene using DoN. ............................................. 70

6.6 Segmented objects of class *car* from TITAN data of Kingston, ONT. 72

6.7 Segmented objects of class *person* from TITAN data of Kingston, ONT. 73

6.8 Segmented objects of class *firehydrant* from TITAN data of Kingston, ONT. .......................................................... 74

6.9 DoN non-ground segmentation results for King St. East and Princess St, Kingston, ON, Canada: 1173513/3606311 points (32%). ....... 75

6.10 DoN curb segmentation results on King St. East and Princess St, Kingston, ON, Canada: 7555/3606311 points (0.20%). ............... 77

6.11 Objects used in LPWSE experiments. ........................................ 78

6.12 Example of a dense model point cloud and view captured of the Rabbit object. ......................................................... 79

6.13 Classification accuracy for a set of 50 randomly sub-sampled views. 80
A.1 The TITAN® terrestrial mobile scanning platform. 93

B.1 LIDAR point cloud for Bagot St, Kingston, ON, Canada: 478377 points. 97
B.2 LIDAR point cloud for Intersection of Aberdeen and William St, Kingston, ON, Canada: 523115 points. 98
B.3 LIDAR point cloud for Intersection of Bagot and Brock St, Kingston, ON, Canada: 484046 points. 99
B.4 LIDAR point cloud for Intersection of Clergy and Johnson St, Kingston, ON, Canada: 614403 points. 100
B.5 LIDAR point cloud for Intersection of Princess and Bagot St, Kingston, ON, Canada: 620820 points. 101

C.1 DoN segmentation results on Bagot St, Kingston, ON, Canada (478377 points). 103
C.3 DoN segmentation results on Intersection of Aberdeen and William St, Kingston, ON, Canada (523115 points). 110
C.5 DoN segmentation results on Intersection of Bagot and Brock St, Kingston, ON, Canada (484046 points). 117
C.7 DoN segmentation results on Intersection of Clergy and Johnson St, Kingston, ON, Canada (614403 points). 124
C.9 DoN segmentation results on Intersection of Princess and Bagot St, Kingston, ON, Canada (620820 points). 131

D.1 DoN non-ground segmentation results for Bagot St, Kingston, ON, Canada: 160989/478377 points (33%). 139
D.2 DoN non-ground segmentation results for Intersection of Aberdeen and
William St, Kingston, ON, Canada: 182334/523115 points (34%). . . 140

D.3 DoN non-ground segmentation results for Intersection of Bagot and
Brock St, Kingston, ON, Canada: 101935/484046 points (21%). . . . 141

D.4 DoN non-ground segmentation results for Intersection of Clergy and
Johnson St, Kingston, ON, Canada: 148287/614403 points (24%). . . 142

D.5 DoN non-ground segmentation results for Intersection of Princess and
Bagot St, Kingston, ON, Canada: 212803/620820 points (34%). . . . 143

E.1 DoN curb segmentation results on Bagot St, Kingston, ON, Canada:
1054/478377 points (0.22%). . . . . . . . . . . . . . . . . . . . . . . . . 145

E.2 DoN curb segmentation results on Intersection of Aberdeen and William
St, Kingston, ON, Canada: 964/523115 points (0.18%). . . . . . . . 146

E.3 DoN curb segmentation results on Intersection of Bagot and Brock St,
Kingston, ON, Canada: 567/484046 points (0.11%). . . . . . . . . . 147

E.4 DoN curb segmentation results on Intersection of Clergy and Johnson
St, Kingston, ON, Canada: 1325/614403 points (0.21%). . . . . . . . 148

E.5 DoN curb segmentation results on Intersection of Princess and Bagot
St, Kingston, ON, Canada: 984/620820 points (0.15%). . . . . . . . 149
Chapter 1

Introduction

1.1 Geographic Information Systems

The goal of understanding our surroundings is limited by the tools with which we measure and discover those surroundings, how we interpret those measurements, and how we record and share our interpretations. Whether it be a measuring stick or the global positioning system (GPS), technology has governed our measurements, as it has the way we record our results, whether it be a cave wall, paper maps or Internet-based repositories. Correspondingly science has helped us interpret the raw measurements and analyze our results.

Geographic information science (GISc) is the study of how to interpret geographic data, the raw information from observations in the real world, and the development of geographic information systems (GIS), which encompass the full cycle described above: measurements, storage, analysis and communication.

As with all schools of research, GISc does not progress in isolation. For the most part GIS tools rely on methods already well understood in computer science.
CHAPTER 1. INTRODUCTION

However, periodically GIS researchers develop tools for measurement that require new CS research and development. This drives development in many other fields including computer science, and in the particular case discussed herein, the research in real-world computer perception, or computer vision.

The engineering of new spatial sensor technology inevitably breeds the need to research new methods in GISc to understand that sensor’s data, and develop new GIS to store, analyze, and display the new geographical information. The recent development of mobile ground-based light detection and ranging (LIDAR) scanners has done just that, and the development of new methods in three-dimensional computer vision will be central to any GISc progress in the analysis, understanding and application of the large and complex data these new scanners provide. In fact, the widespread adoption of LIDAR is currently impeded by the lack of good tools to support data processing.

Here we document our attempt to solve an important part of that problem using existing methods in computer vision and, perhaps more excitingly, developing new methods for assisting with the analysis of LIDAR data for an urban area, that is, for urban LIDAR data.

1.2 Light Detection and Ranging

1.2.1 Introduction to LIDAR

LIDAR, also known as light detection and ranging (LADAR), is a popular sensor technology used for acquiring the raw range data used by many computer vision applications. LIDAR today finds applications in law enforcement in the form of
LIDAR speed detectors, remote sensing in the form of airborne and satellite scanners, and autonomous navigation for robotics such as many of the vehicles in the Defense Advanced Research Projects Agency (DARPA) 2005 Grand Challenge.

For object recognition, in particular, LIDAR’s consistent and accurate 3D readings despite viewing, environmental and lighting conditions make it appealing. LIDAR, like RADAR, is based on using the known speed of light to measure the distance to an object by measuring the time for an electromagnetic signal (in this case light) to reflect off the surface of the object and return to the sensor. Lasers have been used to accurately measure distances (laser ranging) for over 40 years, although the widespread development of semi-automated LIDAR sensors has really taken off only in the last decade.

1.2.2 The Evolution of of LIDAR

Laser ranging was first developed by the National Aeronautics and Space Administration (NASA) in 1964 with the Beacon B satellite [7], and shortly after this first use, laser ranging was used to measure the distance to the moon with the aid of a reflecting mirror placed on its surface. In 1975 the first airborne LIDAR system was tested with the Atmospheric Oceanographic LIDAR (AOL), a joint project between NASA and the National Oceanic and Atmospheric Administration (NOAA). Only with the advent of GPS and inertial measurement systems in the 1990s was a laser measurement system able to scan more than a single line profile of the terrain under an airborne platform, and ultimately enable LIDAR scanning. With airborne LIDAR scanning (such as that illustrated in Figure 1.1), a grid of point locations on the
surface of the earth are sampled, allowing the generation of 3D terrain models, architectural models at the broad scale and the characterization of features such as forest. An airborne system provides an accurate 2.5D view of the scene measuring distance from the known position of the aircraft. However, much of the scene is not visible from the birds-eye vantage point, for example in Figure 1.1 the tree trunk would likely not be scanned. In an urban setting, this poses problems. For example, even ignoring the relatively low resolution of airborne LIDAR data, the sides of buildings are very difficult to capture with airborne LIDAR.

Until relatively recently, LIDAR systems were restricted to airborne and static
CHAPTER 1. INTRODUCTION

ground systems due simply to the size of the scanners and necessary equipment such as the inertial measurement unit (IMU). Recently, however Terrapoint Inc. developed TITAN®, one of the first truly mobile ground LIDAR scanning platforms.

1.2.3 TITAN®

TITAN® is the name of a project by Terrapoint Inc. to develop a practical mobile ground LIDAR scanning platform. TITAN® can drive through an urban environment while continually scanning the scene, as illustrated in Figure 1.2. TITAN® was first developed for the Canadian military, specifically for surveying the road network in Afghanistan. The system has since been further developed to allow TITAN® to offer commercial data in well-developed urban landscapes. The system is based

---

1See Appendix A for details on TITAN®.
on a normal pickup truck, on which is mounted a boom supporting a sensor package including 4 LIDAR units, 4 colour cameras, an IMU, and a highly accurate GPS. On-board computers store both the raw LIDAR and camera data at runtime at highways speeds, and the data is rectified with the IMU and GPS data into one coherent point cloud offline.

TITAN® scans from 10,000 to 20,000 points per second. As a result, at slower speeds TITAN® can scan hundreds of points per square metre, down to 20-40 points per square metre at highway speeds. Unlike airborne scanners, TITAN® has good scanning geometry for scanning vertical surfaces, with an accuracy of 4 – 6 cm. Figure 1.3 shows a small subsection of a TITAN® captured scene. This scene (as with all TITAN data presented in this thesis) was captured by TITAN in the city of Kingston, Ontario, Canada in May 2007. Note that the uneven pattern and artifacts represent multiple scan passes and moving vehicles respectively. TITAN® provides not only a novel method of LIDAR scanning a scene from the ground, but a whole new perspective for computer vision algorithms, and a promising new application for 3D computer vision in particular.

1.3 The state of GIS and LIDAR

The need for accurate 3D models of large-scale infrastructure such as highways and buildings, and small-scale infrastructure such as street furniture - light posts, fire hydrants, post boxes, etc. - was the motivation behind the development of TITAN® by Terrapoint Inc.. However, the final product of an accurate 3D model is far from the raw data TITAN® itself provides, which is an unordered set of points, a point cloud.
Figure 1.3: LIDAR point cloud for King St. East and Princess St, Kingston, ON, Canada: 3606311 points.
After post-processing, which rectifies and combines data from multiple scanners and passes, the data TITAN® provides is a very large point cloud consisting of all the scanned points, in Universal Transverse Mercator (UTM) coordinates and altitude, along with the laser intensity return for each point, and optionally correlated colour information from the 2D colour cameras. Even with all this information, creating a 3D model of any particular object of interest is not easy. As a completely manual process, recognition of objects in a TITAN® scene is relatively easy for a human operator, but the segmentation and modelling of those objects necessary to model a complete urban scene, even as idealized primitives such as a cylinder or plane, proves too time consuming due to the amount of interaction required with the data, as well as the sheer amount of data. Even modelling the data with idealized primitives, such as a cylinder or plane, proves difficult due to the high density and clutter in scenes.

The current method of producing this model is a surprisingly labour intensive process, with little in the way of automation. What little automation does exist is provided by very specialized and expensive software. This software makes the process less tedious by way of providing some minor automation such as finding a cylinder in a scene given a seed line and radius, which then may be used to find pipes or a line following algorithm for helping to segment road lines.

Although this may be feasible for a small segment of road, this methodology does not scale well. The existing automation is too simplistic and low level to help overcome the bottlenecks in processing dense LIDAR data sets. Producing an accurate 3D model of large scale infrastructure, for example a highway network, may not be cost effective and thus limits any useful function that such a model may serve.

Even a relatively small level of automation of the current bottlenecks in LIDAR
processing would reduce the time needed to produce an accurate 3D model from the TITAN® data significantly, and computer vision may provide the solution to that and more.

1.4 TITAN® and Computer Vision: Towards Automatically Generating GIS Models

Computer vision is inherently motivated, and conversely limited, by the sensors and ultimately the data available to it. TITAN®’s LIDAR point clouds provide a whole new motivation to develop computer vision algorithms - they provide a new view of the world unmatched by the previous airborne or static ground incarnations of LIDAR. An application that has often been the focus of computer vision algorithms using LIDAR is creating GIS relevant models of urban infrastructure, and in this application in particular the TITAN® data offers much potential.

