To Families, Teachers and Friends
ACKNOWLEDGMENTS

First of all, I thank my advisors, Dr. K. Clint Slatton and Dr. John G. Harris, for their great inspiration, support and guidance over my graduate studies. I was impressed by their active thoughts and appreciated very much their supervision which gave me a lot of opportunities to explore my research. I am also grateful to all the members of my advisory committee: Dr. Jose C. Principe, Dr. Dapeng O. Wu, and Dr. Bon A. Dewitt for their valuable time and interest in serving on my supervisory committee, as well as their comments, which helped improve the quality of this dissertation. I also express my appreciation to all the ASPL colleagues, especially Carolyn Krekeler, Karthik Nagarajan, Heezin Lee, Pang-Wei Liu, Kittipat Kampa, Hyun-Chong Cho, and Sweungwon Cheung for their help, collaboration and valuable discussions during my PhD study. Finally, I express my great love for my wife, Jui-Ling Yang and our lovely son, Andrew (Ren-Yu). I thank Jui-Ling for her love, caring, and patience, which made this study possible. Also, I am grateful to my parents for their great support for my life.
TABLE OF CONTENTS

ACKNOWLEDGMENTS ................................................................................................................ 4
LIST OF TABLES .......................................................................................................................... 8
LIST OF FIGURES ...................................................................................................................... 10
ABSTRACT ................................................................................................................................ 15
CHAPTER

1 INTRODUCTION .................................................................................................................. 18
   Airborne Light Detection and Ranging (LiDAR) Applications ................................................. 18
   Motivation ............................................................................................................................... 19
   Organization ............................................................................................................................. 20

2 PAPER SURVEYS ................................................................................................................. 22
   Bare-Earth Extraction .............................................................................................................. 22
      Morphological Filtering ........................................................................................................... 22
      Segmentation Modeling ......................................................................................................... 22
      Surface Modeling .................................................................................................................. 23
   Vehicle Detection ..................................................................................................................... 24
      Intelligent Transportation System (ITS) ............................................................................. 25
      Automatic Vehicle Guidance (AVG) ................................................................................ 25
      Traffic Flow Estimation ....................................................................................................... 26

3 TEST DATASETS ................................................................................................................ 27
   International Society for Photogrammetry and Sensing (ISPRS) Commission III ............. 27
      City Sites .............................................................................................................................. 27
      Forest Sites ......................................................................................................................... 27
   University of Florida (UF) Airborne Laser Swath Mapping (ALSM) System ................... 27
      City Site .............................................................................................................................. 28
      Forest Sites ......................................................................................................................... 28
      UF Campus Site .................................................................................................................. 28

4 TREE CANOPY REMOVAL DESIGN .............................................................................. 33
   Multiple Return Analysis ...................................................................................................... 33
   Morphological Filtering ....................................................................................................... 35
   Tree Canopy Point Decision ............................................................................................... 36
<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tree Canopy Removal Result</td>
<td>37</td>
</tr>
<tr>
<td>Summary</td>
<td>38</td>
</tr>
<tr>
<td><strong>5 POINT-BASED BARE-EARTH EXTRACTION DESIGN</strong></td>
<td>45</td>
</tr>
<tr>
<td>Segmentation Modeling</td>
<td>45</td>
</tr>
<tr>
<td>Triangle Classification and Assimilation</td>
<td>45</td>
</tr>
<tr>
<td>Edge Classification and Clustering</td>
<td>47</td>
</tr>
<tr>
<td>Point Classification</td>
<td>48</td>
</tr>
<tr>
<td>Surface Modeling</td>
<td>49</td>
</tr>
<tr>
<td>Phase-I Point Reclassification Based on Digital Terrain Model (DTM)</td>
<td>50</td>
</tr>
<tr>
<td>Phase-II Point Reclassification Based on Digital Surface Model (DSM)</td>
<td>51</td>
</tr>
<tr>
<td>Phase-III Point Reclassification Based on DTM Roughness</td>
<td>52</td>
</tr>
<tr>
<td>Phase-IV Point Reclassification Based on Flattened DSM</td>
<td>53</td>
</tr>
<tr>
<td>Point-Based Bared-Earth Extraction Result</td>
<td>54</td>
</tr>
<tr>
<td>Same Parameters for All 15 Sites of ISPRS</td>
<td>54</td>
</tr>
<tr>
<td>Optimized Parameters for All 15 Sites of ISPRS</td>
<td>57</td>
</tr>
<tr>
<td>Summary</td>
<td>58</td>
</tr>
<tr>
<td><strong>6 GRID-BASED BARE-EARTH EXTRACTION DESIGN</strong></td>
<td>75</td>
</tr>
<tr>
<td>Object Detection</td>
<td>75</td>
</tr>
<tr>
<td>Outlier and Tree Canopy Removal</td>
<td>75</td>
</tr>
<tr>
<td>Segmentation by Non-Flat Regions</td>
<td>76</td>
</tr>
<tr>
<td>Classification of flat regions</td>
<td>77</td>
</tr>
<tr>
<td>Statistical LiDAR Filtering</td>
<td>78</td>
</tr>
<tr>
<td>Chi-Distribution Measurement</td>
<td>78</td>
</tr>
<tr>
<td>Adaptive Height Threshold Derivation</td>
<td>79</td>
</tr>
<tr>
<td>Grid-Based Bared-Earth Extraction Result</td>
<td>80</td>
</tr>
<tr>
<td>Summary</td>
<td>80</td>
</tr>
<tr>
<td><strong>7 OCCLUDED VEHICLE DETECTION DESIGN</strong></td>
<td>94</td>
</tr>
<tr>
<td>Object Clustering</td>
<td>95</td>
</tr>
<tr>
<td>Horizontal Based Morphological Filtering</td>
<td>96</td>
</tr>
<tr>
<td>Vertical Based Morphological Filtering</td>
<td>97</td>
</tr>
<tr>
<td>Object Clustering Result</td>
<td>98</td>
</tr>
<tr>
<td>Object Classification</td>
<td>98</td>
</tr>
</tbody>
</table>
# LIST OF TABLES

<table>
<thead>
<tr>
<th>Table</th>
<th>page</th>
</tr>
</thead>
<tbody>
<tr>
<td>3-1</td>
<td>Special features of the city sites used in the International Society for Photogrammetry and Sensing (ISPRS) test</td>
</tr>
<tr>
<td>3-2</td>
<td>Special features of the forest sites used in ISPRS test</td>
</tr>
<tr>
<td>5-1</td>
<td>Extracted and reference cross matrix</td>
</tr>
<tr>
<td>5-2</td>
<td>Interpretation of the kappa index of agreement</td>
</tr>
<tr>
<td>5-3</td>
<td>Total errors and kappa index for 15 study sites with same parameters (TH1=1.6m and TH2=0.6m) in our filter</td>
</tr>
<tr>
<td>5-4</td>
<td>Total errors and kappa index of agreement for 15 study sites in our filter with two optimized parameters, TH1 and TH2.</td>
</tr>
<tr>
<td>6-1</td>
<td>Associated z positions in the slope formula</td>
</tr>
<tr>
<td>6-2</td>
<td>Kernel weights of the smoothing low-pass filter</td>
</tr>
<tr>
<td>6-3</td>
<td>Related positions of 8-connected neighbored pixels</td>
</tr>
<tr>
<td>6-4</td>
<td>Variable list for the 8-connected pixels</td>
</tr>
<tr>
<td>6-5</td>
<td>Total errors and kappa index for 15 study sites by the slope-based statistical algorithm (SSA)</td>
</tr>
<tr>
<td>6-6</td>
<td>Performance comparison among different methods</td>
</tr>
<tr>
<td>7-1</td>
<td>Summary of some morphological filtering operations</td>
</tr>
<tr>
<td>7-2</td>
<td>Vehicle detection cross matrix</td>
</tr>
<tr>
<td>7-3</td>
<td>Vehicle detection accuracy for Hogtown parking site with different Light Detection and Ranging (LiDAR) scans and overlapped all LiDAR scans</td>
</tr>
<tr>
<td>7-4</td>
<td>Vehicle detection accuracy for Hogtown forest site with different LiDAR scans and overlapped all LiDAR scans</td>
</tr>
<tr>
<td>7-5</td>
<td>Vehicle detection accuracy for Hogtown residential site with different LiDAR scans and overlapped all LiDAR scans</td>
</tr>
<tr>
<td>8-1</td>
<td>Confusion table of the envelope box classifier</td>
</tr>
<tr>
<td>8-2</td>
<td>Confusion table of the SVM (Support Vector Machine) classification by spin-image features</td>
</tr>
<tr>
<td>8-3</td>
<td>Confusion table of the SVM classification by features (blocking length, blocking width)</td>
</tr>
</tbody>
</table>
8-4 Confusion table of the SVM classification by features (Surface Intensity Index (SII), blocking area) ................................................................. 133
8-5 Confusion table of the SVM classification by features of the vehicle recognition #1 ...................................................................................... 133
8-6 Confusion table of the SVM classification by features of the vehicle recognition #2 ...................................................................................... 133
8-7 Confusion table of the SVM classification by features of SPI (Spin image, PCA, and Intensity) method ...................................................................... 133
8-8 Average vehicle detection performance of Vehicle Recognition #1 method with 10 simulations for each downsample rate ............................................................. 134
8-9 Average vehicle detection performance of Vehicle Recognition #2 method with 10 simulations for each downsample rate ............................................................. 134
8-10 Average vehicle detection performance of SPI method with 10 simulations for each downsample rate ........................................................................ 134
8-11 Vehicle detection accuracy for Hogtown parking site with different LiDAR scans and overlapped all LiDAR scans based on SVM with SPI features ................. 135
8-12 Vehicle detection accuracy for Hogtown forest site with different LiDAR scans and overlapped all LiDAR scans based on SVM with SPI features .................... 135
8-13 Vehicle detection accuracy for Hogtown residential site with different LiDAR scans and overlapped all LiDAR scans ........................................................................ 135
## LIST OF FIGURES