One of the simplest yet most important features of urban infrastructure to model is the road network. Historically, LIDAR scanners have been deployed on aerial and satellite platforms, due to their expense and size. Hence road extraction from such data has, at best, been focused on low resolution LIDAR data from a bird’s eye view, and there is much literature on extracting GIS data from such imagery. Some of this literature is reviewed in Chapter 2. While roads are also of interest in TITAN data, smaller scale structure that has not previously been visible in scans from airborne platforms present even more potential.
1.5 Open Problems in GIS

Much of the focus of this thesis will be on segmentation and object recognition of street furniture, simply because this presents an ideal use case for TITAN®. There are several open problems in GIS for which the TITAN® data presents an opportunity for computer vision to solve, where airborne and static ground-based LIDAR platforms could not. These problems include the automatic recognition and localization of street furniture, ground/non-ground segmentation, and ultimately the complete automatic modelling of an urban environment.

1.6 Contributions

In this thesis several significant contributions are made to the fields of 3D computer vision and GISc:

1. A new scale-based operator for range images, difference of normals (DoN) [17], is introduced and evaluated systematically on mobile urban LIDAR data. DoN is based on the concept of scale-space, and scale-space operators widely used in 2D computer vision. To the author’s best knowledge, this work represents the first scale-based operator for unorganized range data.

2. A new local feature for object recognition is introduced, local potential well space embedding (LPWSE) [18], based on an existing global object recognition method, potential well space embedding (PWSE). LPWSE distinguishes itself from existing local methods for object recognition in being designed to be robust in the presence of sparse data, i.e. accurately classify object views on the order of hundreds of points. The classification accuracy of LPWSE on a range of sparse
object views is evaluated, and LPWSE’s standalone application to mobile urban LIDAR data is considered.

3. The application of the DoN operator for scale-based segmentation of urban scenes, allowing accurate and complete segmentation of objects and features in urban range data using real-world parameters, addressing a pressing need in the GISc and LIDAR communities.

4. A novel methodology is proposed for object recognition in large urban range data sets, with DoN scale-based segmentation of the urban scene as an integral step. The potential of this methodology is evaluated on mobile urban LIDAR data, with PWSE as a viable solution to the object recognition step.

5. A new method of non-ground segmentation in urban range data using the DoN operator is introduced and evaluated on mobile urban LIDAR data.

6. A new method of isolating curb points from an urban scene, using the DoN operator and empirically derived thresholds alone, is introduced and evaluated on mobile urban LIDAR data.

1.7 Thesis Outline

This thesis outlines a complete methodology for object recognition in urban LIDAR data, with significant and novel contributions to both the segmentation and object recognition steps. The remaining chapters of this thesis are organized as follows:

Chapter 2: A review of related research in 3D object recognition and the current state of 3D object recognition methods is presented, with emphasis on the basis
of LPWSE, PWSE, along with fundamental concepts related to the implementation of LPWSE and PWSE such as iterative closest point (ICP). In addition, the particular challenges of object recognition in urban mobile LIDAR data are described.

Chapter 3: A methodology for object recognition in urban LIDAR itself is outlined in a top-down approach, with accurate and complete segmentation of the scene as an integral part, motivating segmentation as a key topic of this thesis and distinguishing itself from existing methodologies.

Chapter 4: Details the DoN operator, a significant contribution of this thesis, and a fundamental part of the methodology outlined in Chapter 3. An intuition behind the design of the operator is presented using a 2D scale operator, the difference of Gaussians (DoG), and the concepts of surface processing and scale space theory.

Chapter 5: Further explains the existing PWSE global object recognition method, before detailing a new localized variant of this algorithm, LPWSE, another significant contribution of this thesis.

Chapter 6: Presents the results of the DoN operator introduced in Chapter 4 on real mobile urban LIDAR scenes. Scale-based classification of points is first demonstrated on a complete 3D model, typical of those used in 3D computer vision, before scale-based segmentation of points is demonstrated on the more challenging incomplete mobile urban LIDAR data. Usage of this method for segmentation of street-furniture in an urban scene is demonstrated by finding the scale parameters for a fire hydrant, and then illustrating the use of the
complete methodology as outlined in Chapter 3 for identifying a fire hydrant in a scene. Further examples of segmented urban objects are shown, including people and cars. Finally the localized object recognition algorithm, LPWSE, introduced in Chapter 5, is applied to real 3D views of varying sparsity in order to evaluate its potential for standalone recognition of objects represented by sparse views in a mobile urban LIDAR scene.

**Chapter 7:** Draws conclusions on the feasibility of the overall methodology outlined in Chapter 3 given the DoN segmentation results in Chapter 6. In addition the classification accuracy of LPWSE on sparse data, and its implications for LPWSE’s application to mobile urban LIDAR data are discussed. Finally the most promising strategy for future research into, and implementations of, the methodology outlined in Chapter 3 is proposed.
Chapter 2

A Review of 3D Object Recognition

This chapter reviews the necessary theory required in the application of computer vision to the geographic information systems (GIS) problem of urban modeling, specifically 3D object recognition in urban light detection and ranging (LIDAR). In addition relevant literature to the particular methods explored in this thesis (as presented in Chapter 5) is reviewed.

2.1 Object Recognition

Computer vision is the study of algorithms that allow computers (or equivalently robots) to perceive the world around them, often through 2D visual sensors, but also through 3D sensors. The problem of object recognition is central to perceiving the environment around us, and this task, simple for a human, has been the study of many decades of computer vision researchers. The problem of object recognition is
to recognize a specific object instance in a scene, for example a specific car model in a scene, and additionally locate the object in the scene and determine it’s pose. This is closely related to the object class recognition problem - recognizing an object as belonging to a specific class of objects, for example recognizing all cars in a scene, which despite our intuition is in fact no easier a problem.

### 2.1.1 Feature-based Object Recognition

There are generally two approaches to 3D object recognition in computer vision: feature-based and appearance-based approaches [35]. Feature recognition relies on reliably calculating a discriminating feature given as complete (or complete as possible) view of the object, be it a 2D image or 3D range image of a scene. In contrast, appearance-based approaches are based on the appearance of 3D objects from different views and/or lighting, and attempt to use this to recognize a particular view of the object in a scene. Notable 3D feature-based algorithms include:

- Spin Images [19]
- Point Signatures [4]
- variable dimensional local shape descriptors (VDLSD) [33]

### 2.1.2 Spin Images

Spin images [19] are a local surface feature, proposed to recognize multiple objects in a scene despite clutter and occlusion. For each point on the surface, normals are used to create a local reference frame, about which is ‘spun’ a 2D histogram, histogramming (essentially counting) the intersections of the fixed size histogram with the surface
around the normal. Although a spin image is essentially a 2D image that attempts to describe 3D geometry, the result of this use of a lower dimensional representation is that symmetry in the model can result in identical spin images. The representation does also have it’s advantages: since they are essentially 2D images, they can easily be compressed using standard lossless image compression techniques.

### 2.1.3 Point Signatures

Point signatures are a feature for recognizing 3D free-form surfaces proposed by Chua and Jarvis [4]. A point signature is based on the local geometry around a point, as encoded into a 1D distance profile $d(\theta)$ described by the intersection of a sphere of fixed radius with the surface. The angle of maximum distance is used as the basis for a locally defined frame of reference along with the tangent plane to the point. This allows the point signatures to be rotation and translation invariant.

For a scene, at arbitrarily spaced ‘seed points’ the signature is found, and used to vote for models with similar signature points. This allows efficient verification of the match, by rejecting models with low votes, and ordering the remaining models according to the number of votes received.

### 2.1.4 Variable Dimensional Local Shape Descriptors

VDLSD [33] generalize well-known Local Shape Descriptors (LSDs), such as spin images (see §2.1.2) and point signatures (see §2.1.3), which attempt to encapsulate the local geometry of a point and it’s neighbourhood. The specific representation is determined by systematically and empirically choosing amongst nine basic and several extended local properties (up to a possible total of 22) to form a high-dimensional
set of descriptors.

VDLSD achieves more reliable point correspondences than spin images and point signatures. It is proposed that the view independent properties of these LSDs are in fact just a subset of the properties generated by finding the principle component analysis (PCA) for each point, as used by VDLSD.

2.1.5 Potential Well Space Embedding

Shang et al. introduced a novel algorithm for pose determination of objects in sparse range data using a registration error metric, such as the iterative closest point (ICP) algorithm (see §2.3), called a Potential Well Space Embedding [29]. The algorithm exploits the local minima of the registration error surface. More commonly regarded as a weakness of any registration algorithm, and to be avoided, instead the local minima are used to create feature vectors which are used to generate multiple hypotheses of the pose, which are then tracked by the bounded Hough transform (BHT) [31]. The method is highly robust to sparsity, and has also been shown to be useful for object class recognition [30]. The algorithm is presented in detail in §5.1.

2.2 Object Recognition in LIDAR Data

As LIDAR units decrease in cost and size, LIDAR is becoming an increasingly popular sensor for computer vision applications. For object recognition, in particular, LIDAR’s consistent and accurate readings despite viewing and environmental effects make it very appealing\(^1\). Despite LIDAR’s advantages over other sensors, there are still many challenges in object recognition using LIDAR data:

\(^1\)For a comprehensive yet accessible guide to LIDAR in object recognition see [13].
CHAPTER 2. A REVIEW OF 3D OBJECT RECOGNITION

2.5D Projection the data returned by a stationary ground based LIDAR sensor is essentially a depth image (2.5D), rather than a full 3D view of the scene. A 2.5D image, like a photograph of a scene, can only capture information about the scene from the current viewpoint. Any object in the scene that exhibits self-occlusion, or is occluded by some other object in the scene will not be completely captured.

Surface Re-sampling two scans of the same object will likely never share the same sampling grid. Instead each sampling grid has a slightly different offset for each scan. The higher the resolution of the scan, the less of an issue this is.

Number of Pixels-on-Target the farther an object is from the sensor, the more sparse the sampling grid will be, and hence the fewer the points per unit surface. This is an issue of scale.

Occlusion an object may not be fully viewable in the scene, and thus we must often recognize an object with an incomplete view.

Computation while LIDAR data is relatively accurate and consistent, the raw output, a very large point cloud, ensures a proportionately large amount of computing power is required to perform any but the most trivial operations.

In addition, working with TITAN® LIDAR data has it’s own set of challenges. Although a TITAN® scan, as a combination of many 2.5D scans, may provide more coverage than a depth image captured from a single point-of-view, it will usually produce a full 3D scan of an object. Any objects at the edge of a scan, or out of sight of TITAN®’s viewpoint (e.g. roofs), will not be covered. Thus self-occlusion is still a significant problem.
TITAN’s point cloud for a specific object is much sparser (hundreds of points) than the typical scene used in object recognition. TITAN’s point cloud for the entire scene is very large (easily in the millions of points). Finding a small object in a massive data set suggests intelligently segmenting the data before performing object recognition. Identifying a sparsely sampled object in the scene additionally suggests the use of an object recognition algorithm that can accurately recognize an object in sparse data. When combining both these challenges - identifying a sparsely sampled object in a massive data set, and segmenting objects accurately from such data - no existing segmentation or object recognition algorithm is alone suited to solve the problem.

### 2.3 Iterative Closest Point Algorithm

The **Iterative Closest Point algorithm** (ICP) is a simple yet powerful registration algorithm popular in computer vision first described by Besl et al. [3]. It can estimate the relative *pose* (transformation and rotation) between a pair of 3D point clouds (or meshes) with the assumption that the given two point clouds are relatively close. It is often used by an object recognition algorithm to refine a pose determination, however it is presented in this section due to the usage of ICP by both the potential well space embedding (PWSE) and local potential well space embedding (LPWSE) algorithms as an integral part of object recognition itself.

ICP is an iterative algorithm that has provably monotonic convergence on a solution. It iteratively minimizes the global mean-square distance metric. The original paper used the point-to-point distance between each point and it’s correspondence (closest) point, however a later improvement on ICP described by Chen et al. [39] used
a point-to-plane distance, that is the distance between each point and it’s correspondence’s tangent plane. ICP using a point-to-plane metric has a faster convergence rate in practice despite each iteration being slower than that of the point-to-point metric.