<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>3-1</td>
<td>The multiple-return training site on a Google Earth image (screen grab)</td>
<td>31</td>
</tr>
<tr>
<td>3-2</td>
<td>The forest test sites marked on a Google Earth image (screen grab)</td>
<td>31</td>
</tr>
<tr>
<td>3-3</td>
<td>Images for the University of Florida (UF) campus site</td>
<td>31</td>
</tr>
<tr>
<td>3-4</td>
<td>Normalized digital surface model (nDSM) in the UF campus site</td>
<td>32</td>
</tr>
<tr>
<td>4-1</td>
<td>Tree-canopy removal design flowchart</td>
<td>39</td>
</tr>
<tr>
<td>4-2</td>
<td>LiDAR (Light Detection and Ranging) multiple return shot illustration</td>
<td>39</td>
</tr>
<tr>
<td>4-3</td>
<td>Snapshot image of the site from the Google Earth</td>
<td>40</td>
</tr>
<tr>
<td>4-4</td>
<td>Histogram of shot-wise height differences at the site</td>
<td>40</td>
</tr>
<tr>
<td>4-5</td>
<td>Zoom in the smaller difference part of Figure 4-4</td>
<td>41</td>
</tr>
<tr>
<td>4-6</td>
<td>Shot-wise height difference map clipped to 0.0m-0.1m at the site</td>
<td>41</td>
</tr>
<tr>
<td>4-7</td>
<td>Multiple return map at the samp city site</td>
<td>42</td>
</tr>
<tr>
<td>4-8</td>
<td>Morphological filtering results at the samp city site</td>
<td>42</td>
</tr>
<tr>
<td>4-9</td>
<td>Detected vegetation area (canopy cell) marked on DSM of the city test site</td>
<td>43</td>
</tr>
<tr>
<td>4-10</td>
<td>Canopy removal result for the city test site</td>
<td>43</td>
</tr>
<tr>
<td>4-11</td>
<td>Revealed occluded vehicles at the Hogan town parking site</td>
<td>43</td>
</tr>
<tr>
<td>4-12</td>
<td>Revealed occluded vehicles at the Hogan town forest site</td>
<td>44</td>
</tr>
<tr>
<td>4-13</td>
<td>Revealed occluded vehicles at the Hogan town residential site</td>
<td>44</td>
</tr>
<tr>
<td>4-14</td>
<td>Remained point density in 2-D distribution after removing canopy points</td>
<td>44</td>
</tr>
<tr>
<td>5-1</td>
<td>Segmentation modeling flowchart of our bare-earth extraction design</td>
<td>60</td>
</tr>
<tr>
<td>5-2</td>
<td>Triangle classification and assimilation illustration</td>
<td>60</td>
</tr>
<tr>
<td>5-3</td>
<td>Triangle assimilation curve of samp city site</td>
<td>60</td>
</tr>
<tr>
<td>5-4</td>
<td>Cluster-wise ground ratio illustration</td>
<td>61</td>
</tr>
<tr>
<td>5-5</td>
<td>Vector-based edge classification</td>
<td>61</td>
</tr>
<tr>
<td>5-6</td>
<td>Cluster-wise boundary edge detection</td>
<td>61</td>
</tr>
<tr>
<td>5-7</td>
<td>Initial point classification result</td>
<td>62</td>
</tr>
<tr>
<td>5-8</td>
<td>Surface modeling flowchart of our bare-earth extraction design</td>
<td>62</td>
</tr>
<tr>
<td>5-9</td>
<td>Phase-I point reclassification result</td>
<td>63</td>
</tr>
<tr>
<td>5-10</td>
<td>Roughness map of canopy removal DSM</td>
<td>63</td>
</tr>
<tr>
<td>5-11</td>
<td>Phase-II point reclassification result</td>
<td>63</td>
</tr>
<tr>
<td>5-12</td>
<td>Roughness map of DTM (Digital Terrain Model)</td>
<td>64</td>
</tr>
</tbody>
</table>
5-13 Bridge detection result ........................................................................................................ 64
5-14 Point reclassification phase-III result ................................................................................ 64
5-15 The flattened DSM map ...................................................................................................... 65
5-16 Final point classification result .......................................................................................... 65
5-17 DSM image of the city site #1 ............................................................................................ 65
5-18 The final classification errors in the site of samp11 ............................................................. 66
5-19 The final classification errors in the site of samp12 ............................................................. 66
5-20 DSM image of the city site #2 ............................................................................................ 66
5-21 The final classification errors in the site of samp21 ............................................................. 67
5-22 The final classification errors in the site of samp22 ............................................................. 67
5-23 The final classification errors in the site of samp23 ............................................................. 67
5-24 The final classification errors in the site of samp24 ............................................................. 68
5-25 DSM image of the city site #3 ............................................................................................ 68
5-26 The final classification errors in the site of samp31 ............................................................. 68
5-27 DSM image of the city site #4 ............................................................................................ 69
5-28 The final classification errors in the site of samp41 ............................................................. 69
5-29 The final classification errors in the site of samp42 ............................................................. 69
5-30 DSM image of the forest site #5 ........................................................................................ 70
5-31 The final classification errors in the site of samp51 ............................................................. 70
5-32 The final classification errors in the site of samp52 ............................................................. 70
5-33 The final classification errors in the site of samp53 ............................................................. 71
5-34 The final classification errors in the site of samp54 ............................................................. 71
5-35 DSM image of the forest site #6 ........................................................................................ 71
5-36 The final classification errors in the site of samp61 ............................................................. 72
5-37 DSM image of the forest site #7 ........................................................................................ 72
5-38 The final classification errors in the site of samp71 ............................................................. 72
5-39 Accuracy comparison to the best two filters for all 15 study sites ...................................... 73
5-40 Accuracy comparison of our automated filter to optimized filter ...................................... 73
5-41 Average accuracy comparison to eight different filters before 2004 .................................. 74
5-42 Average accuracy comparison to four different filters proposed after 2004 ...................... 74
6-1 Standard deviation and confidence interval of a normal distribution .................................. 84
6-2 Outlier points found by a normal distribution at site22 ..................................................... 84

11
6-3 Edge detection at the site22 ................................................................. 84
6-4 Obtained flat regions at the site22 ........................................................... 85
6-5 Rule-based methods for flat region classification ...................................... 85
6-6 The associated feature maps and classification result of the site22 .......... 86
6-7 DTM comparison at the site22 ................................................................. 87
6-8 DTM comparison at the site11 ................................................................. 87
6-9 DTM comparison at the site12 ................................................................. 87
6-10 DTM comparison at the site21 ................................................................. 88
6-11 DTM comparison at the site23 ................................................................. 88
6-12 DTM comparison at the site24 ................................................................. 88
6-13 DTM comparison at the site31 ................................................................. 89
6-14 DTM comparison at the site41 ................................................................. 89
6-15 DTM comparison at the site42 ................................................................. 89
6-16 DTM comparison at the site51 ................................................................. 90
6-17 DTM comparison at the site52 ................................................................. 90
6-18 DTM comparison at the site53 ................................................................. 90
6-19 DTM comparison at the site54 ................................................................. 91
6-20 DTM comparison at the site61 ................................................................. 91
6-21 DTM comparison at the site71 ................................................................. 91
6-22 Histograms of slope differences from the initial terrain at the site22 ........ 92
6-23 Adaptive slope difference threshold algorithm based on Chi distribution .... 93
6-24 The change curve of freedom at the site22 .............................................. 93
6-25 Histogram of remaining slope differences at the site22 ......................... 93
7-1 Object segmentation flowchart ............................................................... 113
7-2 Object segmentation result in Hogtown parking site .................................. 113
7-3 Object segmentation result in Hogtown forest site .................................... 113
7-4 Object segmentation result in the residential site ...................................... 114
7-5 Object classification flowchart ............................................................... 114
7-6 Using Principal Component Analysis (PCA) to get the best fitting plane .... 115
7-7 An unoccluded vehicle consisting of 311 LiDAR points ......................... 115
7-8 An occluded vehicle consisting of 82 LiDAR points ............................... 115
7-9 An occluded vehicle consisting of 73 LiDAR points ............................... 116
<table>
<thead>
<tr>
<th>Chapter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>7-10</td>
<td>An occluded vehicle consisting of 69 LiDAR points</td>
</tr>
<tr>
<td>7-11</td>
<td>An occluded vehicle consisting of 67 LiDAR points</td>
</tr>
<tr>
<td>7-12</td>
<td>An occluded vehicle consisting of 41 LiDAR points</td>
</tr>
<tr>
<td>7-13</td>
<td>An occluded vehicle consisting of 14 LiDAR points</td>
</tr>
<tr>
<td>7-14</td>
<td>An occluded vehicle consisting of 12 LiDAR points</td>
</tr>
<tr>
<td>7-15</td>
<td>An occluded vehicle consisting of 8 LiDAR points</td>
</tr>
<tr>
<td>7-16</td>
<td>An occluded vehicle consisting of 7 LiDAR points</td>
</tr>
<tr>
<td>7-17</td>
<td>Bivariate Probability Density Function (PDF) of reference vehicles</td>
</tr>
<tr>
<td>7-18</td>
<td>Bivariate PDF of reference non-vehicles</td>
</tr>
<tr>
<td>7-19</td>
<td>The information divergence for vehicles and non-vehicles</td>
</tr>
<tr>
<td>7-20</td>
<td>Bivariate PDF of detected vehicles with BT=0.7</td>
</tr>
<tr>
<td>7-21</td>
<td>Bivariate PDF of detected non-vehicles with BT=0.7</td>
</tr>
<tr>
<td>7-22</td>
<td>Occluded vehicle detection results in the Hogtown parking site</td>
</tr>
<tr>
<td>7-23</td>
<td>Occluded vehicle detection results in the Hogtown forest site</td>
</tr>
<tr>
<td>7-24</td>
<td>Occluded vehicle detection results in the Hogtown residential site</td>
</tr>
<tr>
<td>7-25</td>
<td>Vehicle detection performance vs. the number of collected LiDAR points</td>
</tr>
<tr>
<td>7-26</td>
<td>Vehicle detection accuracy vs. the number of collected LiDAR points</td>
</tr>
<tr>
<td>7-27</td>
<td>Receiver Operator Characteristics (ROC) curves of our detection performance</td>
</tr>
<tr>
<td>8-1</td>
<td>LiDAR point density histogram of reference vehicles in the UF campus site</td>
</tr>
<tr>
<td>8-2</td>
<td>A DSM and LiDAR point map example</td>
</tr>
<tr>
<td>8-3</td>
<td>Spin image of Figure 8-2</td>
</tr>
<tr>
<td>8-4</td>
<td>SVM (Support Vector Machine) classification result by Spin-image features</td>
</tr>
<tr>
<td>8-5</td>
<td>Length and width estimation by PCA in Figure 8-2</td>
</tr>
<tr>
<td>8-6</td>
<td>SVM classification result by blocking length, blocking width features</td>
</tr>
<tr>
<td>8-7</td>
<td>Intensity Map gridded from LiDAR data at one of the UF campus site</td>
</tr>
<tr>
<td>8-8</td>
<td>Point relationship with the convex hull</td>
</tr>
<tr>
<td>8-9</td>
<td>SVM classification result by surface intensity index and blocking area features</td>
</tr>
<tr>
<td>8-10</td>
<td>The six-parameter representation of the vehicle recognition #1 method</td>
</tr>
<tr>
<td>8-11</td>
<td>Contained information of PCA of the vehicle recognition #1 method</td>
</tr>
<tr>
<td>8-12</td>
<td>Contained information of PCA of the vehicle recognition #2 method</td>
</tr>
<tr>
<td>8-13</td>
<td>Contained information of PCA of the novel features</td>
</tr>
<tr>
<td>8-14</td>
<td>Average recognition rate of vehicle detection vs. downsample rate</td>
</tr>
</tbody>
</table>
8-15  Average Kappa index of agreement of vehicle detection vs. downsample rate .......... 141
8-16  Contained information of principal components of the SPI (Spin image, PCA, and Intensity) method applied to Hogtown forest sites ......................................................... 142
Although vehicle detection has been developed for Intelligent Transportation System (ITS), Automatic Vehicle Guidance (AVG), and traffic flow estimation in LiDAR (Light Detection and Ranging) applications, it has not been exploited in cluttered environments such as forested terrain. State-of-the-art airborne LiDAR can provide data in large spatial extents with varying temporal resolution and can be deployed more or less anywhere and anytime, even in cloudy weather and at night. Thus, occluded vehicle detection in forested terrain from airborne LiDAR data can be potentially applied to many fields such as military surveillance, homeland security, global warming, disaster rescue, emergency road service, and criminal searching. In this study, we finished a system with a goal to detect vehicles underneath canopy in forested terrain from LiDAR point clouds. This system covers three important parts and is described as follows.

First, the thermal imaging cameras can see the heat signature of people, boats, and vehicles in total darkness as well as through smoke, haze, and light fog, but not through forest canopy. LiDAR employs an active optical modality or laser ranging that provides primarily geometric information to detect natural surface features and other hard targets that may be spectrally inseparable in multi-spectral passive optical imagery. Accordingly, the first part of this work is that we developed a novel algorithm to help detect obscure targets underneath forest canopy and mitigate the vegetation problem for bare ground point extraction filters as well. By examining the processed results, the forest canopy was successfully
removed and all obscure vehicles or buildings underneath canopy can be easily seen. The occluded rate or transparency of forest canopy and the detailed underneath x-y point distribution can be easily obtained accordingly which is very useful for predicting the performance of occluded target detection with respect to various object locations.

For the second part of this work, although a variety of algorithms have been developed for extracting the digital terrain model (DTM) from point clouds generated by LiDAR systems, most filters perform well in flat and uncomplicated landscapes, while landscapes containing steep slopes and discontinuities are still problematic. Therefore, we designed a novel bare-Earth extraction algorithm including morphological filtering, segmentation modeling, and surface modeling to automatically classify ground points and non-ground points from LiDAR point clouds. The obtained filtering result has been compared to twelve filters working on the same fifteen study sites which were provided by the ISPRS (International Society for Photogrammetry and Sensing). The average total error and kappa index of agreement of this work in the automated process is 4.6% and 84.5%, respectively, which outperforms all other twelve proposed filters. In addition, we develop another novel slope-based statistical algorithm which is appropriate for any mixed or complicated terrain types. Initially, most objects are removed and initial terrains can be obtained in our object detection algorithm. Slope differences can be assumed to be from a zero-mean normal distribution in all kinds of terrains. Based on slope difference variations, the Chi distribution measurement is used to decide the adaptive slope threshold. Accordingly, the adaptive growing height threshold of each pixel is derived by 8-connected neighbored pixels which can be used to iteratively correct classified points in the initial terrain. The testing results show that this algorithm is even better than our previous algorithm which has outperformed all other twelve algorithms working on the same study sites.

In the last part, the obtained canopy-free LiDAR points are clustered into individual objects by the proposed bare-earth extraction algorithm and associated morphological image processing. The clustered LiDAR points of each object are analyzed and classified into the vehicle class or non-vehicle class by many theories such as Spin image, non-parametric Parzen-window estimation, Bayesian decision, and
relative entropy. Finally, the results are demonstrated, discussed, and verified by the Receiver Operating Characteristic (ROC) curve. In addition, we propose another occluded vehicle detection approach, which combines five features extracted from Spin image, PCA (Principal Component Analysis), and LiDAR Intensity (SPI) and applies them to the Support Vector Machine (SVM) classifier. This SPI (Spin image, PCA, and Intensity) method is compared to a simple method and two other vehicle detection methods proposed by other authors’ papers published by ISPRS. With ten simulations each in different downsampling rates, testing on independent 580 vehicles and 580 non-vehicle objects, our experiments show that this SPI method outperforms all other methods, especially in low sampling rates.
CHAPTER 1
INTRODUCTION

Airborne light detection and ranging (LiDAR) technology is an active remote sensing technology which allows accurate measurements of topography, vegetation canopy heights, and buildings over large areas. Most modern ALTM systems consist of three basic components: the laser scanner, a kinematic Global Position System (GPS), and the Inertial Measurement Unit (IMU). The laser scanner detects the range from aircraft to ground by recording the time difference between laser pulses sent out and reflected back. In this study, we used LiDAR point clouds to develop an automatic canopy removal algorithm and a novel bare-Earth extraction algorithm to reveal those LiDAR points underneath forest canopy and filter ground points, respectively. Accordingly, the goal of occluded vehicle detection in forested terrain can be continually exploited and achieved.

Airborne LiDAR Applications

In the last decade, advances in airborne LiDAR technologies have enabled dramatic improvements in topographic mapping resolution, particularly for ground surfaces beneath vegetation canopies [1], [2]. LiDAR data analyses have proven successful for a variety of forested terrain applications, such as fusing multi-resolution elevation measurements [3], estimation of bare-surface topography [4], [5], extraction of micro-stream channels beneath forest canopies [6], estimation of sunlight flux in forest understories [7], segmentation of individual tree canopies [8], and stand-level forest parameter estimation [9].

A DTM (Digital Terrain Model), commonly used interchangeably with a digital elevation model (DEM), is a digital representation consisting of terrain elevations for ground positions. A DTM is also called a bare-Earth model since it excludes features on the Earth such as tall vegetation, buildings, and bridges. A DTM is generally acquired using field-based land surveying methods, photogrammetric techniques, or through processing remotely sensed data. Airborne LiDAR, an active remote sensing technology, has revolutionized the acquisition of digital elevation data for large-scale mapping applications and has become a fixture of present-day mapping missions, providing cost-effective means to achieve high-accuracy results.
Accordingly, Airborne LiDAR can produce detailed maps of the ground more speedily and, in many cases, more economically than almost any other method. Using LiDAR will result in a DTM of the ground as well as the top (height) of the vegetation so that many features can also be determined, especially in conjunction with standard or small frame aerial photography (such as access roads, structures and water courses). The result is that these data can be used to help assess stands and timber volume, plan roads, review slope stability, run-off and other critical data about any area which are subject to flood and drainage modeling, land-use studies, geological applications, as well as urban planning and management.

**Motivation**

ISPRS (International Society for Photogrammetry and Sensing) Working Group III/3 conducted a test [10] and found that all bare ground point extraction filter algorithms perform well on LiDAR point clouds from smooth rural landscapes, but all produce errors in rough terrain with vegetation canopy. Besides, the thermal imaging cameras can see the heat signature of people, boats, and vehicles in total darkness as well as through smoke, haze, and light fog, but not through forest canopy. LiDAR employs an active optical modality or laser ranging that provides primarily geometric information to detect natural surface features and other hard targets that may be spectrally inseparable in multi-spectral passive optical imagery. Therefore, the first part of this work is to develop a novel algorithm to mitigate the vegetation problem for those ground point extraction filters and help detect obscure targets underneath forest canopy as well.

In the last decade, a variety of algorithms have been developed for extracting DTMs from point clouds generated by LiDAR systems, where automatic and robust ground point extraction has been attracting a great deal of attention. Three kinds of approaches for LiDAR filtering are most prevalent: morphological filtering, segmentation modeling, and surface modeling. Although most filters perform well in flat and uncomplicated landscapes, landscapes containing steep slopes and discontinuities are still problematic. Hence, the second part of this work is to design a novel bare-earth extraction algorithm.
which can combine those three prevalent approaches and to achieve better performance than other existing algorithms.

In addition, airborne LiDAR can provide data in large spatial extents with varying temporal resolution and it can be deployed more or less anywhere and at anytime, including in smoke, haze, fog, and nighttime, which restrict the use of passive optical imagery. Thus, occluded vehicle detection from airborne LiDAR data in forested terrain, the third part of this work, can be applied to many fields, more specifically: 1) military surveillance – searching for enemy vehicles in a battle area with forest, 2) homeland security – border crossing monitoring for vehicles in forest area, 3) global warming – vehicle hunting for illegal deforestation which is a hidden cause of global warming, 4) disaster rescue – finding vehicles stuck by disrupted roads in forest during natural disaster, 5) emergency road service – locating vehicles involved with general car accidents in forest, and 6) criminal searching – uncovering forest canopy to search suspicious vehicles hiding in mountains.

Organization

This dissertation is organized as follows. First we review some papers associated with this study which include bare-Earth extraction and vehicle detection. Morphological filtering, segmentation modeling, and surface modeling, which are three more prevalent approaches for LiDAR filtering algorithms, are briefly discussed. Intelligent transportation system (ITS), automatic vehicle guidance (AVG), and traffic flow estimation, which are related to vehicle detection research, are introduced and summarized. In Chapter 3, we explain test datasets of LiDAR point clouds in this study which were provided by ISPRS Commission III and UF (University of Florida) Airborne Laser Swath Mapping (ALSM) system. Both city sites and forest sites are investigated. In Chapter 4, we show our tree canopy removal design which consists of multiple return analysis, morphological filtering, and tree canopy point decision. The results are also demonstrated and discussed. In Chapter 5, we show our novel bare-earth extraction design which is composed of two sections. The first section, segmentation modeling, is formed by triangle classification and assimilation, edge classification and clustering, and point classification. The second section, surface modeling, is constructed by phase-I point reclassification based on DTM,
phase-II point reclassification based on DSM (Digital Surface Model) roughness, phase-III point reclassification based on DTM roughness, and phase-IV point reclassification based on flattened DSM. The results are also demonstrated and discussed. In addition, a slope-based statistical bare-Earth extraction algorithm is proposed in Chapter 6. Object detection is achieved by slope relationships between clustered points and morphological filtering. With detected and removed object points, initial ground points are obtained and reclassified by comparing the above ground level to the adaptive height threshold derived from Chi distribution measurement. The occluded vehicle detection designs are exploited in Chapter 7 and Chapter 8. In Chapter 7, the object clustering is completed by horizontal based and vertical based morphological filtering, where the results are also presented. Object classification is followed and achieved by Spin image, non-parametric Parzen-window estimation, Bayesian decision, and relative entropy. In Chapter 8, we propose to extract features from Spin image, principal component analysis, and LiDAR intensity and apply them to the support vector machine for detecting vehicles. The testing database includes 580 independent vehicles in open area and 580 independent non-vehicles in cluttered and occluded environment. The occluded vehicle detection is simulated by detecting those objects from randomly downsampling LiDAR points of vehicles mixing with cluttered objects. Finally, Chapter 9 is our conclusions and contributions.
CHAPTER 2
PAPER SURVEYS

Bare-Earth Extraction

There are many existing approaches for filtering LiDAR to classify LiDAR points into ground and object points for DTM generation and further reconstruction of topographic features. Three kinds of approaches for LiDAR filtering are more prevalent: morphological filtering, segmentation modeling, and surface modeling. Although most filters perform well in flat and uncomplicated landscapes, landscapes containing steep slopes and discontinuities are still problematic.

Morphological Filtering

In this first kind of approach for LiDAR filtering, the structure element of mathematical morphological filtering was used by Vosselman [11], in which the admissible height difference for ground points depends on the horizontal distance between a ground point and its neighboring points. A larger neighboring horizontal distance will allow a greater height difference among accepted ground points. This structure element is placed at each point, so ground points are identified as those that fall below the admissible height difference. Using multiple structure elements was also proposed by Kilian [12], where the likelihood of points as ground is weighted by each window size and then ground points are determined accordingly in the final classification. Zhang [13] proposed a progressive morphological filter to detect non-ground features. By gradually increasing the window size of the filter and using elevation difference thresholds, the measurements of vehicles, vegetation, and buildings are removed, while ground data are preserved. A similar concept is a slope-based method proposed by Sithole [14], Roggero [15], and Kampa [16]. The slope or height difference between two points is measured. If that measurement exceeds a certain threshold, the higher point is assumed to belong to an object point.

Segmentation Modeling

In this second kind of approach for LiDAR filtering, points are aggregated into segments by the geometric relationship of neighborhoods based on height, slope or curvature difference. Sithole [17] proposed a method for object segmentation, where a point cloud is sliced into parallel vertical profiles
and aggregated based on proximity to determine the adjacency of surface segments. The purpose of segmentation is to obtain higher level information from points in a point cloud, which is usually knowledge of the extent of homogeneous regions in a landscape, e.g., buildings, vegetation, and bridges. Nardinocchi [18] presented a strategy for the classification of raw LIDAR data as terrain, buildings, and vegetation. Guided by that classification, the size and relationship of adjacent segmented objects are analyzed in order to get final point classification. Filin [19] proposed a clustering analysis in which the position, the best fitting plane parameters, and height difference of neighboring points are used. Vosselman [20] also proposed the use of the Hough transformation to detect planar roof surfaces within the given building boundaries for clustering. In Brovelli [21], it was proposed that any points belonging to an object must be segmented as a cluster if they are above its neighborhood height differences. The primary objectives for segmentation [17] are to obtain better discrimination of large objects in a landscape, preserve discontinuities, and allow both Type I and II errors to be reduced instead of making a trade off between them.

**Surface Modeling**

Finally, the filters in the third group are based on a surface model, where the entire point set is updated iteratively to approach the ground surface. The residuals of all points are measured by the relation between their height levels and a surface model. Measured points that lie above a surface should have less influence on the shape of the surface, while points lying below should have more influence. In Kraus’s paper [22], this surface runs in an averaging way between terrain points and vegetation points. The terrain points are more likely to have negative residuals, whereas the vegetation points are more likely to have small negative or positive residuals. Elmqvist [23] used an active shape model, also referred to as snakes, for detection of contours in images, where the inner forces of the surface determine its stiffness and the external forces are a negative gravity. Iteration starts with a horizontal surface below all points that moves upward to reach the point, but inner stiffness prevents it from reaching up to the points on vegetation or house roofs. Axelsson [24] used a sparse TIN derived from neighborhood minima as the first reference surface. In each iteration, if a point is found with offsets below given threshold
values, it is classified as a ground point and added to the TIN such that the TIN is progressively densified. The iterative process stops when no more points are below the threshold.

However, an experimental study of some filtering algorithms was described by Sithole in 2004 [25], where their performances were compared based on the results of filtering the same LiDAR data sets provided by ISPRS. The comparison results determined that in flat and uncomplicated landscapes (i.e., small to medium sized buildings standing well off a fairly flat ground) algorithms tend to do well. Significant differences in accuracies of filtering appear in landscapes containing steep slopes and discontinuities. These differences are a result of the ability of algorithms to preserve terrain discontinuities while detecting large objects.

After the year 2004, Sithole [26], Silvan-Cardenas [27], Lu [28], and Meng [29] also used the same data sets to develop new filtering algorithms. However, these filters perform insignificantly worse than the filter developed by Axelsson in 2000 [24]. In this study, we present a novel filter composed of tree canopy removal design, bare-earth extraction design, and point reclassification design, which includes features of morphological filtering, segmentation modeling, and surface modeling, respectively, described in the above three filtering groups. The final filtering results show that our automated process works better than Axelsson’s filter, and the performance is even better after two dominant parameters are optimized manually.

**Vehicle Detection**

Although vehicle detection has been utilized in Intelligent Transportation System (ITS), Automatic Vehicle Guidance (AVG), and traffic flow estimation, it has not been exploited in forested terrain. Airborne LiDAR can provide data in large spatial extents with varying temporal resolution and it can be deployed more or less anywhere and anytime, including in smoke, haze, fog, and nighttime, which restrict the use of passive optical imagery. Thus, occluded vehicle detection from airborne LiDAR data in forested terrain can be applied to many fields, more specifically: 1) military surveillance – searching for enemy vehicles in a battle area with forest, 2) homeland security – border crossing monitoring for vehicles in forest area, 3) global warming – vehicle hunting for illegal deforestation which is a hidden
cause of global warming, 4) disaster rescue – finding vehicles stuck by disrupted roads in forest during natural disaster, 5) emergency road service – locating vehicles involved with general car accidents in forest, and 6) criminal searching – uncovering forest canopy to search suspicious vehicles hiding in mountains.