2.3.1 ICP Algorithm

The ICP algorithm accepts two point clouds (or equivalently meshes) as input, $X$ the set of model points and $P$ the set of data points representing the scene. It iteratively minimizes the distance measure:

$$\sum_i \|X_i - (RP_i + T)\|^2,$$

where $X_i$ is the $i$th model point, $P_i$ is the corresponding $i$th data point, $R$ is the rotation matrix and $T$ is the translation matrix.

Assume we are given two point clouds, the data set $P$ with $N_p$ points, and the model set $X$ with $N_x$ points. The ICP algorithm can then be stated for each iteration:

1. Compute the closest (correspondance) point to each point $P_i$ in the data set, $X_i = C(P_i, X)$, where $C(P_i, X)$ is a nearest neighbour search,

2. Calculate the registration between the two sets, i.e. the transformation $q_k = (R, T)$ with mean square distance $d_k$.

3. Apply the new registration to the original data set, $P^{k+1} = q_k(P^0)$.

4. Terminate when the change in mean square distance is less than a threshold $\tau$, i.e. $\Delta\tau = d_k - d_{k-1} < \tau$, or a maximum number of iterations is exceeded.
Figure 2.1: The nearest neighbour algorithm applied to a simple point set, as in the first step of ICP.

### 2.3.2 Complexity

We can outline the complexity for each of the major steps:

1. Nearest neighbours search - traditionally $O(N_p N_x)$ worst case, as Besl et al. described the ICP algorithm originally, which is essentially brute-force. However using a k-d tree [2] (or similar space partitioning technique) $O(N_p \log N_x)$ is possible. Building a static k-d tree is also $O(N_p \log N_x)$ (assuming a linear median-finding algorithm is used [5]).

2. Registration - typically singular value decomposition (singular value decomposition (SVD)) is used to recover the transformation between the two point clouds and is bounded in the worst case by $O(N_p)$.

3. Apply the new registration to the original data set $O(N_p)$. 
The bottleneck in ICP is the nearest neighbour search, therefore overall ICP is of a complexity \( O(N_p \log N_x) \) when using a k-d tree.
Chapter 3

A Methodology for Object Recognition in Urban LIDAR Data

3.1 Motivation

In 2D computer vision, modern object recognition algorithms generate features based on a segmented image of some form. Often this segmentation is based on some form of point salience, for example edge points, or in the simplest case a linear pixel threshold. In the weakest case, segmentation of the image provides better performance compared to a brute force search for strong features. In the strongest case it allows the feasibility of global methods, which require complete and accurate segmentation of objects from the scene.

In 3D computer vision, salience operators are not a new development either. To cite just one example, VDLSD uses the ratio of the largest and smallest eigenvectors [33] to avoid generating points for relatively flat regions. However, segmentation of points based on real-world parameters of interest, such as object size/scale is not as
In approaching the problem of object recognition in large data sets, such as urban LIDAR, the naive approach is to use object recognition methods directly. Few of the current object recognition methods were developed with the application of processing point clouds on the orders of hundreds of millions of points in mind, hence this approach is too slow (in the case of local features), and simply not feasible without some form of segmentation (in the case of global features).

The approach of previous research on this problem has however been to first find features in the entire data set, with little or no pre-segmentation based on real-world parameters. This approach is exemplified by Golovinskiy et al. [10]. Their approach is to first generate features in the entire data set, identify objects using these features, and finally use a region growing algorithm to segment those objects from the scene.

For the application of object recognition in urban LIDAR data, a clear parameter for segmentation is scale. We are often only interested in objects of a distinct range of scales, and urban LIDAR scenes exhibit a much more diverse range of scales than typical range data. For example, if we wish to recognize fire hydrants in an urban LIDAR scene, we are not interested in points belonging to roads, building and ground surfaces which are very distinct in scale from the relatively small scale of fire hydrants.

We believe the typical approach is in fact the incorrect approach, and instead segmentation should be the first goal in processing large urban LIDAR data sets rather than the last.
Figure 3.1: Object Recognition Pipeline for Urban LIDAR Data.
3.2 A Different Approach

We envision an object recognition pipeline for urban LIDAR data that exploits this intuitive notion of object scale for segmentation of a scene before object recognition. Not only does this pipeline drastically increase the potential for the performance of object recognition using local features, but it also opens up the possibility of using global object recognition methods on segmented data, as will be demonstrated in our segmentation results in Chapter 6.1. Figure 3.1 illustrates the steps in this object recognition pipeline, along with examples and a representative point cloud size at each step in the segmentation process.

The first step is to filter the scene using the Difference of Normals operator, as explained in Chapter 4. In this context the difference of normals (DoN) acts as an interest operator and isolates points in the scene belonging to objects of a narrow range of scales.

This allows the second step, which is the automatic segmentation of each isolated object. Since the Difference of Normals operator leaves the objects quite distinct, this is possible by a 3D connected components algorithm, for example a voxel-based approach as used by Gorte et al. [12] or a clustering/region growing method as used by Golovinskiy et al. [10].

The third step is object recognition on the individual components of the scene. Since segmentation is close to complete, this may be achieved by a local feature, as described in Chapter 5, or a global method as outlined in Chapter 2, and further explained in Chapter 5.

The final step in a full system would be the use of these recognized objects’ pose and positions in modelling the urban scene in a more abstract manner suitable for a
GIS system.

Although we outline the entire methodology, this thesis proposes solutions to the hardest steps in the methodology, that is the segmentation problem and object recognition, while proposing existing (and proven) solutions to the remaining required steps. Connected components in 3D using voxels is an algorithm based on the well understood classic 2D solution [12], while the modelling of the urban scene is relatively straightforward given the objects, pose and positions as provided by either Potential Well Space Embedding, Local Potential Well Space Embedding or indeed any object recognition algorithm that will provide accurate results on sparse unstructured urban LIDAR data.

Thus this thesis provides a complete and novel methodology for object recognition in large urban LIDAR data, with the expectation that interested parties may develop a practical application based upon it.
Chapter 4

Segmentation of Urban LIDAR Data

With urban LIDAR scenes typically consisting of millions of points, extracting even the most efficient of object recognition features over the entire scene is prohibitively expensive. With a focus on the recognition of street furniture in GIS (i.e. lamp posts, fire hydrants, curbs) the vast majority of the points in scene (e.g. road surface, pavement, etc.) are irrelevant to the task.

Human perception easily distinguishes between the relatively uniform ground, street and building points and smaller scale structures (including street furniture) in the scene (refer to Figure 1.3 to demonstrate). The question arises, is there a segmentation algorithm that could efficiently distinguish between points belonging to structures of different scales in range data? Here we present such an algorithm, but before we describe the algorithm itself, an intuition of the motivation behind it is presented through the concepts of scale space and surface processing.
4.1 Interest Operator Background

The difference of Gaussians (DoG) operator has been used extensively in image processing as an edge detector [26], blob detector and salience operator. It has been used in 2D computer vision in numerous applications, such as a blob/edge detector to find points of salience and pre-segment images, and perhaps most notably in the form of a Difference of Gaussian pyramid for scale invariance in object recognition [25]. Although it easily generalizes to 3D structured data, such as volumetric images [38], for unorganized range data there is a distinct lack of such an operator, despite the need for efficient 3D edge detection methods, fast scale-based segmentation and accurate salience operators.

An operator of similar function and efficiency to the DoG for unorganized range data, in the form of point clouds or meshes, is introduced. Furthermore its application to 3D edge detection and scale-based segmentation in real 3D data is demonstrated.

4.1.1 Difference of Gaussians

The DoG operator [11] used in signal processing (and correspondingly image processing) is simply the subtraction of a wide Gaussian from a narrow Gaussian. In the continuous 2-dimensional case, centred on the origin, it may be defined:

\[
f(u, v, \sigma_1, \sigma_2) = \frac{1}{2\pi\sigma_1^2} \exp\left(-\frac{u^2+u^2}{2\sigma_1^2}\right) - \frac{1}{2\pi\sigma_2^2} \exp\left(-\frac{u^2+u^2}{2\sigma_2^2}\right), \tag{4.1}\]

where \(u, v\) are the 2D coordinates, and \(\sigma_1 < \sigma_2\) are the standard deviations of the two Gaussians.
The DoG is an approximation to the Laplacian of the Gaussian, i.e. the second derivative of the Gaussian function. It is, in effect, a bandpass filter [11]. A band-pass filter only preserves the frequencies in a signal/an image in a certain narrow band of frequencies.

4.1.2 Surface Processing

There is a tendency in computer vision to regard range data as simply a collection of points with 3D spatial coordinates. Alternatively, just as 2D images can be argued to be 3D (two spatial coordinates, one intensity), range data, as a discrete sampling of a 3D surface, may be argued to be $(3 + n)$-dimensional (3 spatial coordinates, n-dimensional surface descriptor vector).

Tasdizen et al. [34] proposed the concept of surface processing, for application to surface simplification of meshes. A generalization of a high-boost surface filter to surface processing was demonstrated, amplifying high frequency surface structures. Similarly, we will use an approximation to a frequency-band filter for surfaces.

In surface processing, the surface descriptor typically considered is the unit normal $\mathbf{n}(\mathbf{p})$ at each point $\mathbf{p}$. Together with the point coordinates these normals form a normal map, i.e. each point is described by 6-dimensional vector which may be processed in a similar manner to the pixel map of a 2D image.

There are of course differences between image processing and surface processing. The most obvious is that while an image has only one non-spatial element (i.e. pixels) a surface is described as a vector map, and so any surface processing result will necessarily also be a vector map. In the case of using normals as a surface descriptor, for example, each spatial coordinate will have associated with it a 3D vector.
CHAPTER 4. SEGMENTATION OF URBAN LIDAR DATA

Another key difference is the fact that 2D images are normally highly structured and regularly sampled (i.e., a grid of pixels), and although this is the case in some range data (e.g., Magnetic Resonance Imaging volume images, as illustrated in [34]), most range data is unorganized. For example, a LIDAR scanner’s sampling density depends on the distance of the target surface from the scanner. In addition, most LIDAR scenes of any significant size are a registration of multiple scans, and so have densities that vary irregularly and often suddenly, even across individual scanned features.

In a 2D image, the variation of neighbouring pixels in a real-world image (as opposed to an image of white noise for example) is bounded, although high variations can be produced by lighting conditions, object boundaries (i.e., edges) and scale transitions. In real-world range data there is a much higher coupling between a normal and its spatial coordinates, since the change in the surface between sampled points is bounded by the distance between those points on a continuous smooth surface (i.e., finite surface curvature); however, high variations are found on object boundaries.

4.1.3 Scale Space

In 2D computer vision, scale space [23] is a common concept used in segmentation and object recognition [25]. Due to the nature of 2D imagery, recognizing scale, specifically scale invariance, is vital to 2D object recognition, as two identical objects at a different distance from the camera will have a very different projected image.

In 3D computer vision scale is often an ignored concept. Since scale is preserved by 3D sensors (instead sampling density, noise characteristics, etc. change as a function
of object distance), no special handling is needed to discover the scale of an object in the scene. In fact scale invariance is to be avoided in a 3D object recognition algorithm, as objects of different sizes in range data are always of different scale. The typical scene encountered in 3D computer vision research consists of a cluttered scene of a few objects all of roughly similar scales, whilst that of urban LIDAR consists of many objects at distinct and very different scales. The concept of utilizing scale for segmentation of an urban scene can most easily be related in 2D, by a Gaussian pyramid, as illustrated using satellite imagery of a typical urban scene in Figure 4.1.

This Gaussian pyramid is generated by repeatedly applying a Gaussian filter (i.e. Gaussian blur) with a fixed standard deviation to the image, while sub-sampling the
image to half the size at each level. The resulting pyramid isolates objects in the scene on different scales. With repeated applications of the Gaussian, small-scale structure such as the cars in Figure 4.1 soon disappear, and large scale structures such as roads, and eventually only land usage (i.e. paved vs non-paved) is visible.

Large scale structure in a 2D image is low-frequency, while small scale structure is high-frequency. The Gaussian is a low-pass filter, thus at each scale in the Gaussian pyramid the higher frequencies in the image are removed, with the effect that the lower frequencies in the image are further isolated.

4.2 Difference of Normals Operator

4.2.1 Normals as a Scale Space for Surfaces

A scale space like operator for the implicit surfaces in unorganized range data can be defined simply by exploiting the method of normal estimation used for 3D point clouds. Although there are many different methods of estimating normals (see §4.2.3), normals are always estimated with a support radius (or correspondingly a fixed number of neighbours in structured data). This support radius determines what minimum scale surface structure the normal is representing. For normals estimated with a small support radius, they will be highly affected by noise and small-scale surface structure. On the other hand normals estimated with a large support radius will be effected less by noise and small-scale structure, and more by large-scale surface structures. Figure 4.2 illustrates this effect in 1D.

Suppose we define a scale operator for surfaces,

\[ L(p, r) = \hat{n}(p, r) \]  

(4.2)
with scale parameter $r$, effected by the normal map of a point cloud $P$ estimated with support radius $r$.

Just as described by the basic intuitive scale space axiom defined by Lindeburg et al. [23], the effect of the normals on the implicit surface sampled by a point cloud is to suppress most of the structures in the surface with a characteristic dimension of less than $r$. Furthermore with increasing values of the scale parameter $r$, fine scale surface structure is increasingly suppressed. Despite this, Equation 4.2 does not satisfy all scale space axioms originally outlined by Witkin et al. [36] and more recently enumerated by Lindeburg et al. [23]. One of the scale space axioms in particular that Equation 4.2 violates is the causality requirement introduced by Koenderink et al. [20], who also proved the only filter to satisfy all these axioms (in the continuous case) is the Gaussian.

An implementation of a surface scale space operator for which all scale space axioms hold, might smooth the implicit surface by the weighted sum of the normals in a given support radius, where the weights are sampled from the sampled continuous
Gaussian function. Lindeberg et al. [22] have shown however that the sampled Gaussian kernel does not satisfy the scale space axioms for such an implementation, and instead introduced the discrete Gaussian kernel. Thus a scale operator for unorganized range images sampling surfaces, that satisfies all the scale space axioms, might be defined,

\[ L_T(p, t) = \sum_{x \in P} \hat{n}(x)T(p, x, t), \quad (4.3) \]

where the discrete Gaussian is defined,

\[ T(p, x, t) = \exp^{-t I_{\|p - x\|}(t)}, \quad (4.4) \]

and \( I_{\|p - x\|} \) represents the modified Bessel functions of integer order.

In practice however, the author believes the high time complexity of Equation 4.3 as compared to Equation 4.2 does not justify it’s application, particularly given the results of the latter as shown in §6.1.

### 4.2.2 Difference of Normals

In §4.2.1 it was shown that normals estimated with a specific radius share some similarities with a scale space operator. Therefore simply subtracting two normals of different support radius may in effect approximate a band-filter, similar to how subtracting two Gaussians in 2D images of different support radii effects a band-filter. The DoN operator \( \Delta_{\hat{n}} \) [17] may thus be stated for any point \( p \) in a point cloud \( P \),

\[ \Delta_{\hat{n}}(p, r_1, r_2) = \frac{1}{2} (\hat{n}(p, r_1) - \hat{n}(p, r_2)), \]

where \( r_1, r_2 \in \mathbb{R}, r_1 < r_2. \quad (4.5) \]
Since the DoN is the sum of two unit vectors, it may be normalized simply by halving the difference of the two normals. The result is a vector normal difference map for all points in the scene. These normal differences may be thresholded based on magnitude, i.e. $\Delta \hat{n}(p) = \|\Delta \hat{n}(p)\|$ and/or component value, i.e. $\Delta x\hat{n}(p), \Delta y\hat{n}(p)$, and $\Delta z\hat{n}(p)$. Since the normals, and therefore the normal differences are normalized (i.e. have a magnitude between 0 and 1 inclusive), thresholds are scene independent.

Given the two normal maps estimated with support radii $r_1, r_2$ for a scene, calculating the DoN is $O(N)$ where $N$ is the number of points in the point set $P$.

### 4.2.3 Approximating Normals in Range Data

There are many methods for estimating normals in point clouds (or equivalently tangent planes) [14, 28, 27, 15, 1]. However only those using a fixed support radius, rather than a fixed number of neighbours, are suitable for point clouds with unorganized data, especially when the point cloud density is highly variable. Applying a method based on a fixed number of neighbours to a point cloud with a high variability in sampling density, e.g. urban LIDAR data, results in each normal being computed using what may be a very different support radius, and thus the estimated normals at each point will represent the surface at very different scales. Such normals would be unsuitable for DoN calculations.

In our experiments we estimated normals by finding the tangent plane using the principal components of a local neighbourhood of fixed support radius around each point. This neighbourhood may contain any number of points. The result was that all the normals in the scene were calculated at the same scale. However, due to the highly variable sampling density/resolution of some range data, the accuracy of the
normal estimate may vary considerably across a scene.

Formally we can estimate the unit normal to each point with a support radius \( r \),

\[
\hat{n}(p, r) = \hat{e}_3, \tag{4.6}
\]

where \( \hat{e}_3 \) is the eigenvector associated with the smallest eigenvalue of the local neighbourhood, i.e. the direction of least change, found by PCA,

\[
\{\hat{e}_1, \hat{e}_2, \hat{e}_3\} = \text{PCA}(\{x \mid x \in P, \|x - p\| \leq r\}). \tag{4.7}
\]

The complexity of this method of normal approximation is determined by the nearest neighbours algorithm used to find the neighbourhood of each point. Nearest neighbours implemented using a k-d tree [2] requires \( O(N \log N) \) to build the k-d tree (assuming a linear median-finding algorithm is used [5]), and \( O(\log N) \) to search the tree, where \( N \) is the number of points in the point set \( P \).

As the radius of the neighbourhood increases, the number of points within that neighbourhood increases exponentially\(^1\), and so too does the time required to find all the points within that neighbourhood. In practice however, the very large number of neighbours found in mobile urban LIDAR data with large radii is far more than required to accurately find the normal of a point. Thus by running nearest neighbours on a random (uniform) sub-sample of the data for large radii, we can effectively reduce the runtime of normal approximation for large radii to be the same as with small radii.

---

\(^1\)Generally the number of points in a neighbourhood is proportional to the area of the neighbourhood, although given that LIDAR data is highly unstructured and variable in density this may not hold for all cases.
4.2.4 Resolving Normal Ambiguity

All surface normals for a particular point should have an angle of orientation within the same quadrant regardless of their support radius. However, any normals estimated on range data will exhibit an ambiguity in one axis: for any tangent plane to a point there are two equal but opposite (negative or positive) normals, either of which is mathematically valid. This normal ambiguity is typically resolved with the sensor context; the correct normal is always that pointing in the hemisphere towards the range sensor. For our application, however, which normal we disambiguate to is of no consequence, since the difference of the normals is unaffected as long as the two normals for the point are disambiguated consistently, i.e. in the same direction.

Therefore without loss of generality we can disambiguate the normals for a point \( p \), such that they are both in the same octant of 3-space. Which octant in particular we disambiguate to is inconsequential. Thus disambiguation of the normals can simply be achieved by negating one of the normals if the angle between the two normals is greater than \( \frac{\pi}{2} \), under the assumption that both true surface normals must be within an angle of \( \frac{\pi}{2} \) of each other, since the normals are for a surface sampled from a self-occluded view. In some applications it was found this normal ambiguity can in fact be used to an advantage, as seen in §6.1.2.

4.2.5 Segmentation with Difference of Normals

It is important to note that although the DoN algorithm may provide points of interest, and in doing so isolate objects of interest in the scene, it is by no means in and of itself a segmentation algorithm. A segmentation algorithm must segment individual set of points (e.g. sets of points belonging to objects), from the scene. An
interest operator such as DoN is a key part of a segmentation algorithm, and in the case of urban LIDAR, perhaps the most difficult.

A segmentation algorithm using the DoN operator as an interest operator may be accomplished simply by using a 3D connected components algorithm, for example a voxel-based approach as used by Gorte et al. [12] or a clustering/region growing method as used by Golovinskiy et al. [10].
Chapter 5

Object Recognition in Segmented Urban LIDAR Data

This chapter will explain the methods and algorithms used to solve the problem of recognizing urban infrastructure using computer vision and the TITAN® LIDAR scanner.

In a typical urban LIDAR scene, sparsity renders most current 3D object recognition algorithms ineffective, even with complete segmentation of the object from the scene. Global methods would be expected to perform best on dense segmented data, however they are typically very sensitive to any segmentation outliers. A localized version of a global method known to work well with sparse data is instead proposed as a more tolerant alternative in the presence of imperfectly segmented objects in urban scenes.
5.1 Global Potential Well Space Embedding

A global method that has recently shown promise for sparse object recognition is PWSE [29]. PWSE exploits the distinctiveness of the registration error surface (e.g. ICP correspondence point-to-point error) between two point sets to match the pose of a view to another view of the same object in a similar pose stored in a database. PWSE is robust to the sparsity of a view, to the point of accurately finding the pose of a view with only hundreds of points, while being efficient enough to process a view in realtime [29].

PWSE features themselves are defined by the location of the local minima of the registration error surface in pose (rotational/translational) space, where registration is of a fixed set of points $M$, known as the registration model, to a view $M^i$ of a 3D surface model $M$ in an arbitrary relative pose.

Formally the error surface of the model $M$ with respect to the view $M^i$, is defined as the function:

$$\xi_i (R, T) = \sum_n \|M^i_n - (RM_n + T)\|^2, \quad (5.1)$$

where $M_n$ is the correspondence point to the registration model to a point $M^i_n \in M^i$ (determined by nearest neighbours), $R$ is a $3 \times 3$ rotation matrix, and $T$ is a $3 \times 1$ translation vector representing the parameters of the six-dimensional pose space.

Representing even the discretized error surface itself would be impractical due to its high dimensionality. Fortunately all that PWSE requires are the locations of the local minima (i.e. potential wells) in the pose space for each view $M^i$. Such a compact representation is called an embedding of the error surface $\xi_i$ with respect to
the view $\mathcal{M}^i$ and is defined:

$$E_i = \{ (T^i_k, R^i_k) | k = 1, \cdots, K \},$$

(5.2)

where there are $K$ local minima found on the error surface $\xi_i$, and $T^i_k$, $R^i_k$ are the translation and rotation respectively of the registration model $M$ relative to the view $\mathcal{M}^i$.

In both offline and online processing the same procedure is used to generate the embedding for each model and its associated views. At runtime these online features are matched to those generated offline by finding the closest local minima in the pose space and using the associated view to determine the pose. Finally the BHT [31] is used as a verification step.

Since PWSE is not a contribution of this thesis, and has been well documented elsewhere [30, 29], the reader is encouraged to reference these external resources for a better understanding of PWSE itself. This thesis will focus on the novel localization of PWSE, LPWSE, and PWSE’s application to mobile urban LIDAR data in the context of the overall object recognition methodology as described in Chapter 3.

### 5.2 Local Potential Well Space Embedding

The appeal of using PWSE in real world scenes, such as those produced by ground based LIDAR, is countered by PWSE’s limitations with occlusion and clutter in such data. Without the ability to appropriately segment objects in the scene, a global feature has limited use.

This thesis demonstrates a novel localization of the PWSE algorithm, LPWSE [18], in the form of a local shape descriptor which does not suffer from changes in
global properties while maintaining a robustness to sparsity. This is possible because just as the registration error surface of a 3D model is distinctive, so too is that of a local surface patch.

Covering the registration’s initial pose space as PWSE does, by perturbing every surface patch a large number of times, would be computationally inefficient. Instead we only use a single canonical pose for the registration model, transformed such that it is centred on each point. With only a single relative pose however, the orientation of a registration model with respect to each surface patch must be consistent under different orientations of a view. A different orientation may result in a different registration error. PWSE also uses a large complex registration model consisting of many points, which is not practical for a local surface patch that is composed of much fewer points than a full 3D model.