**Intelligent Transportation System (ITS)**

Vehicle detection has been applied to Intelligent Transportation System and its summary can be reviewed in [30]. The associated technologies for vehicle detection can be categorized into intrusive sensors and non-intrusive sensors which depend on whether the sensors are required to be installed directly onto or into the road surface or not. The intrusive sensors include the pneumatic road tube [31], inductive loop [32], piezoelectric cable [33], and magnetic sensors [34], [35], [36] and weigh-in-motion (WIM) systems [37]. The non-intrusive sensors are mounted overhead or on the side of the roadway, such as the video image processor [38], [39], [40], microwave radar [41], active and passive infrared sensors [40], ultrasonic sensors [42], and passive acoustic array sensors [30]. However, both the intrusive sensors and non-intrusive sensors need installation in appropriate spots which also restrains the ability to detect vehicles everywhere.

**Automatic Vehicle Guidance (AVG)**

Vehicle detection with vision-based methods has been devoted to Automatic Vehicle Guidance (AVG) [43]. A review of on-road vehicle detection can be found in [44]. So far, several prototypes and solutions have been produced based on different approaches [45], [46], [47], [48]. Looking at research on intelligent vehicles worldwide, Europe pioneers the research, followed by Japan and the United States. Two kinds of sensors, passive type and active type, are mounted on the experimental vehicle and exploited to the approach of vehicle detection. For the passive sensors [49], such as normal cameras, their drawbacks are that they are too susceptible to shadows, occlusion, day-to-night transition, and inclement weather which restrict these kinds of sensors to be used at any time. For the active sensors [49], such as radar-based [50], laser-based (i.e. LiDAR) [51], [52], and acoustic-based [53], their performances in fog, rain, or snow are better than passive sensors. But, a big problem is posed by the interference
among the same types of sensors when a large number of vehicles is moving simultaneously in the same direction. However, that is not a problem for using airborne LiDAR to detect vehicles since the mission for searching vehicles in the region of interest can be completed by the divide and conquer strategy [54].

**Traffic Flow Estimation**

Recently, vehicle detection with airborne LiDAR data is exploited to support traffic flow estimation [55], [56], [57], [58], [59]. In their system architecture, the main processing steps are road surface extraction, vehicle extraction, primary parameterization of vehicle shape, feature space selection, vehicle classification, vehicle velocity estimates, and traffic flow data computation. However, the road area has to be filtered in the beginning for their following vehicle extraction process. Although the road surface modeling can be derived from airborne LiDAR data [56], they still need the GIS or CAD database to assist in extracting the road edges and road medians. Therefore, if a vehicle is being driven or parked on some road or place which information is out of date or unavailable in the associated database, it will cause missing errors for vehicle detection. In addition, the detection for those occluded vehicles which are underneath the trees or forest is out of their scope.
ISPRS Commission III

City Sites

ISPRS Commission III/WG3 provides LiDAR data sets including both urban landscapes (city sites) and rural landscapes (forest sites) which were acquired by an Optech Airborne Laser Terrain Mapper (ALTM) scanner over the Vaihingen/Enz test field and the Stuttgart city center [57]. The average point density for the city sites is roughly 0.67 points per square meter. Every record in the raw data represents three-dimensional coordinates and intensity of first and last returned points, while each record in the reference data has been compiled by semi-automatic filtering and manual editing as either a ground or non-ground point. These areas were chosen because of their diverse feature content such as vegetation, buildings, roads, railroads, ramps, bridges, power lines, etc. Some special features of these sites are listed in Table 3-1.

Forest Sites

The average point density for the forest sites is roughly 0.18 points per square meter. Every record in the raw data also represents three-dimensional coordinates and intensity of first and last returned points. Each record in the reference data has also been compiled by semi-automatic filtering and manual editing as either a ground or non-ground point. These areas include diverse feature content such as vegetation, rivers, ridges, buildings, etc. Some special features of these sites are listed in Table 3-2. The forest site 8 was not used in this research since no reference data was available.

UF ALSM System

The UF airborne LiDAR system, purchased in 2007, is a state-of-the-art commercial LiDAR (Optech Gemini) offering laser pulse rates of up to 167 kHz, with up to 4 returns per transmitted pulse, recorded intensities of each return, and multiple beam divergences [58]. For this study, the LiDAR data sets were collected in Dec 2008 by the Optech Gemini operated at 125 KHz pulse rate, 45 Hz scan frequency, 0.3 mrad beam divergence, 213 Km/h airplane speed, 0 to ±15 scan angle, and 600m Above
Ground Level (AGL) for 14 repeat passes around the Hogtown area and UF campus with vertical
accuracy of about 15 cm and the average LiDAR density is 3.52 shots per square meter.

City Site

In order to further investigate and verify the multiple-return characteristic of airborne LiDAR data,
the site close to the UF campus was introduced and collected by the UF airborne LiDAR system. This
site’s image snapshot from Google Earth is shown in Figure 3-1. This area was chosen because of its
diverse feature content including uncovered ground, still vehicles, light poles, buildings, and isolated
trees. Besides, the average LiDAR point density is 4.44 points per square meter which can give better
rendering resolution than the data set provided by ISPRS. Since the edges of small or large objects and
discontinuity of trees have less probability to be scanned and reflected, the higher LiDAR density can
provide more detailed information and make the multiple-return analysis easier.

Forest Sites

Three distinct forest sites in Hogtown area were collected by the UF airborne LiDAR system and
used in this research. The Hogtown area is a mixed warm temperate forest composed of roughly 80
percent deciduous and 20 percent coniferous trees, but this mix varies significantly with location.
Hogtown consists of trees as tall as 35 meters and possesses the characteristics expected in a natural
forest where there are multiple layers of foliage, varying and random spacing between trees, and
significant undergrowth. The locations of 3 survey sites (Hogtown forest site, Hogtown parking site, and
a residential site) close to the UF campus are marked on Google Earth and labeled in Figure 3-2.

UF Campus Site

The LiDAR scanning data of one flight around the north-west corner of the UF campus is selected
for downsampled vehicle detection simulation. The photo image, LiDAR digital surface model (DSM)
and DTM of this area are showed in Figure 3-3. We are interested in this area because it covers many
unoccluded vehicles and trees. In order to see those objects clearer in this area, the partial data in a
parking lot and a residential site are cropped which are marked by squares in the Figure 3-3b and Figure
3-3c. The normalized digital surface models (nDSM) are shown in Figure 3-4 which are calculated by using their DSM minus DTM to get relative heights of those objects.
Table 3-1. Special features of the city sites used in ISPRS test

<table>
<thead>
<tr>
<th>Site</th>
<th>Reference data</th>
<th>Special features</th>
</tr>
</thead>
<tbody>
<tr>
<td>City site1</td>
<td>Samp11</td>
<td>Buildings on steep slopes</td>
</tr>
<tr>
<td>City site1</td>
<td>Samp12</td>
<td>Vegetation mixed with buildings</td>
</tr>
<tr>
<td>City site2</td>
<td>Samp21</td>
<td>Bridge and large building</td>
</tr>
<tr>
<td>City site2</td>
<td>Samp22</td>
<td>Bridges &amp; terrain discontinuities</td>
</tr>
<tr>
<td>City site2</td>
<td>Samp23</td>
<td>Irregularly shaped buildings</td>
</tr>
<tr>
<td>City site2</td>
<td>Samp24</td>
<td>Ramp &amp; terrain discontinuities</td>
</tr>
<tr>
<td>City site3</td>
<td>Samp31</td>
<td>Large buildings</td>
</tr>
<tr>
<td>City site4</td>
<td>Samp41</td>
<td>Data gaps and outliers</td>
</tr>
<tr>
<td>City site4</td>
<td>Samp42</td>
<td>Railway station with trains</td>
</tr>
</tbody>
</table>

Table 3-2. Special features of the forest sites used in ISPRS test

<table>
<thead>
<tr>
<th>Site</th>
<th>Reference Data</th>
<th>Special Features</th>
</tr>
</thead>
<tbody>
<tr>
<td>Forest site 5</td>
<td>Samp51</td>
<td>Vegetation on steep slopes</td>
</tr>
<tr>
<td>Forest site 5</td>
<td>Samp52</td>
<td>Vegetation on river bank</td>
</tr>
<tr>
<td>Forest site 5</td>
<td>Samp53</td>
<td>Terrain discontinuities</td>
</tr>
<tr>
<td>Forest site 5</td>
<td>Samp54</td>
<td>Low resolution buildings</td>
</tr>
<tr>
<td>Forest site 6</td>
<td>Samp61</td>
<td>Sharp ridge and embankments</td>
</tr>
<tr>
<td>Forest site 7</td>
<td>Samp71</td>
<td>Ramp, bridge, and underpass</td>
</tr>
</tbody>
</table>
Figure 3-1. The multiple-return training site on a Google Earth image (screen grab)

Figure 3-2. The forest test sites marked on a Google Earth image (screen grab)

Figure 3-3. Images for the UF campus site. A) Snapshot image from Google Earth, B) LiDAR DSM, and C) LiDAR DTM.
Figure 3-4. Normalized digital surface model in the UF campus site: A) nDSM of the parking lot and B) nDSM of the urban forest.
CHAPTER 4
TREE CANOPY REMOVAL DESIGN

This chapter is the first design part of our work. First, the multiple-return characteristic of LiDAR data is analyzed and accordingly laser shots are classified as single-return or multiple-return shots. The major challenge of removing canopy is that some foliage will unexpectedly reflect single-return shots rather than normal multiple-return shots when they are very dense. This challenge can be solved by using our developed algorithms such as analyzing distance relationships between foliage, applying morphological filters to process the canopy/non-canopy image and creating rough digital terrain models to calculate above ground levels of points, etc.

The unique feature of this approach is that no parameter tweaking is required. Both of the city and forest sites are tested where the data are from ISPRS and UF, respectively. It shows that all tree or forest canopy points have been removed such that all obscure vehicles or buildings underneath canopy can now be easily seen. For military application, although thermal imaging cameras can see the heat signature of people, boats, and vehicles in total darkness as well as through smoke, haze, and light fog, they cannot be used to see through the forest canopy, unlike our algorithm. The canopy removal algorithm, a block diagram of which is shown in Figure 4-1, takes the multiple-return feature to classify raw point clouds into canopy and non-canopy points through morphological filtering techniques.

Multiple Return Analysis

At the beginning, our LiDAR datasets consist of one, two, three or four return points for each laser shots recorded in terms of x, y, and z coordinates and reflected intensity. Based on the observation and analysis in the LiDAR scanned data of a shopping plaza near the UF campus, all multiple-return shots occur on tree crowns, bushes, edges of buildings, etc, where multiple-return shots posses two, three or four return points in their individual shots and are represented by black dots in the Figure 4-2, in which the area size is about 180m by 130m. It is noted that there are no multiple-return shots occurring on any vehicles. These multiple-return shots are caused due to the fact that the beam area of a single laser shot covers more than one reflected spot which often happens on small leaves and sharp edges.
After analyzing the observation in Figure 4-2, two challenges which cause missing errors and false alarm errors are required to be conquered for removing tree canopy. The first challenge is that dense foliage will unexpectedly reflect single-return shots instead of normal multiple-return shots which can be observed on some tree tops in Figure 4-2 and lead to missing errors of tree canopy detection. The second one is that some sharp edges or protrusions of buildings and non-tree objects also give multiple-return shots similar to reflection from leaves of trees which will cause false alarm errors. These two problems can be solved by four steps: 1) analyze distance property between normal and dense foliage; 2) apply morphological filters including closing, opening, and dilating operators to process the binary canopy/non-canopy image; 3) create rough DTM to calculate above ground levels of points; 4) remove all canopy points above the maximum height of vehicles.

We also analyzed another city site “samp22” from ISPRS data sets, where its average point density is about 0.91 points per square meter and its image was snapshot from Google Earth [59] and is shown in Figure 4-3. This area was chosen because of its diverse feature content such as vegetation, buildings, roads, vehicles, bridges, etc. Its dataset only consists of the first-return and last-return points of laser shots. Although every shot of ISPRS was also recorded in terms of x, y, and z coordinates and reflected intensity, each shot always goes with two returns, first-return and last-return. Under this situation, one more procedure is required to categorize each shot into either the multiple-return class or single-return class. By calculating the difference between the z coordinates of the first-return point and last-return point of each shot, all shot-wise vertical distances are immediately obtained. By analyzing the histogram of shot-wise height differences in Figure 4-4 at the site samp22, it is found that multiple-return shots and single-return shots are in two separate groups which are either above or below 1 meter. If we zoom in on the smaller difference part and set all differences larger than 1 meter equal to 1 meter, shown as Figure 4-5, most single-return shots can be classified by a simple threshold of 10 cm. This threshold covers 99% of single-return shots, while the rest of them can be considered outlier situations. Additionally, we also map these height differences of shots to the site as Figure 4-6. It can be compared to Figure 4-3 and observed that 10 cm is a good threshold which helps to identify where the tree canopy regions are. Thus,
the laser shots can be put into categories of multiple-return shots and single-return shots simply by comparing vertical distances of individual shots to 10 cm. Therefore, if the height difference of one shot is below 10 cm, it is categorized as a single-return shot. Otherwise, it is considered as a multiple-return shot because its first-return and last-return laser points are reflected from two different spots in which we call potential tree spots and potential non-tree spots, respectively.