To address these problems, we use very simple registration models, the simplest of which is a spherical shell composed of much fewer points (∼100) than the typical registration model used by PWSE (> 1000 points). Using a non-orientable registration model (such as a spherical shell) and rotationally invariant feature derived from the registration error surface (such as the registration translation’s magnitude) avoids the problem of identifying an invariant basis for each point on the surface. Finally, to provide more discriminative power to compensate for the loss of dimensionality of our new error surface, a set of registration models of different sizes is used to completely characterize each surface patch at a range of scales, e.g. a set of concentric spherical shells of different radii.

The registration error is calculated solely on the set of correspondence points
between two point sets. These correspondences are usually determined by least-distance. An implicit localization may therefore be achieved by using registration models of sizes proportional to the size of the desired surface patch and centring the registration model on the point defining the centre of the surface patch. In practice this may not always prevent a point on the back face of an object model being chosen as a correspondence if the object is thin. Therefore a support angle [32] is used to find a surface patch around each point.

5.2.1 LPWSE Definition

Formally we can calculate a set of features for each point \( p \) in a model point cloud \( M \), on a surface patch \( S(p) \) of radius \( r_S \) using a support angle \( \theta_S \):

\[
S(p) \overset{\text{def}}{=} \{ x \mid x \in M, \| p - x \| < r_S, \theta_{N_p} < \theta_S \},
\]  

(5.3)

where \( \theta_{N_p} = \arccos \frac{N_x \cdot N_p}{\|N_x\| \|N_p\|} \), and \( \theta_S < \frac{\pi}{2} \).

Translating the origin of the registration model to the point \( p \) and running ICP for a fixed number of iterations generates a transformation in the form of a rotation matrix \( R(p) \) and translation vector, \( T(p) = (T_x(p), T_y(p), T_z(p)) \) for the point \( p \). The translation magnitude \( T(p) \overset{\text{def}}{=} \|T(p)\| \) may be used as an invariant feature for that point.

To represent the surface at more than one scale we use \( N \) registration models each with a different scale, i.e. in the case of spherical shells, \( N \) shells of radius \( r = R, \frac{R}{D}, \frac{R}{D^2}, \ldots, \frac{R}{D^N} \), where \( R \) is the initial radius. This produces \( N \) sets of features each representing a structure in a distinct scale related to that of the previous model by a divisor \( D \). Considering the scaled registration models as a whole we have an additional
Figure 5.1: Illustration of the Local PWSE features, when using 3 spherical shells as registration models, on an example model point cloud.

$$F(p) = \{T_1, \theta_1, T_2, \theta_2, T_3\}$$
set of rotationally invariant features - the relative angle \( \theta_i \) where \( i = 1, \ldots, N - 1 \) between each adjacent scale’s translation vector \( T_N(p) \).

Figure 5.1 illustrates the registration of \( n = 3 \) spherical shell models, \( \mathcal{G} = \{G_i|i=1,\cdots,N\} \) of radius \( r = R, \frac{R}{D}, \cdots, \frac{R}{D^N} \) centred on a point \( p \) in a sample point cloud. Each sphere has an ICP translation \( T_N \), and there exists \( (N-2) \) interscale angles. Thus the feature vector for the current point \( p \) may be defined as \( F(p) = \{T_1, \theta_1, T_2, \theta_2, T_3\} \).

5.2.2 LPWSE Algorithm

The LPWSE algorithm is described below, following directly from the definition presented in §5.2.1. A step-by-step description of the algorithm is presented, with each step representing a discrete operation in the complete algorithm.

**Feature Generation**

1. Generate a scaled set of sampled registration models. For example \( N \) spherical shell point sets, of radius \( r = (R, \frac{R}{D}, \frac{R}{D^2}, \ldots, \frac{R}{D^N}) \), each consisting of a constant number of points. The initial radius \( R \) is less than maximum scale of the objects we wish to recognize and \( D \) is the registration model divisor relating the radius of each scaled registration model.

2. For each object with model \( M \), and each point \( p \in M \):
   
   (a) Translate each of the \( N \) registration models such that they are centred on \( p \).
(b) Find the surface patch of radius $r_S$, satisfying a support angle $\theta_S$, surrounding $p$ (see the definition in Equation 5.3),

(c) Match each of the $N$ registration models to the object at point $p$ using a fixed number of ICP iterations,

(d) Calculate invariant features from the resulting ICP transformation as described in §5.2.1 for all of the $N$ registration models,

(e) Form a feature vector representing the point $p$ on the model’s surface.

**Offline**

The following preprocessing steps are completed offline in order to populate a feature database to be used online for object recognition.

1. Run the feature generation method to generate the feature vectors for every point in the models $M$.

2. The set of features for all points $p \in M$ forms the complete feature description of model $M$, to be stored in a database for future online processing.

**Online**

The following steps are completed at runtime to identify an object in the scene, with existing features in the database created offline.

1. Repeat the feature generation method to generate the same feature vectors for every point in the view $V$.

2. Perform classification (see §5.3) based on the stored features for each model $M$ found offline and stored in the database.
5.2.3 Oriented Registration Model

The more complex the registration model, the more distinctive the registration error surface will be, and hence the better the accuracy. All but the most simple registration models (e.g. a spherical shell) with complete symmetry require a local basis to be oriented consistently with a surface patch. However, any basis defined on a surface patch will require a minimum number of points to be accurately determined. The more complex the basis the more points, and the less robust it will be in sparse data.

For optimum performance we must balance the complexity of the registration model, and the increased discriminative power of the registration error surface, with the complexity of the basis, and the robustness to sparsity. We found that using a registration model with a simple unidirectional symmetry (such as a spherical shell with a line emerging from it), and a surface normal basis struck a good balance.

5.3 Classification

The LPWSE algorithm generates distinctive features by which objects in the scene may be recognized by, however the actual task of identifying an object is solved by a classifier. This section describes the classification method used in the experiments presented in §6.2. The possibility remains that other classification methods might potentially be used, and may even prove more effective.

5.3.1 Normalized Distance Vote

The simplest classifier we can use to classify a set of view feature vectors $F_V = \{F_{V_i} \mid i = 1 \ldots N\}$ to a model based on a set of similar model feature vectors $F_M = \{
\{ \mathbf{F}_M \mid j = 1 \ldots M \} \text{ is based on the } L_2 \text{ (Euclidean) distance,}

\[ D_{ij} = \| \mathbf{F}_M - \mathbf{F}_V \|_{L_2}. \tag{5.4} \]

The \( L_2 \) distance unfortunately will give a disproportionate weight to features with a larger numerical range. In order to prevent this we normalize each feature column of all feature vectors for each model \( F_M \) and view \( F_V \) prior to the classification.

To classify a set of view features to a model:

1. Normalize each set of model features and the set of view features (in the case of the models this should be done offline),

2. For each model \( M \) in the offline database, calculate the distance \( D_{ij}(M) \) between all possible model and view feature vector pairs, \( \mathbf{F}_M \) and \( \mathbf{F}_V \),

3. Of these distances, find the minimum distance over all feature vectors for each model, \( D(M) = \min(D_{ij}(M)) \),

4. Classify the sample vector as belonging to the model \( M \) with the minimum distance \( D(M) \), and increment the vote for that model.

Finally to classify a segmented view, choose the model \( M \) with the majority of votes.

### 5.3.2 RANSAC Verification

Each feature correspondence found by the classifier is a hypothesis on a point correspondence between a model in the database and the view being classified. Not all of these point correspondences will be correct, and so there will be a significant number
of outliers (i.e. incorrect correspondences). Given a minimum set of three hypothe-
sized correspondence points, we may recover the pose between a model and a view
using a robust estimator of the model, such as random sample consensus (RANSAC)
[6], and calculate a correspondence error. This correspondence error is indicative
of the validity of the point correspondences for each associated model classification.
Therefore a RANSAC verification step runs until a minimum correspondence error
is achieved for a particular model, or for a maximum number of iterations, and the
model with the least correspondence error is chosen.

5.3.3 Parameter Selection

There are many parameters to the LPWSE algorithm which may require optimizing
in offline processing for best performance on a specific model database. These param-
eters include the initial radius $R$ of the registration model, the number of registration
models $N$, the registration model scale divisor $d$, and the support angle $\theta_S$ used for
the surface patch.

An exhaustive search of the parameter space would be infeasible, however pa-
rameter optimization is a well addressed problem with many solutions. We used a
readily available genetic programming library [21] to perform a genetic search on the
parameter space.
Chapter 6

Experimental Results

6.1 Segmentation Experiments

This section presents various applications of the DoN operator as an interest operator in pressing segmentation problems on mobile urban LIDAR data for real environments.

Results presented are of the form of the resulting point cloud of the interest operator, which may be compared to the original point clouds. Evaluation of the isolation results is necessarily visual: with no ground truth data (i.e. a point cloud representing the ideal points to have been segmented for a given problem and set of parameters), a formal evaluation of the isolation accuracy is not possible. This lack of ground truth is a problem endemic to the field, especially given the subject of these experiments and dense urban LIDAR data sets in general. Furthermore, given the medium of this thesis, only a 2D projection of the 3D result is possible.

Despite the lack of formal evaluation, and limitations of displaying projected point clouds, the author is confident that the visual results presented in this section are
sufficiently pronounced that little doubt will be left in the reader’s mind as to the usefulness of the method and the significance of the results.

6.1.1 Complete 3D Point Cloud

DoN was first tested on a complete point cloud model of a rabbit lawn ornament, shown in Figure 6.1a, scanned using a NextEngine 3D Scanner. This data has the advantage of being complete, as a full 3D sampling of the model, and relatively noise and artifact free. The rabbit exhibits a range of surface structure on different scales, from fine ridges on the surface representing hair to large scale surfaces such as the stable bottom of the ornament. The rabbit model is of dimensions $24.6 \times 26.7 \times 15.5$ cm. Figure 6.1 shows the magnitude (represented by scaled intensity) of the DoN for each surface point on the rabbit model at various scales. At the lowest scale the result resembles noise, but as the scale rises, lower frequency regions are highlighted. In particular the fine “hair” of the rabbit model emerges in the smallest of scales, the rabbit’s eye is only highlighted on medium-small scales (where the lower radius is 0.5 cm through to 0.8 cm) while the outline of the soles of the rabbit’s feet are only outlined in larger scales (where the lower radius is 0.8 cm through to 2.0 cm). These results show a clear highlighting of points in the rabbit model, based on the scale of the surface structure to which those points belong.

6.1.2 TITAN® Urban Mobile LIDAR Data

As was discussed in Chapter 1, recent advances in sensor technology mean that collection of dense urban LIDAR data using a mobile platform is now possible, leading
CHAPTER 6. EXPERIMENTAL RESULTS

Figure 6.1: DoN segmentation results on Rabbit model.
to the possibility of automatically generating urban models suitable for use in Geographical Information Systems (GIS). With these urban LIDAR scenes typically consisting of hundreds of millions of points (and complete cities on the order of billions of points) however, extracting even the most efficient of object recognition features from an entire scene is prohibitively time expensive.

Various algorithms for segmentation of range data have been proposed, however most of these do not apply to unorganized range data. Liu et al. introduced Cell Mean Shift (CMS) [24], which maps the normal map of an unorganized point cloud to a Gaussian sphere, producing a Gaussian image, this spherical image can be clustered to identify shapes. Huang et al. [16] proposed a method that involves generating a mesh for the data, and finding a set of well-defined borders by which the object is segmented. Woo et al. [37] propose an octree-based method for handling large unorganized point clouds, using edges to segment structures within. None of these methods are as computationally simple as DoN, in fact all of these algorithms require, as an integral step, calculating normal estimates for the point cloud.

With a focus on the recognition of street furniture (i.e. lamp posts, fire hydrants, curbs) and extraction of large-scale infrastructure (e.g. buildings, roads) in GIS applications, there is also an intrinsic scale-based focus of interest on the scene. The following results demonstrate various applications for DoN on real-world urban LIDAR data.

For illustration purposes results are demonstrated using small (25 m²) subsections of a real-world urban LIDAR data set of Kingston, ON, Canada by the TITAN® mobile terrestrial scanner [8]. No noise filtering, meshing, re-sampling or similar pre-processing was performed. These results were also observed on much larger data sets
of hundreds of millions of points.

**Multi-Scale Isolation**

The vast majority of the points in a typical urban LIDAR scene (e.g. road surface, pavement, etc.) are irrelevant to the task of identifying street furniture, or in fact any objects of a particular scale. Any pre-filtering algorithm that could accurately yet efficiently identify points of interest in 3D unorganized point clouds based on scale would significantly speed-up any further processing of the scene, including object recognition using local methods such as LPWSE and spin images [19]. Segmentation of a scene is also a necessary pre-requisite for many object recognition algorithms, especially global methods, such as PWSE [30].

DoN has two parameters, a large radius ($r_h$) and a small radius ($r_l$). The parameter $r_l$ corresponds to the upper-bound of the lowest sized structure to isolate from the scene, while $r_h$ controls the bandwidth of the filter. The resulting DoN vector map may be thresholded on any of the DoN components, or the DoN magnitude. Empirically it was found that the relationship of the two radii providing a good bandwidth was $r_h = 10 r_l$. Figures 6.2, C.1, C.3, C.5, C.7 and C.9 show the results of DoN filtering of various mobile urban LIDAR scenes into points of various scales. At the lowest scale, shown in Figure 6.3a (0.05 – 0.5 m), noise in the data is clearly dominant, although edges for some small scale structure are already visible. As the scale increases from 0.05 – 0.5 m to 0.1 – 1.0 m, as shown in Figures 6.3a through 6.3f, noise fades away and sharp edges are clearly segmented, including and architectural facades and building edges, such as window outlines and pipes, and ground edges, such as street curbs. As the scale is increased above 0.1 – 1.0 m, starting with Figure
Figure 6.2: DoN segmentation results on King St. East and Princess St, Kingston, ON, Canada (3606311 points).
(a) $|\Delta n(0.05 \text{ m}, 0.5 \text{ m})| \geq 0.250$: 586532 points.

Figure 6.3: DoN segmentation results on King St. East and Princess St, Kingston, ON, Canada (pg. 2 of 12, starting pg. 56).
Figure 6.3: DoN segmentation results on King St. East and Princess St, Kingston, ON, Canada (pg. 3 of 12, starting pg. 56).

(b) $|\Delta_\theta(0.06\text{ m}, 0.6\text{ m})| \geq 0.250$: 730457 points.
Figure 6.3: DoN segmentation results on King St. East and Princess St, Kingston, ON, Canada (pg. 4 of 12, starting pg. 56).
Figure 6.3: DoN segmentation results on King St. East and Princess St, Kingston, ON, Canada (pg. 5 of 12, starting pg. 56).

(d) $|\Delta n(0.08 \text{ m}, 0.8 \text{ m})| \geq 0.250$: 714914 points.
Figure 6.3: DoN segmentation results on King St. East and Princess St, Kingston, ON, Canada (pg. 6 of 12, starting pg. 56).

(e) $|\Delta n(0.09\text{ m}, 0.9\text{ m})| \geq 0.250$: 703265 points.
(f) $|\Delta_\delta(0.1\text{ m}, 1.0\text{ m})| \geq 0.250$: 717577 points.

Figure 6.3: DoN segmentation results on King St. East and Princess St, Kingston, ON, Canada (pg. 7 of 12, starting pg. 56).
(g) $|\Delta_n(0.15 \text{m}, 1.5 \text{m})| \geq 0.250$: 800552 points.

Figure 6.3: DoN segmentation results on King St. East and Princess St, Kingston, ON, Canada (pg. 8 of 12, starting pg. 56).
(h) $|\Delta_n(0.2\text{ m}, 2.0\text{ m})| \geq 0.250$: 839593 points.

Figure 6.3: DoN segmentation results on King St. East and Princess St, Kingston, ON, Canada (pg. 9 of 12, starting pg. 56).
(i) $|\Delta h(0.5\text{m,} 5.0\text{m})| \geq 0.250$: 1110400 points.

Figure 6.3: DoN segmentation results on King St. East and Princess St, Kingston, ON, Canada (pg. 10 of 12, starting pg. 56).
Figure 6.3: DoN segmentation results on King St. East and Princess St, Kingston, ON, Canada (pg. 11 of 12, starting pg. 56).
(k) $|\Delta_n(1.0 \text{ m}, 10.0 \text{ m})| \geq 0.250$: 1524829 points.

Figure 6.3: DoN segmentation results on King St. East and Princess St, Kingston, ON, Canada (pg. 12 of 12, starting pg. 56).
6.3g, larger scale structures begin to emerge from the scene, such as cars, and wall segments. We continue to find larger and larger objects to be clearly segmented, while smaller objects are increasingly ignored, until at the largest scale shown, in Figure 6.3k (1.0 – 10.0 m), larger building fronts and walls are segmented from the rest of the scene. The ability to do just such scale-sensitive segmentation, quickly, on large mobile urban LIDAR data is of significance to the many organizations faces with processing similar data sets on a daily basis.

**Isolation of Street Furniture**

Figure 6.4 shows a closeup of a fire hydrant in an input scene (Figure D.2) after applying DoN with a set of radii between 0.1 – 2.0 m with scale ratios \( \frac{r_h}{r_l} \) of 10, and a normal difference magnitude threshold of \( \| \Delta_\hat{n}(p) \| \leq 0.25 \). The points are coloured according to the (scaled) magnitude of the DoN per-axis component magnitude, where the axis is \( x, y \) and \( z \) for red, green and blue respectively. The isolation of a particular object along with the whole scene are evaluated.

While it can be seen in Figure 6.4 that different scales do achieve isolation of the fire hydrant from the ground, only a few scales preserve the majority of the fire hydrant in the process, the 0.2 – 2.0 m scale in particular was found to provide good isolation from the scene, while preserving the majority of the fire hydrant points. Figure 6.5 shows the complete process for the segmentation of a fire hydrant from the scene, going from a scene of over 600,000 points to a segmented fire hydrant of only 121 points. After DoN filtering, objects in the scene are well isolated, allowing use of a connected components algorithm [12] or clustering method. DoN filtering also allows easy manual segmentation of objects, and for simplicity in this thesis
CHAPTER 6. EXPERIMENTAL RESULTS

Figure 6.4: DoN segmentation results on TITAN scene fragment containing a fire hydrant.
(a) Original LIDAR Data (620820 points)

(b) Normal Estimation w/ Small Support Region ($r_1 = 0.2$ m)

(c) Normal Estimation w/ Large Support Region ($r_1 = 2.0$ m)

(d) DoN Segmented Data $\Delta_n(0.2, 2.0) \geq 0.25$ (132214 points)

(e) Connected Components/Manual Segmentation (121 points)

Figure 6.5: Illustration of the complete process of segmentation of an object from a scene using DoN.
segmented objects were isolated from DoN filtered scenes manually using a simple fencing segmentation and existing point cloud manipulation software (Cyclone).

Figures 6.6, 6.7, and 6.8 show a variety of objects segmented using DoN from mobile urban LIDAR data, demonstrating the generality of the isolation results. As can be seen the sparsity of the scans varies considerably, even amongst objects of the same class.

Using DoN and a connected components algorithm to automatically segment objects in the scene would fundamentally speed up an object recognition pipeline for finding street furniture, for example fire hydrants, in an urban LIDAR scene for object recognition algorithms based on local methods. As demonstrated in these results, segmentation of objects is complete and accurate enough to even enable the use of global object methods.

Isolation of Non-Ground Points

Although normal ambiguity is normally a problem that must be resolved for DoN results to be meaningful, it was found in some applications it could actually be used to advantage. Segmentation of ground from non-ground in an urban LIDAR image is a pressing problem, to which the authors found the naive solution of removing planar points or points with a normal threshold is not sufficient to solve without a great deal of outlier or inlier removal. By only fixing the normal ambiguity in the Z axis into the positive hemisphere (which in urban LIDAR data can be assumed to contain the ground normal), the DoN for non-ground points, whose normals have principle directions mostly in the X-Y plane, is highly exaggerated. The result is a scene where non-ground points are easily segmented from the rest of the scene simple
(a) Car 1: 2902 points.  
(b) Car 2: 1362 points.  
(c) Car 3: 708 points.  
(d) Car 4: 382 points.  
(e) Car 5: 853 points.  
(f) Car 6: 4177 points.  
(g) Car 7: 739 points.  
(h) Car 8: 643 points.  
(i) Car 9: 534 points.  
(j) Car 10: 391 points.  
(k) Car 11: 722 points.

Figure 6.6: Segmented objects of class car from TITAN data of Kingston, ONT.
(a) Person 1: 209 points.

(b) Person 2: 189 points.  
(c) Person 3: 176 points.  
(d) Person 4: 101 points.

(e) Person 5: 170 points.  
(f) Person 6: 101 points.

Figure 6.7: Segmented objects of class *person* from TITAN data of Kingston, ONT.
(a) Fire Hydrant 1: 246 points.

(b) Fire Hydrant 2: 474 points.

(c) Fire Hydrant 3: 54 points.

(d) Fire Hydrant 4: 97 points.

Figure 6.8: Segmented objects of class firehydrant from TITAN data of Kingston, ONT.
(a) original: 3606311 points.

(b) $|\Delta_{n}(0.2\text{ m}, 2.0\text{ m})| \geq 0.250$: 1173513 points.

Figure 6.9: DoN non-ground segmentation results for King St. East and Princess St, Kingston, ON, Canada: 1173513/3606311 points (32%).
by thresholding the magnitude of the DoN in the scene.

Figures 6.9, D.1, D.2, D.3, D.4 and D.5 show input scenes before and after DoN, without solved normal ambiguity in the X and Y axis, is applied at $0.2 - 2.0\ m$, $\|\Delta_n(p)\| \leq 0.425$. In both cases it can be seen that the majority of ground points have been removed, leaving non-ground points intact, a reduction in points of 75%.

**Curb Isolation**

The DoG operator is commonly used for edge detection in 2D images, similarly the DoN operator may isolate 3D edges from scenes, as seen in Figure 6.2. A particularly strong edge found in urban LIDAR data is the road curb\(^1\). A set of scale and threshold parameters was found to isolate the curb in an urban LIDAR scene. Since the DoN is normalized, these parameters hold for any data in which a curb of similar type exists.

Figures 6.10, E.1, E.2, E.3, E.4 and E.5 show the curbs extracted from the input scenes using these parameters. The curb points isolated from the scene represent only approximately 0.20% of the total scene points. Note that although there is noise present in the filtered result, the clear majority of points with any order in the result belong to the curb, and some form of outlier post-filtering would improve the results. Discontinuities in the curb are due to a lack of data rather than the segmentation, and can be attributed to parked cars, other occlusions, or simply a lack of sufficient data for the curb in those locations when the scene was scanned.

---

\(^1\)Accurate separation of the curb is of strong interest to TITAN system users.
(a) original: 3606311 points.

(b) curb: 7555 points.

Figure 6.10: DoN curb segmentation results on King St. East and Princess St, Kingston, ON, Canada: 7555/3606311 points (0.20%).
6.2 Object Recognition Experiments

6.2.1 Motivation

Without accurate and complete segmentation of an urban scene, a global method such as PWSE is not feasible for identifying objects. As such a localized variant of PWSE, LPWSE, was presented in Chapter 5 to address the problem of object recognition in sparse urban LIDAR data with no prior segmentation of the scene. One of the primary considerations in the design of the LPWSE algorithm was a robustness to data sparsity, an issue typically not well addressed by local object recognition methods. Hence LPWSE was first evaluated on the recognition of segmented objects, and associated views, to test the method’s robustness to sparsity on real range data.

6.2.2 Experimental Setup

![Image of objects](image)

Figure 6.11: Objects used in LPWSE experiments.