Taking this city site as an example, we first generate its DSM by resampling the irregular LiDAR point clouds to a regular grid image (Figure 4-7a). After projecting the x and y coordinates of first-return points of multiple-return shots represented by black dots to the corresponding DSM image, the fantastic characteristic of small-footprint LiDAR data can be observed again (Figure 4-7b). Those projected black dots in Figure 4-7b consist of tree tops, bushes, some edges of buildings and bridges, etc. The z coordinate of the first-return point is normally higher than the last-return point for any multiple-return shot. Hence, the first- and last-return points of multiple-return shots are classified as the potential tree spots and potential non-tree spots, respectively (Figure 4-1).

**Morphological Filtering**

Two further challenges need to be conquered after potential tree/non-tree spots are identified: 1) how to reject false potential tree spots as the non-canopy cell, and 2) how to accept potential non-tree spots as the canopy cell (Figure 4-1). The canopy/non-canopy cell here is defined as a 1m by 1m gridded DSM image and determined by whether the cell is occluded by trees or not. False potential tree spots include edges or protrusions of buildings, edges of bridges, and any objects lower than normal trees. Potential non-tree spots could also be the actual tree spots in the dense foliage case, because some height distances between leaves are probably less than the vertical accuracy of small-footprint LiDAR. These two challenges can be solved simultaneously by taking advantage of morphological filtering techniques. First, the binary image of tree spots is generated by labeling every potential tree and non-tree cell as 1 or 0, respectively (Figure 4-8a).

Then, we perform the morphological closing filter defined by [61]

\[ A \bullet B = (A \oplus B) \ominus B, \]

(4.1)
where \( A \oplus B \) and \( A \ominus B \) are the equation of morphological dilation and erosion, respectively. In this closing operation, the disk-shaped structuring element with a radius of 2 pixels is used to preserve the circular nature of the tree tops and recover missed tree spots occurred in the dense foliage case (Figure 4-8b). The closing filtering is followed by the morphological opening filter defined as [61]

\[
A \circ B = (A \oplus B) \ominus B,
\]

where the same structuring element is used to remove false potential tree spots in Figure 4-8b which are sparser than actual tree spots. This filtering effect can be observed from Figure 4-8c. Since the opening will decrease the size of detected tree spots, morphological dilating filtering of the same structuring element is performed to retrieve reduced edges of tree spots (Figure 4-8d). Because the dilation only increases the area where cell values are 1, the dilating filtering is able to prevent false alarm spots coming back since their cell values were changed from 1 to 0 by the morphological opening process.

**Tree Canopy Point Decision**

Based on previous morphological filtering procedures, the vegetation area is detected and showed in Figure 4-9. Comparing the multiple return points marked in Figure 4-7b to the result of detected tree spots or canopy cell marked in Figure 4-9, it shows that false potential tree spots are rejected and missed dense tree tops are recovered as well. However, the goal of this tree-canopy removal design is to classify LiDAR points into canopy and non-canopy points (Figure 4-1). The canopy points here are defined as LiDAR points which are from the top part of trees and possess the ability to occlude under-canopy objects. Oppositely, the non-canopy points are any other points which are not tree canopy points. All the LiDAR points inside our non-canopy cells are classified as non-canopy points, but points at canopy cells are not always canopy points since both vegetation and ground points probably exist at the same cell. Therefore, non-canopy points can exist in a canopy cell because some points inside the canopy cells could be reflected from the ground surface occluded by trees.

In order to recognize the canopy/non-canopy points inside the same canopy cell, the ground reference is generated by resampling the irregular LiDAR points inside those non-canopy cells to a
regular grid image. The above ground level (AGL) of each point then is easily obtained by calculating the height difference between its z coordinate and its corresponding ground reference. If the AGL of one point inside the canopy cell is greater than 0, it will be classified as a canopy point. Otherwise, it will be a non-canopy point (Figure 4-1).

**Tree Canopy Removal Result**

By removing those classified canopy points, the canopy removal result of the city test site (Figure 4-10) can be obtained from the remaining non-canopy points. When compared to its original DSM (Figure 4-7a), it is easy to see that most vegetation objects are successfully removed. This design also helps mitigate the vegetation impact in the bare-earth extraction algorithm.

On the other hand, three forest test sites are investigated to examine our canopy removal algorithm. First, The DSM of forest test sites (Hogtown parking site, Hogtown forest site, and Hogtown residential site) are generated by resampling their irregular LiDAR point clouds to regular grid images showed as Figure 4-11a, Figure 4-12a, and Figure 4-13a, respectively. After applying the canopy removal algorithm to these forest test sites, the new DSM without tree canopy can be obtained and are shown in Figure 4-11b, Figure 4-12b, and Figure 4-13b, respectively. By observing and comparing these forest test sites between their original DSM and canopy removal DSM, the difference is obvious that most forest canopy points have been successfully removed and all obscure vehicles or buildings underneath canopy can be easily seen.

In addition, since the canopy points have been classified from our algorithm, the occluded rate of forest canopy can be easily obtained by the following equation,

\[
Occluded Rate = \frac{\#Canopy Points}{\#Canopy Points + \#NonCanopy Points} \times 100\%. \quad (4.3)
\]

Accordingly, the occluded rates for the Hogtown parking site, Hogtown forest site, and Hogtown residential site are calculated and they are 63.51%, 80.59% and 48.31%, respectively. In fact, the original average LiDAR densities are 43.22, 50.13, and 36.72 points per square meter for the Hogtown parking site, Hogtown forest site, and Hogtown residential site, respectively. And, their remaining LiDAR
densities after removing canopy points are 15.77, 9.73, and 18.98, respectively. Therefore, the vehicles underneath canopy in the Hogtown forest site will be more difficult to detect than in the two other sites. Furthermore, the detailed distribution of remaining point density can be found as well since all x-y locations of non-canopy points are known. The density distributions for the Hogtown parking site, Hogtown forest site, and Hogtown residential site are shown in Figure 4-14a, Figure 4-14b, and Figure 4-14c, respectively. It is noted that this kind of distribution is very useful for predicting the performance of occluded target detection with respect to various object locations.

**Summary**

This canopy removal algorithm is demonstrated which helps 1) detect obscure targets underneath forest canopy and 2) mitigate the vegetation problem for those DTM extraction algorithms. As a matter of fact, the thermal imaging cameras can see the heat signature of people, boats, and vehicles in total darkness as well as through smoke, haze, and light fog, but not through the forest canopy. This proposed algorithm is learned from a city training site and verified by two LiDAR systems, another city test site, and three forest test sites. Whether in a city or a forest site, the vegetation area can be correctly detected and canopy points are successfully removed. All obscure vehicles or buildings underneath tree canopy are revealed as we demonstrated above. Accordingly, the occluded rate of forest canopy can be obtained. Furthermore, the detailed x-y distribution of remaining point density can be found as well which will be very useful for predicting the performance of occluded target detection with respect to various object locations.
Figure 4-1. Tree-canopy removal design flowchart

Figure 4-2. LiDAR multiple return shot illustration represented by black dots and overlapped on the training site.
Figure 4-3. Snapshot image of the site22 from Google Earth.

Figure 4-4. Histogram of shot-wise height differences at the site22
Figure 4-5. Zoom in the smaller difference part of Figure 4-4 and set the clipping range from 0 to 1.

Figure 4-6. Shot-wise height difference map clipped to 0.0m-0.1m at the site22.
Figure 4-7. Multiple return map at the samp22 city site: A) Original DSM. B) Multiple return points represented by marked circles.

Figure 4-8. Morphological filtering results at the samp22 city site: A) Binary image of Canopy/Non-Canopy cell represented by white/black color at the city test site. B) Image of applying morphological closing filter to Figure 4-8a. C) Image of applying morphological opening filter to Figure 4-8b. D) Image of applying morphological dilating filter to Figure 4-8c.
Figure 4-9. Detected vegetation area (canopy cell) marked on DSM of the city test site

Figure 4-10. Canopy removal result for the city test site

Figure 4-11. Revealed occluded vehicles at the Hogtown parking site: A) Original DSM B) Canopy removal DSM where vehicles are circled by white lines in which 7 occluded vehicles are revealed.
Figure 4-12. Revealed occluded vehicles at the Hogtown forest site: A) Original DSM B) Canopy Removal DSM where the occluded tarpaulin, mimic vehicle, circled by white lines is revealed.

Figure 4-13. Revealed occluded vehicles at the Hogtown residential site: A) Original DSM B) Canopy Removal DSM where vehicles are circled by white lines in which 4 occluded vehicles are revealed.

Figure 4-14. Remained point density in 2-D distribution after removing canopy points in A) Hogtown parking site, B) Hogtown forest site, and C) Hogtown residential site.
CHAPTER 5
POINT-BASED BARE-EARTH EXTRACTION DESIGN

In the last decade, a variety of algorithms have been developed for extracting the DTM from point clouds generated by LiDAR systems, where automatic and robust ground point extraction has been attracting great attention. Although most filters perform well in flat and uncomplicated landscapes, landscapes containing steep slopes and discontinuities are still problematic. We developed a novel bare-earth extraction algorithm and compared our performance to a dozen filters working on the same fifteen study sites which were provided by ISPRS. The result is that the average total error and kappa index of agreement of our algorithm in the automated process is 4.6% and 84.5%, respectively, which both outperform all other twelve proposed filters. Our kappa index, 84.5%, can be interpreted as almost perfect agreement.

Segmentation Modeling

The proposed bare-earth extraction design, consisting of segmentation modeling (Figure 5-1) and surface modeling, is the second design part of this entire work.

The input data, remaining non-canopy points, of our segmentation modeling are obtained by removing those canopy points from our tree-canopy removal design. This segmentation modeling is a triangulated irregular network (TIN) based design including triangle assimilation, edge clustering, and point classification and aims to achieve better discrimination of objects and preserve terrain discontinuities in the segmentation modeling. The triangle type, which is either flat or steep, is assimilated and reclassified by iteratively calling the majority vote from its neighboring triangles. In the surface modeling, the point is reclassified by using the roughness estimation of canopy removal DSM and DTM, bridge detection, and sharp ridge detection to further reduce both Type I and Type II errors. An extra advantage of this point-based algorithm is that we can avoid choosing any filtering window size during a pixel-based processing.

Triangle Classification and Assimilation

At the beginning, the Delaunay triangulation is used to build up the Triangulated Irregular Network
Two properties make the Delaunay triangulation attractive [62]. First, it is a local definition for triangulation. The insertion of a new point need only examine that region in the plane which is closer to the new point than any other point. Second, it is known that a given existing triangle is affected only if a new point is added within the circle circumscribed through its vertices. This prescribes a well-defined local search area for new points which can improve the fit of the terrain model to the region inside the triangle.

The TIN is a network of non-overlapping triangles generated from irregularly distributed LiDAR points with three dimensional coordinates (x, y, and z). Based on the TIN, each triangle with its three points can be classified either as a flat or steep triangle by calculating the height difference between its highest and lowest point and the slope degree, which is the degree difference between normal direction to the triangle plane and the direction to the sky parallel to the z axis (Figure 5-2a). If its height difference, $h$, is lower than 0.3 meter and slope degree, $\theta$, is smaller than 45°, this triangle will be classified as a flat triangle. Otherwise, it will be classified as a steep triangle. The threshold for height difference, $h$, should be as small as possible to ensure the quality of the flat triangles. Since the vertical accuracy of the LiDAR data is 0.3 meter, it is a good choice for the height difference threshold.

Although all triangles can be categorized into either flat or steep triangles, this simple decision rule of comparing the height difference and slope degree cannot give a satisfactory categorization. Hence, we add on the following novel triangle assimilation to get improved triangle classification. Every triangle has three edges which are shared with, at most, three other triangles. From those 3 triangles, we are able to give a better triangle classification, steep or flat, by taking a majority vote. For instance, there are three bold edges of the triangle A which are shared by triangles B, C, and D in Figure 5-2b. In this triangle group, if there are two or more of the adjacent triangles (B, C, and D) belonging to the flat triangle class, then triangle A will be classified as a flat triangle, where the vote of flat to steep could be 3-0 or 2-1. Likewise, if the number of adjacent triangles B, C, and D classified as steep triangles is more than flat triangles, then the triangle A will be assimilated to the steep triangle class, where the vote of flat to steep could be 0-3 or 1-2. During this assimilation process, the higher agreement vote 3-0 or 0-3 has to have...
the higher priority than the lower agreement vote 2-1 or 1-2. Therefore, those triangle groups giving higher agreement vote are searched first and used to assimilate their central triangle. Then, the other triangle groups having lower agreement votes are followed for the triangle assimilation.

The above triangle assimilation composed of two kinds of agreement votes has to be iterated until the number of triangles needed to be assimilated cannot be decreased any more. Figure 5-2c shows the fewest number configuration for a stable triangle cluster, which consists of uniformly classified, either flat or steep, and edge-connected triangles E, F, and G. Any clusters including less than three triangles or in unstable configurations continue to be reclassified during this iterated triangle assimilation. Hence, the triangle classification will become better and better through step by step iterations since the number of triangles which need to be assimilated becomes less and less (Figure 5-3).

The number of reclassified or assimilation-needed triangles can be regarded as the desirable noise level to be removed. Therefore, the performance of this triangle classification can be represented as the reduced noise in dB by the following formula,

\[
\text{ReducedNoise} = 10 \times \log_{10} \left( \frac{\text{NoiseBeforeProcess}}{\text{NoiseAfterProcess}} \right).
\]

For this case in the samp22 site, the reduced noise is equal to \( 10 \times \log_{10} \left( \frac{2572}{96} \right) = 14.3 \text{dB} \). This value shows how meaningful the iterated triangle assimilation is. The performance is contributed to the fact that not only are three immediate neighbor triangles considered to make a decision whether the center triangle is a steep or flat triangle, but the iteration steps also extend the assimilation process to the global area by the iterative procedure, which are like chained reactions. The triangle assimilation has the capability of performing a local search while preserving the merits of global treatment. Therefore, the final stable assimilation gives a triangle classification result which is from the majority vote among neighboring triangles locally and globally.

**Edge Classification and Clustering**

Every triangle is composed of 3 edges, each of which can be shared by, at most, two adjacent triangles. The combination cases of these two adjacent triangles can be both flat, both steep, or mixed
which is one flat and one steep triangle. For these three cases, each edge can be classified into three categories: flat, steep, and boundary, respectively. For example suppose that Figure 5-4 shows a small part of the TIN and is processed by the above triangle classification. There are three shaded and three white triangles which represent flat and steep triangles, respectively. Then, the dash lines, bold lines (A, B, and C), and regular lines are going to be classified as the flat edges, boundary edges, and steep edges, respectively, based on the above rule. Although the regular lines or steep edges are not shared with other adjacent steep triangles in Figure 5-4, they must be in reality. These triangles are not shown here in order to keep the figure concise. Otherwise, these three white, or steep, triangles should be assimilated to be flat triangles by the majority vote.

One object could include many edges of the TIN. The edge clustering then is needed to represent objects and three kinds of edge clusters are formulated: flat edge clusters, boundary edge clusters, and steep edge clusters. The edge clustering is simply used to cluster those edges which are connected together and have the same edge class. Figure 5-5 shows the edge classification result of the samp22 site where the flat and steep edges are represented by black and gray lines, respectively. Since the TIN here is a vector based representation of the physical surface built up by the nodes and lines from LiDAR point clouds, our edge classification is vector based. Figure 5-6 is the boundary edge classification result which is cluster-wise and indicates individual boundaries for all potential objects. The advantage of our boundary edge classification is that the spotty and disconnected edges, which are of most concern in edge detection methods of image processing, do not exist in our cluster-wise and vector-based edge detection (Figure 5-6).

Point Classification

While all points inside a steep edge cluster obtained from the above edge classification can be simply classified as object points, classifying points inside a flat edge cluster is a challenge since a flat edge cluster could be either a ground or roof surface. This challenge can be solved by calculating the ground ratio for each flat edge cluster:
\[
\text{GroundRatio} = \frac{\# \text{Boundary(FlatHeight < SteepHeight)}}{\# \text{Boundary(FlatHeight > SteepHeight)}},
\]

where the denominator is the number of boundary edges where flat triangles are higher than steep triangles, both of which have to share the same boundary edges, and the numerator is just to calculate the other lower part. Taking Figure 5-4 as an example again, a flat edge cluster is formed by three dashed lines which include the point P0 and three white points. Its boundary edges are three bold lines A, B, and C. The height comparison between a flat and steep triangle sharing the same boundary edge A, B, or C is equivalent to calculating the height difference between one of the black points P1, P2, or P3 and the point P0. Suppose that the point P0 is lower than other points P1, P2, and P3. Then, the denominator in Equation (5.2) is equal to 0 and the numerator is 3. The ground ratio of this flat edge cluster is equal to 3/0, or infinity. Conversely, if the point P0 is higher than points P1, P2, and P3, then this cluster-wise ground ratio is equal to 0/3, or 0.

The ground ratio is a good indicator for how confidently we classify a flat edge cluster to a ground surface. In general, when the ground ratio is greater than 1, this flat edge cluster has higher probability to be ground than roof. However, when many boundary edges of the roof are connected to other higher parts of buildings, this ground ratio could be greater than 1. Then, the threshold for ground ratio should be considered as strict as possible to ensure the quality of extracted ground points. Eventually, the threshold for ground ratio is set at 8 by our experimental tests, where the error of accepting roof points as ground points is minimized. Therefore, all points inside a flat cluster would be classified as ground points if its ground ratio is higher than our threshold. Otherwise, those points would be classified as object points. In addition, all points inside steep clusters are classified as object points simply because those points do not belong to flat clusters.

**Surface Modeling**

The above segmentation modeling is to ensure the quality of ground points such that those extracted ground points can safely be considered as actual ground points and act as seed points to progressively extend the ground regions. Our extracted ground/object points can be compared to
reference ground/object points by the cross matrix in Table 5-1, where the False Negative (FN) case, or Type I error, is for falsely rejecting ground points, and the False Positive (FP) case, or Type II error, is for incorrectly accepting object points. The remaining True Positive (TP) and True Negative (TN) cases are for the correct classifications of ground and object points, respectively.

Figure 5-7 shows the result after applying the method in the previous section to the samp22 site, where the Type I error is represented by black x-marks and the Type II error is represented by white x-marks. It is noted that the Type I error is very high and Type II error is very low in Figure 5-7. That is by design of our bare-earth extraction technique because the ratio of TP to FP will be high due to low Type II error so that all extracted ground points should be actual ground points. In order to reduce both Type I and Type II errors, the four-phase point reclassification based on the surface modeling is developed and its flowchart is shown as Figure 5-8 where each phase is iterated.

**Phase-I Point Classification Based on DTM**

In the phase I reclassification, the terrain raster map is generated by resampling the extracted ground points to a regular grid image and used as the ground level reference, where the grid cell size is 2 meters by 2 meters for all 15 sites. Due to the good quality of extracted ground points in which Type II error is very low, it is possible to recover some missed ground points by comparing their height levels to the terrain raster map. To do so, the AGL of each point is calculated by subtracting its corresponding elevation on the terrain raster map from its z coordinate. If the AGL of one ground point is higher than the height threshold defined as the variable TH1, it is reclassified as an object point. On the other hand, if the AGL of one object point is lower than the TH1, it is reclassified as a ground point.

The choice of TH1 will influence the result of this phase reclassification. With our experiments, setting the TH1 as 1.6 meters is a good choice for all 15 sites. However, it can be adjusted manually to obtain optimal results for each site. In each iteration, those updated ground points will be used to update the terrain raster map as the ground level reference for the next iteration. The total number of corrected ground and object points will be decreased step by step with each iteration. The iterative process ends when the total corrected number cannot be decreased any more. The result of the phase I reclassification
Phase-II Point Classification Based on DSM

Although the Type I error is reduced in the phase I reclassification, the Type II error is increased when comparing Figure 5-7 to Figure 5-9. Thus, this phase II reclassification is designed to reduce the Type II error resulting in the phase I reclassification. First, the roughness raster map is generated from the previous non-canopy points because tree canopy points will impact the estimation of surface roughness. The chosen pixel size and window size is 0.8 meters and 4 meters, respectively, for each side. In each running window, its roughness value here is defined as the elevation difference between the highest and lowest pixel among 25 pixels (5 pixel by 5 pixel square) located in this window. After the roughness map is generated (Figure 5-10), all ground or object points can find their own corresponding roughness values by their x and y coordinates. The raster DTM has to be generated by resampling the current ground points for the ground level reference of all points, where the grid cell size is 0.8 meters by 0.8 meters for all 15 sites, and then the AGL values of all points can be calculated by their z coordinates and the obtained DTM. Based on obtained roughness and AGL values, each point can be reclassified by the following rule. If one object point has a roughness value less than 0.3 meters and its AGL is lower than 0.3 meters, then this point will be reclassified as a ground point. Since a flat local area, 5 pixels by 5 pixels, normally consists of either all ground points or all object points, a point inside the area and close to the DTM should be a ground point rather than an object point. Similarly, if one ground point has a roughness value greater than 0.3 meters and its AGL is higher than 0.3 meters, then this point will be reclassified as an object point. Since a rough local area usually consists of both object and ground points, a point inside the area and above the DTM should have a higher probability of being an object point than a ground point.

In each iteration of this phase, updated ground points will be used to update the DTM in the next iteration. The roughness map does not have to be updated since it is generated from all points. The total number of corrected ground and object points will also be decreased step by step with each iteration. The
iterative process ends when the total corrected number cannot be decreased any more. Figure 5-11 shows the result of phase-II point reclassification in the samp22 site. By observing Figure 5-9 and Figure 5-11, it is obvious that the Type II error is dramatically reduced by this phase reclassification. It is the roughness map that helps to reduce the Type II error because most false ground points which are corrected to object points are located at high roughness areas, which can be found by mapping those white x-marks of Figure 5-9 to the corresponding roughness map of Figure 5-10.

**Phase-III Point Classification Based on DTM Roughness**

The sharp ridges on the ground surface are usually a major problem for ground filtering [15]. This problem can be mitigated by this phase-III point reclassification. First, a roughness map for DTM (Figure 5-12) is generated by resampling those ground points from phase-II point reclassification, where the chosen pixel size and window size are the same as before. Second, we propose a simple and effective bridge detection for the later correction.

Our bridge detection can be separated into two parts: bridge body and bridge ends. For the bridge body, the morphological top-hat filtering with a disk-shaped structuring element and a radius of 12 pixels is performed on the canopy removal DSM, where the top-hat is useful for enhancing detail in the presence of local tops [61]. Then, this top-hat transformation is converted to a binary image where pixel levels larger than 4 are changed to 1 and other pixels are changed to 0. This is followed by the opening transformation with a disk-shaped structuring element and a radius of 2 pixels to remove small isolated noise. For bridge ends detection, the roughness of DTM (Figure 5-12) is useful since the bridge ends always occur at sharp ridges on the ground surface. Then, the other binary image for possible bridge ends is generated where those roughness values of pixels greater than 0.7 meters are changed to 1 and other pixels are changed to 0. By overlapping these two binary images, any one bridge body which is connected to two bridge ends is our final detected bridge. The LiDAR points of detected bridges are represented by black x-marks in Figure 5-13 which show no obvious missing or false alarm errors in our bridge detection.

Based on the roughness map of DTM, it is easy to know where the regions of sharp ridges on the
ground surface are. For those problematic areas, each rough area of DTM can be corrected by its nearest flat part in which the height level of the flat part will extend to the rough part. This approach can shrink the width of rough areas since those rough parts have been updated to the same ground levels of their nearest neighboring flat parts of the DTM. Therefore, those object points in rough regions of the DTM are reclassified as ground points if their AGL from the updated DTM are lower than 0.3 meters. It is noted that the detected bridges have potential to be reclassified as ground points, so all bridge points are detected first and are excluded in this rule. In addition, those ground points in flat regions of the DTM are reclassified as object points if their AGL from the updated DTM are higher than 0.3 meters.

In each iteration of this phase, the DTM is updated to make the width of rough areas become narrower and narrower. The iteration ends as before when the total number of corrected ground and object points cannot be smaller than previous iterations. The result (Figure 5-14) shows that the number of Type I errors occurring in the sharp ridges on the ground surface is indeed decreased significantly which can be easily seen by checking the road sides near the bridge in the left bottom parts of Figure 5-11 and Figure 5-14.

**Phase-IV Point Classification Based on Flattened DSM**

It is easy for the final point reclassification to achieve good performance when most of the sharp ridge points are able to be extracted as ground points. First, the DTM is generated by resampling those ground points from phase-III point reclassification, where the grid cell size is 0.7 meters by 0.7 meters for all 15 sites. Then, the AGL values of all points can be obtained by referring to the generated DTM in this phase. The flattened DSM (Figure 5-15) is generated by resampling all points with AGL values instead of their original z coordinates, where it is clear to see that most of the rough areas including sharp ridges on the ground surface are flattened, especially for those road sides near the bridge in the bottom left part of Figure 5-15.