The experimental results were based on the classification of 5 objects with a wide range of sizes, as shown in Figure 6.11. Each object was scanned using a NextEngine
3D Scanner generating 5 dense model point clouds. In addition ten views were captured with the scanner for each of the models, resulting in 50 views altogether for which no additional (e.g. noise) filtering was performed. Figure 6.12 shows an example model point cloud and view. In testing each model point cloud used for offline feature generation was randomly sub-sampled to 10,000 points, whilst each view was randomly sub-sampled to 5000, 3750, 2500, 1250, 500, 100, and 50 points.

6.2.3 Experimental Results

The optimum parameters for this particular set of objects (see §5.3.3) were found to be $N = 5$, $R = 172.2\text{mm}$, $\theta_S = 74$ degrees, $D = 1.275$ for the spherical shell with line registration model.

Figure 6.13 shows the classification accuracy for a range of randomly sub-sampled views, from a maximum of 5000 points to a minimum of 50, both before and after 500 iterations of RANSAC verification. Accuracy is relatively high until approximately 500 points, after which the accuracy drops off quickly. Accuracy peaks at 92% for 3750
Figure 6.13: Classification accuracy for a set of 50 randomly sub-sampled views.

<table>
<thead>
<tr>
<th>Model</th>
<th>Dimensions (mm)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Rabbit</td>
<td>$246 \times 267 \times 155$</td>
</tr>
<tr>
<td>Goose</td>
<td>$336 \times 365 \times 301$</td>
</tr>
<tr>
<td>Sailor</td>
<td>$251 \times 135 \times 250$</td>
</tr>
<tr>
<td>Gnome</td>
<td>$183 \times 277 \times 156$</td>
</tr>
<tr>
<td>Horse</td>
<td>$83 \times 40 \times 149$</td>
</tr>
</tbody>
</table>

Table 6.1: Object Dimensions
### Table 6.2: Confusion matrix for classification of 50 views after 500 RANSAC verification with 5000 points.

<table>
<thead>
<tr>
<th>True Model</th>
<th>Rabbit</th>
<th>Goose</th>
<th>Sailor</th>
<th>Gnome</th>
<th>Horse</th>
</tr>
</thead>
<tbody>
<tr>
<td>Rabbit</td>
<td>9</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Goose</td>
<td>0</td>
<td>10</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Sailor</td>
<td>0</td>
<td>0</td>
<td>7</td>
<td>3</td>
<td>0</td>
</tr>
<tr>
<td>Gnome</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>9</td>
<td>0</td>
</tr>
<tr>
<td>Horse</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>10</td>
</tr>
</tbody>
</table>

### Table 6.3: Confusion matrix for classification of 50 views after 500 iterations of RANSAC verification with 1250 points.

<table>
<thead>
<tr>
<th>True Model</th>
<th>Rabbit</th>
<th>Goose</th>
<th>Sailor</th>
<th>Gnome</th>
<th>Horse</th>
</tr>
</thead>
<tbody>
<tr>
<td>Rabbit</td>
<td>6</td>
<td>0</td>
<td>3</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>Goose</td>
<td>0</td>
<td>9</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Sailor</td>
<td>0</td>
<td>0</td>
<td>10</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Gnome</td>
<td>0</td>
<td>0</td>
<td>5</td>
<td>4</td>
<td>1</td>
</tr>
<tr>
<td>Horse</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>10</td>
</tr>
</tbody>
</table>

### Table 6.4: Confusion matrix for classification of 50 views after 500 RANSAC verification with 500 points.

<table>
<thead>
<tr>
<th>True Model</th>
<th>Rabbit</th>
<th>Goose</th>
<th>Sailor</th>
<th>Gnome</th>
<th>Horse</th>
</tr>
</thead>
<tbody>
<tr>
<td>Rabbit</td>
<td>6</td>
<td>0</td>
<td>3</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>Goose</td>
<td>0</td>
<td>9</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Sailor</td>
<td>0</td>
<td>0</td>
<td>10</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Gnome</td>
<td>0</td>
<td>0</td>
<td>5</td>
<td>4</td>
<td>1</td>
</tr>
<tr>
<td>Horse</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>10</td>
</tr>
</tbody>
</table>
points with 500 iterations of RANSAC, remaining above 80% until approximately 1000 points.

Tables 6.2, 6.3 and 6.4 show the confusion matrices for 5000, 1250 and 500 points respectively. As seen in these confusion matrices, as the number of points decreases, discrimination between the Sailor and Gnome objects, which are of similar dimensions/general shape, to the point that at 500 points there is a good chance that views of gnome will be mistaken for sailor. While such confusion is detrimental to solving the problem of object recognition, this may be indicate that LPWSE would be useful in application to object class recognition, as Shang et al. recently demonstrated with PWSE [30]. Despite the results for views of 500 points, overall accuracy in sparse data is fairly robust, and models with a more distinctive geometry, e.g. Horse and Goose, fair well even at 500 points.

6.2.4 Applying LPWSE to Mobile Urban LIDAR Data

The results presented in §6.2.3 indicate LPWSE is relatively robust to sparse data as compared to most local object recognition methods, where views on the order of thousands of points are the de facto minimum. LPWSE shows a graceful decline in accuracy as object views decrease from thousands of points to hundreds. Despite this, the results illustrated by Figure 6.13 suggest that the usage of LPWSE in urban LIDAR scenes would be limited to objects with views of at the very least 600 points for a reasonable degree of accuracy (approximately 80%), or views on the order of 3500 points for an accuracy of greater than 90%. However as discovered in Chapter 6.1, in practice this is not the case for street furniture and other typical objects of interest, at least with the current state of mobile urban LIDAR data.
Chapter 7

Conclusion

This thesis addresses an emerging topic of interest to both the computer vision and geographic information systems (GIS) communities, and the broad set of LIDAR data users in general. With systems such as TITAN® now entering the market, the increased availability of dense, accurate terrestrial urban LIDAR data means that the current manually intensive methods for analysis of such data are now a significant bottleneck. Not only does the lack of good tools for automatic analysis impede the ability of companies to deliver products based on such data, but it also precludes the usage of mobile urban light detection and ranging (LIDAR) data in other applications by creating a significant cost barrier, due mostly to the high cost of the current, labour intensive, analytical methods. As LIDAR sensor technology and integration improve, the availability of dense, accurate terrestrial LIDAR data, and range data in general, can only be expected to increase dramatically in future, simultaneously further effecting the need of better tools for analysis of such data, towards which 3D computer vision will play a fundamental role.

Towards such an end, this thesis outlined a complete methodology for automatic
object recognition in mobile urban LIDAR data, of which segmentation and object recognition are fundamental building blocks. Two novel algorithms, one for segmentation and the other for object recognition were introduced and evaluated in their potential for application to mobile urban LIDAR data alone, and towards the overall methodology.

7.1 Segmentation

A novel segmentation algorithm, using difference of normals (DoN) as an interest operator, was proposed for scale-based segmentation of unorganized range data. In §6.1 DoN was shown to be an effective method of isolating objects of interest in mobile urban LIDAR data at difference scales, for example as illustrated by Figure 6.2. It was also demonstrated to effectively isolate street furniture, as illustrated by the segmentation of a fire hydrant from an urban scene in Figure 6.4 and 6.5. Furthermore non-ground segmentation was demonstrated, for example in Figure 6.9, removing ground points from scenes while leaving non-ground points intact. Finally isolation of curb points was demonstrated, for example in Figure 6.10, using only DoN and empirically derived thresholds. Each of these applications of the DoN operator to real mobile urban LIDAR data effectively addressed real-world problems for which there are currently few, if any, efficient solutions.

7.2 Object Recognition

Global object recognition methods have the inherent potential to perform better in the presence of sparse data, since they may rely on an entire object in recognition
rather than a subset of it’s surface points as local methods must rely on. Consequently however, global methods rely on accurate and complete segmentation of the objects of interest from the scene. Therefore without accurate and complete segmentation of mobile urban LIDAR data, only local object recognition methods are feasible. However the density of mobile urban LIDAR data is significantly less than that encountered in the typical scene for which existing local methods were designed.

A novel local object recognition method, local potential well space embedding (LPWSE), was introduced in Chapter 5 based on an existing global object recognition method, potential well space embedding (PWSE). LPWSE was designed to be relatively robust to data sparsity as compared to current global object recognition methods. In §6.2, Figure 6.13, a graceful degradation in classification accuracy was exhibited as object views decreased from 5000 down to 500 points, showing LPWSE to have some potential as a local feature for object recognition in sparse range data. However, the accuracy of classification, even for a small number of models, drops off severely after approximately 500 points. Meanwhile in §6.1 segmented street furniture in mobile urban LIDAR data was found to consist of only 100-300 points, as seen in Figures 6.7 and 6.8. Even views of larger objects, such as cars, might only consist of as few as 400 points, as shown in Figure 6.6. It can thus only be concluded that LPWSE is likely not suitable for the level of sparsity exhibited by segmented objects in mobile urban LIDAR data, at least in it’s current state.

7.3 Methodology

The nature of mobile urban LIDAR data, and existing segmentation algorithms, would make accurate and complete segmentation of objects very difficult and preclude
the use of global object recognition methods, hence motivating research resulting in a local object recognition method robust to sparsity, i.e. LPWSE. In conclusion however, both the unexpected success of DoN’s segmentation accuracy and the failure of LPWSE’s robustness to sparsity to scale to such extremes, means that global methods, in particular PWSE, represent the most promising solution to object recognition in mobile urban LIDAR data.

PWSE has been shown to be very robust to sparse data, as the rate of recognition varied by only about 0.3% when the number of points was reduced from 1,000 to 500 [30]. Even when only using 125 points per image (approximately the same as the fire hydrant in Figure 6.5), PWSE still achieved a 92.6% correctness rate. Furthermore PWSE has been shown to be effective at solving the object class recognition problem [30], meaning it might be used to find classes of objects on urban LIDAR data, e.g. vehicles, even when a-priori information (i.e. an object model) is not available before runtime.

It seems unlikely that any future local method for object recognition might scale to sparsity levels of just over 100 points while maintaining a high classification accuracy. Thus the author believes the key to a methodology for recognizing objects in mobile urban LIDAR data includes algorithms for accurate, efficient and complete segmentation of the objects of interest, such as the Difference of Normals algorithm.

### 7.4 Future Work

The DoN operator in particular appears to be ripe with potential for further research, in particular a look at a pyramid-like structure to assign points to specific scales, much like what is done with the difference of Gaussians (DoG) in the SIFT [25] algorithm...
to identify the scale of pixels in a 2D image. Furthermore it’s potential to find points of salience might be exploited by a new method for local object recognition. Any method in 2D computer vision that makes use of a scale operator, such as DoG, provides an avenue for further research of the DoN operator in range data.

In the implementation of the DoN operator, there is much potential for improvement. Normal calculations represent the bottleneck for the DoN operator, and stand as an obstacle for any realtime applications. Specifically any spatial structure used in computing the range queries required for normal calculations requires a computational overhead which may imply application of the DoN operator is impossible in realtime, specifically a construction and update penalty. Alternative methods of normal calculation, and the highly parallelizable nature of the calculations mean that there may still be the potential for realtime performance.

Another potential area of future research would be a hybrid methodology using both terrestrial and airborne LIDAR. Airborne LIDAR has the advantage of representing a very large scale, and since it can be processed as a depth image, a broad set of existing methods for road, building, and land use identification. Terrestrial mobile LIDAR has the advantage of being able to scan much smaller scales, and coverage of areas not scanned by airborne platforms.
Bibliography


[33] Babak Taati and Michael Greenspan. Satellite pose acquisition and tracking with 
variable dimensional local shape descriptors. In IEEE/RSJ IROS 2005 Workshop 

[34] Tolga Tasdizen, Ross Whitaker, Paul Burchard, and Stanley Osher. Geometric 

[35] Emanuele Trucco and Alessandro Verri. Introductory Techniques for 3-D Com-

In Acoustics, Speech, and Signal Processing, IEEE International Conference on 

method for point cloud data. International Journal of Machine Tools and Man-

[38] C. Wyatt, E. Bayram, and Yaorong Ge. Minimum reliable scale selection in 3d. 