The ground/object points can simply be reclassified from the flattened DSM by their AGL values obtained in this phase. If object points are lower than the threshold TH2, those points are reclassified as ground points. Conversely, if ground points are higher than the threshold TH2, those points are
reclassified as object points. This also needs iterating to make the flattened DSM become flatter and flatter in those rough terrain areas. The iterated process ends, as before, when the total number of corrected ground and object points of the current iteration is not reduced from the previous iteration. The TH2 here is applied as 0.6 meters for all 15 sites. It can be manually adjusted to reach a better performance. The Type I and Type II errors of the final classification of ground/object points are represented by black and white x-marks, respectively, in which the TH1 and TH2 are applied by 1.6 meters and 0.6 meters, respectively (Figure 5-16).

**Bare-Earth Extraction Result**

The following two sections show the automatic and manual performances of our proposed bare-Earth extraction design, respectively. In order to evaluate our algorithm, the open data sets from ISPRS were used in order to compare our results to other algorithms using the same data sets.

**Same Parameters for All 15 Sites of ISPRS**

Through processes of our tree-canopy removal, bare-earth extraction, and point reclassifications, the final point classification results for all 15 study sites are shown, where the Type I and Type II errors are represented by black and white x-marks, respectively, and marked on their corresponding canopy removal DSM images. In order to show the geographic relation between the study sites and their original sites, the corresponding areas are also marked by their boundaries (Figure 5-17 ~ Figure 5-38). It is noted that both applied parameters TH1 and TH2 are fixed and the same for all 15 sites, which are 1.6 meters and 0.6 meters, respectively.

The accuracy of the classification map is assessed with the use of the error matrix from which the total error and the Kappa index of agreement are derived [64]. The total error is calculated by adding up the Type I and Type II errors and dividing this sum by the total number of points, so this is the overall probability of a reference pixel being incorrectly classified. The kappa index of agreement is an alternative measure of the overall classification accuracy that subtracts the effect of chance agreement and quantifies how much better a particular classification is, as compared to a random classification [65]. The equation for the kappa index of agreement [66], \( \kappa \), is
\[ \kappa = \frac{\Pr(a) - \Pr(e)}{1 - \Pr(e)}, \quad (5.3) \]

where \( \Pr(a) \) is the relative observed agreement among raters, and \( \Pr(e) \) is the hypothetical probability of chance agreement, using the observed data to calculate the probabilities of each observer randomly deciding each category. Accordingly, the \( \Pr(a) \) and \( \Pr(e) \) can be derived to the following equations

\[ \Pr(a) = \frac{TP + TN}{TP + FN + FP + TN}, \quad (5.4) \]

\[ \Pr(e) = \frac{(TP + FN)(TP + FP) + (FP + TN)(FN + TN)}{(TP + FN + FP + TN)^2}, \quad (5.5) \]

where the TP, FN, FP, and TN are defined as in Table 5-1. The significance of the kappa index of agreement is interpreted by Table 5-2 [47].

The accuracy of our point classification for every study site is assessed by the percentages of total errors and the kappa index of agreement (Table 5-3). The dense objects on steep terrain in a city site will cause more false positive, or Type II, errors (accept object points) which occur in the site of samp11. The rough and narrow terrain with sharp ridges will cause more false negative, or Type I, errors (reject ground points) which happens in the sites of samp53 and samp61.

Other difficult terrain types, however, do not impact our performance. For example, the complex scene in the site of samp23 is a plaza surrounded on three sides by a block of buildings. There is a sunken arcade in the center of the plaza. Both the plaza and arcade should be assumed to be bare-Earth. For this scene, the filters that make use of local surface assumptions like in Axelsson, Preifer, and Shon performed best [42]. In our filter, the ground ratio for judging each flat edge cluster in the TIN is also a rule based on the local surface in which the height difference of its all connected clusters will be considered rather than comparing its height to the local lowest place. Thus, this complex scene is not a problem for our filter.

Bridges can also be a difficult terrain type. The Type II error (classify object points as ground points) usually occurs near the beginning and end of bridges while bridges in the test should be treated as objects
In our filter, the bridge is detected by searching for and connecting its bridge body and both beginning and ends. The detected bridges including both ends are treated as object points instead of ground points such that this kind of Type II error will be reduced.

Similar to bridges are ramps. Ramps bear similarity to bridges in that they span gaps in the bare-Earth. However, they differ in that they do not allow movement below them. As such, ramps were treated as bare-Earth in the reference data. All the tested algorithms filtered off the ramps [42] and caused Type I errors. In our filter, most points on ramps are initially classified as object points. By iterating point reclassification, those points initially classified as object points but near to bare-Earth can be corrected and reclassified as ground points. Thus, even though only partial and lower points on ramps are classified as ground points, they will grow upward and extend gradually to cover most of the ramps through iterated correction. Accordingly, those ramps are treated as the bare-Earth in our filter. It can be checked in our point classification result for the site of samp71, where ramps are used to connect the road across the bridge.

Vegetation on slopes can be another difficult terrain type [42], but this is not an issue for our filter. At the beginning, most vegetation points are already removed in our canopy removal design where the fact that vegetation objects usually reflect different height levels for the first and last return of each laser shot is used. Besides, it is noted that vegetation will impact the roughness estimation for bare-Earth. As such the canopy removal design becomes very meaningful in vegetated terrain.

For the accuracy assessment, our filter in the same parameters case or automated process (called as Chang#1 here) is compared to the best two filters in the report [68]. The total errors and kappa index of agreement for each site are represented by empty and filled legends, respectively (Figure 5-39). This result shows that Chang#1’s filter often gives better performance than Pfeifer’s filter and is little better than Axelsson’s filter. The average total errors and kappa index of agreement for 15 sites are compared and given in the next subsection.

**Optimized Parameters for All 15 Sites of ISPRS**

By testing different associated parameters of our filter in all study sites and comparing their
influences, it is found that the TH1 and TH2 values will dominate the final performance. Accordingly, our filter is optimized by adjusting these two parameters to substitute the initial values of TH1 (1.6m) and TH2 (0.6m). The accuracy of our point classification for every study site with two optimized parameters, TH1 and TH2, is listed in Table 5-4, which also includes corresponding FN and FP. It is found that the smaller the TH1 is, the better the ability to detect the smooth terrain will be. Conversely, the larger the TH1 is, the better the ability to mitigate the sharp ridges will be. On the other hand, the TH2 is applied to try to correct those errors from the TH1. Hence, the more the error from the TH1 is, the larger the TH2 will be. This phenomenon can be observed in the TH2 column of Table 5-4 for the sites samp11, samp52, samp53, and samp61.

The automated (Chang#1) and optimized (Chang#2) result is also compared in Figure 5-40. It shows that Chang#2’s filter always performs better than Chang#1’s filter. Figure 5-41 shows the average performance of all study sites for those filters in [68] and both of our automated and optimized filters, where the average total errors and average kappa index of agreement are calculated for each filter. It is clear that our automated filter outperforms all other filters in [68] and the performance is even better after two dominant parameters are optimized manually.

Recently, Sithole [26], Silvan-Cardenas [27], Lu [28], and Meng [29] proposed new filtering algorithms and used the same data sets to assess their performance. Some authors presented their filtering results by the kappa index of agreement only, so we have to calculate other authors’ TP, FN, FP, and TN values defined in Table 5-1 of each site to obtain their average kappa value for 15 study sites for comparing the performance difference among filters. The average accuracy comparison of 15 study sites for our automated and optimized algorithms to four different filters proposed recently (after year 2004) is represented by the kappa index of agreement and shown in Figure 5-42. It is noted that these recent filtering performances are even worse than the filter developed by Axelsson in 2000. However, both of our automated and optimized filters are better than all these twelve filters, in which eight were proposed before 2004 and four were proposed after 2004.
**Summary**

Automatic and robust ground filtering is important in LiDAR applications where classified ground and object points can be used for DTM generation and further reconstruction of topographic features. Many methods have been proposed to extract points on bare-Earth from LiDAR data. However, most filters perform well in flat and uncomplicated landscapes, while the landscapes containing steep slopes and discontinuities are still a problem which has not been fully solved.

We present an algorithm which is composed of segmentation modeling and surface modeling. The special features inside our designs include the TIN based platform, iterated triangle assimilation, edge clustering, roughness estimation of canopy removal DSM and DTM, bridge detection, and sharp ridge detection, etc. These designs aim to mitigate vegetation interference, obtain better discrimination of objects, preserve terrain discontinuities, and reduce both Type I and Type II errors.

The performance of our algorithm is compared to twelve proposed filters and evaluated by working on the same fifteen study sites. The average total errors and kappa index of agreement of this work in the automated process is 4.6% and 84.5%, respectively, which outperforms all twelve other filters and such kappa index is interpreted as almost perfect agreement. In addition, this work applied with optimized parameters performs even better.
Table 5-1. Extracted and reference cross matrix

<table>
<thead>
<tr>
<th>Points</th>
<th>Extracted Ground</th>
<th>Extracted Object</th>
</tr>
</thead>
<tbody>
<tr>
<td>Reference ground</td>
<td>True positive (TP)</td>
<td>False negative (FN)/Type I error</td>
</tr>
<tr>
<td>Reference object</td>
<td>False positive (FP)/Type II error</td>
<td>True negative (TN)</td>
</tr>
</tbody>
</table>

Table 5-2. Interpretation of the kappa index of agreement

<table>
<thead>
<tr>
<th>Kappa index</th>
<th>Interpretation</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt; 0%</td>
<td>No agreement</td>
</tr>
<tr>
<td>0% ~ 20%</td>
<td>Slight agreement</td>
</tr>
<tr>
<td>21% ~ 40%</td>
<td>Fair agreement</td>
</tr>
<tr>
<td>41% ~ 60%</td>
<td>Moderate agreement</td>
</tr>
<tr>
<td>61% ~ 80%</td>
<td>Substantial agreement</td>
</tr>
<tr>
<td>81% ~ 100%</td>
<td>Almost perfect agreement</td>
</tr>
</tbody>
</table>

Table 5-3. Total errors and kappa index for 15 study sites with same parameters (TH1=1.6m and TH2=0.6m) in our filter.

<table>
<thead>
<tr>
<th>Site</th>
<th>TP</th>
<th>TN</th>
<th>FN</th>
<th>FP</th>
<th>Error</th>
<th>Kappa</th>
</tr>
</thead>
<tbody>
<tr>
<td>Samp11</td>
<td>20421</td>
<td>11943</td>
<td>1365</td>
<td>4281</td>
<td>14.9%</td>
<td>68.9%</td>
</tr>
<tr>
<td>Samp12</td>
<td>26292</td>
<td>23534</td>
<td>399</td>
<td>1894</td>
<td>4.4%</td>
<td>91.2%</td>
</tr>
<tr>
<td>Samp21</td>
<td>10081</td>
<td>2675</td>
<td>4</td>
<td>200</td>
<td>1.6%</td>
<td>95.3%</td>
</tr>
<tr>
<td>Samp22</td>
<td>22299</td>
<td>9159</td>
<td>205</td>
<td>1043</td>
<td>3.8%</td>
<td>90.9%</td>
</tr>
<tr>
<td>Samp23</td>
<td>12919</td>
<td>11169</td>
<td>304</td>
<td>703</td>
<td>4.0%</td>
<td>91.9%</td>
</tr>
<tr>
<td>Samp24</td>
<td>5161</td>
<td>1846</td>
<td>273</td>
<td>212</td>
<td>6.5%</td>
<td>83.9%</td>
</tr>
<tr>
<td>Samp31</td>
<td>15522</td>
<td>12393</td>
<td>34</td>
<td>913</td>
<td>3.3%</td>
<td>93.4%</td>
</tr>
<tr>
<td>Samp41</td>
<td>5408</td>
<td>4981</td>
<td>194</td>
<td>648</td>
<td>7.5%</td>
<td>85.0%</td>
</tr>
<tr>
<td>Samp42</td>
<td>12377</td>
<td>29199</td>
<td>66</td>
<td>828</td>
<td>2.1%</td>
<td>95.0%</td>
</tr>
<tr>
<td>Samp51</td>
<td>13915</td>
<td>3250</td>
<td>35</td>
<td>645</td>
<td>3.8%</td>
<td>88.2%</td>
</tr>
<tr>
<td>Samp52</td>
<td>19679</td>
<td>1857</td>
<td>433</td>
<td>505</td>
<td>4.2%</td>
<td>77.5%</td>
</tr>
<tr>
<td>Samp53</td>
<td>31156</td>
<td>1217</td>
<td>1833</td>
<td>172</td>
<td>5.8%</td>
<td>52.2%</td>
</tr>
<tr>
<td>Samp54</td>
<td>3890</td>
<td>4441</td>
<td>93</td>
<td>184</td>
<td>3.2%</td>
<td>93.5%</td>
</tr>
<tr>
<td>Samp61</td>
<td>33354</td>
<td>991</td>
<td>500</td>
<td>215</td>
<td>2.0%</td>
<td>72.4%</td>
</tr>
<tr>
<td>Samp71</td>
<td>13781</td>
<td>1495</td>
<td>94</td>
<td>275</td>
<td>2.4%</td>
<td>87.7%</td>
</tr>
</tbody>
</table>

Table 5-4. Total errors and kappa index of agreement for 15 study sites in our filter with two optimized parameters, TH1 and TH2.