[39] Gerard Medioni Yang Chen. Object modeling by registration of multiple range 
images. In Proceedings 1991 IEEE International Conference on Robotics and 
Appendix A

TITAN®

A.0.1 Introduction to TITAN®

Figure A.1: The TITAN® terrestrial mobile scanning platform.
TITAN® is a terrestrial mobile LIDAR scanning platform\textsuperscript{1} from Terrapoint Inc. The TITAN® platform consists of multiple of-the-shelf LIDAR scanners, inertial measurement unit (IMU)s, and cameras on a boom that may be raised when in operation, or lowered when not in operation. This scanning platform is in turn connected to a computer and highly accurate global positioning system (GPS) system. The scanning platform is usually mounted on the back of a lightly modified pickup truck, although it has been mounted on other vehicles when required. Figure A.1 shows the current configuration.

TITAN® represents the state-of-the-art terrestrial mobile scanning platform as of writing, in fact until TITAN® the only practical mobile platforms for scanning urban regions were airborne. Airborne platforms however have limited usefulness for urban modelling, they are relatively sparse, and the majority of the return off of buildings is from the roof, with very little return from walls and practically none from street furniture such as lamp posts, fire hydrants, signs, etc.

TITAN® can collect thousands of data points with a best-case accuracy of approximately 2 cm horizontally and 3 cm vertically for targets within 25 meters of the vehicle. It may do so at normal road speeds (50–80 km/h), and hence not interfere with normal traffic flow. In practice TITAN® is often driven both directions to collect a more complete scan of the throughway, and allow verification and further rectification of the points by post processing, and some estimation of systematic error. Furthermore fixed targets are used to ensure correctness of the global localization of the data. For an in-depth study of the error calculations behind the TITAN® specs, please refer to [8].

\textsuperscript{1}For an introduction to LIDAR and LIDAR scanning see §1.2.
A.0.2 TITAN® Post-processing

The TITAN® platform is not realtime, the raw output of the scanners, IMU and GPS systems is collected for post-processing. This allows for extrapolation of the position of the vehicle when GPS is not available (in tunnels, urban canyons, etc), and the calculations required to correlate the data. While rectification of a stationary platform is simple, rectifying the raw output of 4 LIDAR scanners with 4 cameras a GPS and an IMU, is a daunting task that is easy to understate. The engineering feat of Terrapoint’s TITAN® represents is as much of a software effort as hardware [8, 9].

The output of this processing (by Terrapoint using their own software) is a complete correlated pointcloud with points encoded with UTM coordinates and attributed with LIDAR intensity (the intensity return of the laser, in this case infrared), RGB from camera and optionally the rectification parameters.

A.0.3 TITAN® Specification Overview

Accuracy (absolute): $\pm 3 - 5$ cm.

Accuracy (relative): $\pm 2.5$ cm.

LIDAR Scanners: 4 off-the-shelf scanners, 2 on back $45^\circ$ up/down, 2 on sides $10^\circ$ forward.

Positioning: GPS/IMU for positioning, with fixed targets used to check localization accuracy.
Appendix B

LIDAR Scenes Used in Experiments
Figure B.1: LIDAR point cloud for Bagot St, Kingston, ON, Canada: 478377 points.
Figure B.2: LIDAR point cloud for Intersection of Aberdeen and William St, Kingston, ON, Canada: 523115 points.
Figure B.3: LIDAR point cloud for Intersection of Bagot and Brock St, Kingston, ON, Canada: 484046 points.
Figure B.4: LIDAR point cloud for Intersection of Clergy and Johnson St, Kingston, ON, Canada: 614403 points.
Figure B.5: LIDAR point cloud for Intersection of Princess and Bagot St, Kingston, ON, Canada: 620820 points.
Appendix C

Supplemental Difference of Normal Segmentation Results
Figure C.1: DoN segmentation results on Bagot St, Kingston, ON, Canada (478377 points).
(a) $|\Delta h(0.1 \text{ m}, 1.0 \text{ m})| \geq 0.250$: 97767 points.

Figure C.2: DoN segmentation results on Bagot St, Kingston, ON, Canada (pg. 2 of 7, starting pg. 103).
Figure C.2: DoN segmentation results on Bagot St, Kingston, ON, Canada (pg. 3 of 7, starting pg. 103).

(b) $|\Delta n(0.16 \text{ m}, 1.6 \text{ m})| \geq 0.250$: 113298 points.
Figure C.2: DoN segmentation results on Bagot St, Kingston, ON, Canada (pg. 4 of 7, starting pg. 103).
Figure C.2: DoN segmentation results on Bagot St, Kingston, ON, Canada (pg. 5 of 7, starting pg. 103).
(e) $|\Delta_\delta(0.8 \text{ m}, 8.0 \text{ m})| \geq 0.250$: 191779 points.

Figure C.2: DoN segmentation results on Bagot St, Kingston, ON, Canada (pg. 6 of 7, starting pg. 103).
Figure C.2: DoN segmentation results on Bagot St, Kingston, ON, Canada (pg. 7 of 7, starting pg. 103).
Figure C.3: DoN segmentation results on Intersection of Aberdeen and William St, Kingston, ON, Canada (523115 points).
Figure C.4: DoN segmentation results on Intersection of Aberdeen and William St, Kingston, ON, Canada (pg. 2 of 7, starting pg. 110).
(b) $|\Delta_0(0.16\, m, 1.6\, m)| \geq 0.250$: 139746 points.

Figure C.4: DoN segmentation results on Intersection of Aberdeen and William St, Kingston, ON, Canada (pg. 3 of 7, starting pg. 110).
APPENDIX C. SUPPL. DON SEGMENTATION RESULTS

Figure C.4: DoN segmentation results on Intersection of Aberdeen and William St, Kingston, ON, Canada (pg. 4 of 7, starting pg. 110).

(c) $|\Delta n(0.2\text{ m}, 2.0\text{ m})| \geq 0.250$: 152044 points.
Figure C.4: DoN segmentation results on Intersection of Aberdeen and William St, Kingston, ON, Canada (pg. 5 of 7, starting pg. 110).
Figure C.4: DoN segmentation results on Intersection of Aberdeen and William St, Kingston, ON, Canada (pg. 6 of 7, starting pg. 110).
(f) $|\Delta h(1.0 \text{ m}, 10.0 \text{ m})| \geq 0.250$: 205324 points.

Figure C.4: DoN segmentation results on Intersection of Aberdeen and William St, Kingston, ON, Canada (pg. 7 of 7, starting pg. 110).
Figure C.5: DoN segmentation results on Intersection of Bagot and Brock St, Kingston, ON, Canada (484046 points).
Figure C.6: DoN segmentation results on Intersection of Bagot and Brock St, Kingston, ON, Canada (pg. 2 of 7, starting pg. 117).
Figure C.6: DoN segmentation results on Intersection of Bagot and Brock St, Kingston, ON, Canada (pg. 3 of 7, starting pg. 117).
Figure C.6: DoN segmentation results on Intersection of Bagot and Brock St, Kingston, ON, Canada (pg. 4 of 7, starting pg. 117).

\[(c) \quad \|\Delta \hat{n}(0.2 \text{m}, 2.0 \text{m})\| \geq 0.250: \quad 70220 \text{ points.}\]
Figure C.6: DoN segmentation results on Intersection of Bagot and Brock St, Kingston, ON, Canada (pg. 5 of 7, starting pg. 117).
(e) $|\Delta n(0.8 \text{m}, 8.0 \text{m})| \geq 0.250$: 126351 points.

Figure C.6: DoN segmentation results on Intersection of Bagot and Brock St, Kingston, ON, Canada (pg. 6 of 7, starting pg. 117).
Figure C.6: DoN segmentation results on Intersection of Bagot and Brock St, Kingston, ON, Canada (pg. 7 of 7, starting pg. 117).
Figure C.7: DoN segmentation results on Intersection of Clergy and Johnson St, Kingston, ON, Canada (614403 points).
Figure C.8: DoN segmentation results on Intersection of Clergy and Johnson St, Kingston, ON, Canada (pg. 2 of 7, starting pg. 124).
Figure C.8: DoN segmentation results on Intersection of Clergy and Johnson St, Kingston, ON, Canada (pg. 3 of 7, starting pg. 124).
Figure C.8: DoN segmentation results on Intersection of Clergy and Johnson St, Kingston, ON, Canada (pg. 4 of 7, starting pg. 124).
Figure C.8: DoN segmentation results on Intersection of Clergy and Johnson St, Kingston, ON, Canada (pg. 5 of 7, starting pg. 124).

(d) $|\Delta_\alpha(0.5\text{ m}, 5.0\text{ m})| \geq 0.250$: 130622 points.
Figure C.8: DoN segmentation results on Intersection of Clergy and Johnson St, Kingston, ON, Canada (pg. 6 of 7, starting pg. 124).
(f) \(|\Delta \hat{n}(1.0 \text{ m}, 10.0 \text{ m})| \geq 0.250: 146002\) points.

Figure C.8: DoN segmentation results on Intersection of Clergy and Johnson St, Kingston, ON, Canada (pg. 7 of 7, starting pg. 124).
Figure C.9: DoN segmentation results on Intersection of Princess and Bagot St, Kingston, ON, Canada (620820 points).
Figure C.10: DoN segmentation results on Intersection of Princess and Bagot St, Kingston, ON, Canada (pg. 2 of 7, starting pg. 131).
Figure C.10: DoN segmentation results on Intersection of Princess and Bagot St, Kingston, ON, Canada (pg. 3 of 7, starting pg. 131).
Figure C.10: DoN segmentation results on Intersection of Princess and Bagot St, Kingston, ON, Canada (pg. 4 of 7, starting pg. 131).
Figure C.10: DoN segmentation results on Intersection of Princess and Bagot St, Kingston, ON, Canada (pg. 5 of 7, starting pg. 131).
Figure C.10: DoN segmentation results on Intersection of Princess and Bagot St, Kingston, ON, Canada (pg. 6 of 7, starting pg. 131).
(f) $|\Delta n(1.0 \text{ m}, 10.0 \text{ m})| \geq 0.250$: 268746 points.

Figure C.10: DoN segmentation results on Intersection of Princess and Bagot St, Kingston, ON, Canada (pg. 7 of 7, starting pg. 131).
Appendix D

Supplemental Non-Ground Segmentation Results
Figure D.1: DoN non-ground segmentation results for Bagot St, Kingston, ON, Canada: 160989/478377 points (33%).
Figure D.2: DoN non-ground segmentation results for Intersection of Aberdeen and William St, Kingston, ON, Canada: 182334/523115 points (34%).
Figure D.3: DoN non-ground segmentation results for Intersection of Bagot and Brock St, Kingston, ON, Canada: 101935/484046 points (21%).
APPENDIX D. SUPPL. NON-GROUND SEGMENTATION RESULTS

142

(a) original: 614403 points.

(b) $|\Delta_n(0.2\text{ m}, 2.0\text{ m})| \geq 0.250$: 148287 points.

Figure D.4: DoN non-ground segmentation results for Intersection of Clergy and Johnson St, Kingston, ON, Canada: 148287/614403 points (24%).
Figure D.5: DoN non-ground segmentation results for Intersection of Princess and Bagot St, Kingston, ON, Canada: 212803/620820 points (34%).
Appendix E

Supplemental Curb Segmentation Results
APPENDIX E. SUPPL. CURB SEGMENTATION RESULTS

Figure E.1: DoN curb segmentation results on Bagot St, Kingston, ON, Canada: 1054/478377 points (0.22%).
Figure E.2: DoN curb segmentation results on Intersection of Aberdeen and William St, Kingston, ON, Canada: 964/523115 points (0.18%).
Figure E.3: DoN curb segmentation results on Intersection of Bagot and Brock St, Kingston, ON, Canada: 567/484046 points (0.11%).
APPENDIX E. SUPPL. CURB SEGMENTATION RESULTS

(a) original: 614403 points.

Figure E.4: DoN curb segmentation results on Intersection of Clergy and Johnson St, Kingston, ON, Canada: 1325/614403 points (0.21%).
Figure E.5: DoN curb segmentation results on Intersection of Princess and Bagot St, Kingston, ON, Canada: 984/620820 points (0.15%).