<table>
<thead>
<tr>
<th>Site</th>
<th>TP</th>
<th>TN</th>
<th>FN</th>
<th>FP</th>
<th>Error</th>
<th>Kappa</th>
</tr>
</thead>
<tbody>
<tr>
<td>Samp11</td>
<td>0.5 m</td>
<td>0.8 m</td>
<td>1158</td>
<td>3903</td>
<td>13.3%</td>
<td>72.2%</td>
</tr>
<tr>
<td>Samp12</td>
<td>1.0 m</td>
<td>0.5 m</td>
<td>571</td>
<td>1262</td>
<td>3.5%</td>
<td>93.0%</td>
</tr>
<tr>
<td>Samp21</td>
<td>0.5 m</td>
<td>0.5 m</td>
<td>24</td>
<td>135</td>
<td>1.2%</td>
<td>96.4%</td>
</tr>
<tr>
<td>Samp22</td>
<td>1.1 m</td>
<td>0.5 m</td>
<td>269</td>
<td>574</td>
<td>2.6%</td>
<td>94.0%</td>
</tr>
<tr>
<td>Samp23</td>
<td>1.5 m</td>
<td>0.6 m</td>
<td>298</td>
<td>707</td>
<td>4%</td>
<td>92.0%</td>
</tr>
<tr>
<td>Samp24</td>
<td>0.6 m</td>
<td>0.5 m</td>
<td>279</td>
<td>197</td>
<td>6.4%</td>
<td>84.3%</td>
</tr>
<tr>
<td>Samp31</td>
<td>0.5 m</td>
<td>0.4 m</td>
<td>99</td>
<td>256</td>
<td>1.2%</td>
<td>97.5%</td>
</tr>
<tr>
<td>Samp41</td>
<td>2.4 m</td>
<td>0.5 m</td>
<td>185</td>
<td>610</td>
<td>7.1%</td>
<td>85.9%</td>
</tr>
<tr>
<td>Samp42</td>
<td>0.9 m</td>
<td>0.5 m</td>
<td>108</td>
<td>463</td>
<td>1.3%</td>
<td>96.8%</td>
</tr>
<tr>
<td>Samp51</td>
<td>0.3 m</td>
<td>0.5 m</td>
<td>77</td>
<td>480</td>
<td>3.1%</td>
<td>90.5%</td>
</tr>
<tr>
<td>Samp52</td>
<td>2.2 m</td>
<td>0.9 m</td>
<td>151</td>
<td>634</td>
<td>3.5%</td>
<td>79.6%</td>
</tr>
<tr>
<td>Samp53</td>
<td>2.6 m</td>
<td>1.1 m</td>
<td>647</td>
<td>465</td>
<td>3.2%</td>
<td>60.8%</td>
</tr>
<tr>
<td>Samp54</td>
<td>1.2 m</td>
<td>0.7 m</td>
<td>80</td>
<td>162</td>
<td>2.8%</td>
<td>94.4%</td>
</tr>
<tr>
<td>Samp61</td>
<td>2.4 m</td>
<td>0.8 m</td>
<td>153</td>
<td>234</td>
<td>1.1%</td>
<td>82.8%</td>
</tr>
<tr>
<td>Samp71</td>
<td>1.0 m</td>
<td>0.7 m</td>
<td>79</td>
<td>244</td>
<td>2.1%</td>
<td>89.3%</td>
</tr>
</tbody>
</table>
Figure 5-1. Segmentation modeling flowchart of our bare-earth extraction design

Figure 5-2. Triangle classification and assimilation illustration: A) The height difference $h$ and slope degree $\theta$ of a triangle. B) Triangle assimilation illustration. C) The fewest number configuration for a stable triangle cluster.

Figure 5-3. Triangle assimilation curve of samp22 site
Figure 5-4. Cluster-wise ground ratio illustration.

Figure 5-5. Vector-based edge classification where black and gray lines represent flat and steep edges, respectively.

Figure 5-6. Cluster-wise boundary edge detection
Figure 5-7. Initial point classification result where Type I (black x-mark) and Type II (white x-mark) errors are marked on canopy removal DSM image.

Figure 5-8. Surface modeling flowchart of our bare-earth extraction design.
Figure 5-9. Phase-I point reclassification result: Type I (black x-mark) and Type II (white x-mark) errors marked on canopy removal DSM image.

Figure 5-10. Roughness map of canopy removal DSM

Figure 5-11. Phase-II Point reclassification result: Type I (black x-mark) and Type II (white x-mark) errors marked on canopy removal DSM image.
Figure 5-12. Roughness map of DTM

Figure 5-13. Bridge detection result showed by marking detected bridges as black x-marks on the canopy removal DSM.

Figure 5-14. Point reclassification phase-III result: Type I (black x-mark) and Type II (white x-mark) errors marked on canopy removal DSM image.
Figure 5-15. The flattened DSM map obtained by resampling all points where their z coordinates are replaced by the above ground levels referred to those ground points extracted from the phase-III point reclassification.

Figure 5-16. Final point classification result: Type I (black x-mark) and Type II (white x-mark) errors marked on canopy removal DSM image.

Figure 5-17. DSM image of the city site #1, where the marked rectangular a and b is the boundary of the study site samp11 and samp12, respectively.
Figure 5-18. The final classification errors in the site of samp11.

Figure 5-19. The final classification errors in the site of samp12.

Figure 5-20. DSM image of the city site #2, where the marked rectangular a, b, c, and d is the boundary of the study site samp21, samp22, samp23, and samp24, respectively.
Figure 5-21. The final classification errors in the site of samp21.

Figure 5-22. The final classification errors in the site of samp22.

Figure 5-23. The final classification errors in the site of samp23.
Figure 5-24. The final classification errors in the site of samp24.

Figure 5-25. DSM image of the city site #3, where the marked rectangle is the boundary of the study site samp31.

Figure 5-26. The final classification errors in the site of samp31.
Figure 5-27. DSM image of the city site #4, where the marked rectangles a and b are the boundaries of the study sites samp41, and samp42, respectively.

Figure 5-28. The final classification errors in the site of samp41.

Figure 5-29. The final classification errors in the site of samp42.
Figure 5-30. DSM image of the forest site #5, where the marked rectangles a, b, c and d are the boundaries of the study sites samp51, samp52, samp53, and samp54, respectively.

Figure 5-31. The final classification errors in the site of samp51.

Figure 5-32. The final classification errors in the site of samp52.
Figure 5-33. The final classification errors in the site of samp53.

Figure 5-34. The final classification errors in the site of samp54.

Figure 5-35. DSM image of the forest site #6, where the marked rectangle is the boundary of the study site samp61.
Figure 5-36. The final classification errors in the site of samp61

Figure 5-37. DSM image of the forest site #7, where the marked rectangle is the boundary of the study site samp71.

Figure 5-38. The final classification errors in the site of samp71
Figure 5-39. Accuracy comparison of our automated filter (Chang#1) to the best two filters (Pfeifer and Axelsson) in [68] for all 15 study sites.

Figure 5-40. Accuracy comparison of our automated filter (Chang#1) to our optimized filter (Chang#2) for all 15 study sites.
Figure 5-41. Average accuracy comparison of 15 study sites to eight different filters proposed before year 2004 represented by the total error and kappa index of agreement.

Figure 5-42. Average accuracy comparison of 15 study sites to four different filters proposed after year 2004 represented by the kappa index of agreement.
CHAPTER 6
GRID-BASED BARE-EARTH EXTRACTION DESIGN

Object Detection

Outlier and Tree Canopy Removal

Local LiDAR point outliers are often randomly distributed over a study area which may be caused by passing birds, airplanes, or the sensor itself. Normally they are extremely higher or lower than adjacent points and isolated from other points which must be removed during preprocessing [29,69-72]. The simplest way to identify these outliers is to examine the frequency distribution of elevation values [29,72,73]. Manual examination of the dataset is another viable option [72]. In this study, only the last return points of LiDAR data need to be considered since none of the bare-Earth points come from the other return points. In the last return data set, a 5m×5m window $w_i$ is used to get neighbored LiDAR points for each point $p_i$ where its z coordinate is $z_i$. The point $p_i$ is decided as an outlier if

$$|z_i - \mu_i| > 2\sigma_i$$

(6.1)

where the mean ($\mu_i$) and standard deviation ($\sigma_i$) are obtained from those points inside the window $w_i$. It is assumed that the height values belong to a normal distribution in which values less than one standard deviation from the mean (dark blue) account for about 68% of the set, while two standard deviations from the mean (medium and dark blue) account for about 95%, and three standard deviations (light, medium, and dark blue) account for about 99.7% (Figure 6-1). So, Equation (6.1) results in the 95% confidence intervals.

As an example, in site22 there are 69 outlier points found by Equation (6.1) in all 32,707 last return points which occupy 0.21% of total points shown as Figure 6-2, where 20 points are lower and 49 points are higher than two standard deviations. After we get outlier-free data sets, the developed tree canopy removal algorithm in Chapter 4 is used again to get canopy-free sites, which are ready for the following segmentation and classification in this object detection section.
Segmentation by Non-Flat Regions

In LiDAR data, the ground points are the measurements from bare-Earth terrain that are usually the lowest surface features in a local area. Non-ground points are the measurements from the objects above the bare-Earth terrain, such as trees, buildings, bridges, and shrubs. Surface slope is generally lower between two neighboring bare ground points than between one bare ground and one non-ground point [74]. Those non-ground points between objects and bare-Earth points usually generate the non-flat regions which can be easily observed by watching the slope map of a DSM. Thus, the slope feature is used in this study and the 3rd order finite difference (3FD) algorithm is selected and defined by

\[
S = \arctan \sqrt{f_x^2 + f_y^2}
\]  

(6.2)

\[
f_x = \left( z_3 - z_1 + z_6 - z_4 + z_9 - z_7 \right) / 6g
\]

(6.3)

\[
f_y = \left( z_7 - z_1 + z_8 - z_2 + z_9 - z_3 \right) / 6g
\]

(6.4)

where \( g \) is spatial resolution (i.e., grid cell size) set as 1m and 0.5m for ISPRS and UF ALSM study sites, respectively, and the associated positions of \( z \) coordinates are assigned by Table 6-1.

The 3FD derived slope method was recommended by [75] since it is less sensitive to the LiDAR data error which makes it more appropriate for applications. In addition, a low-pass 3×3 filter (Table 6-2) is applied to the derived slope map to reduce further noise and its kernel weights are defined as 1/16 of values shown in Table 6-2. With smoothed slope information, the surface steepness is easily detected by morphological tophat filtering which returns the image minus the morphological opening of the image (erosion followed by dilation). Based on our experiments, the disk structure element with radius 2 is a good choice for the tophat transform. Similarly, a smoothed tophat transform is obtained with the smoothing low-pass filtering. A pixel is decided to be flat if its smoothed slope is less than 45° and smoothed tophat value less than 8. Then, a flat region is built up simply by clustering those connected flat pixels. Figure 6-3 shows as an example for the smoothed slope map and the smoothed tophat filtering at the site22. Figure 6-4 shows its flat region decided by our rule.
**Classification of flat regions**

A flat region could be a ground or an object region such as a building roof, car roof, etc. In order to classify these flat regions into either a ground or non-ground class, we extracted four features and put them into the rule-based methods to do this classification task. These features are: 1) the number of inside clusters, 2) height indicator, 3) area size, and 4) slope indicator of a tested flat region. Feature #1, the number of inside clusters, is defined as the number of clusters which are completely enclosed in the tested flat region. Feature #2, height indicator, represents an indicator of a height difference on the boundary of a tested flat region. Concerning height values of pixels locating in a $5\times5$ square window centered on each boundary pixel, the height difference is calculated by subtracting its average outside pixel heights from inside pixel heights. The height indicator is the mean value of the positive part of height difference, where the negative part is ignored since we only want to know how high the tested flat region is when it is compared to ground surface. Feature #3, area size, is simply the amount of the occupied area of a tested flat region. Feature #4, slope indicator, is obtained similarly to feature #2. Concerning those slope values of pixels which are located in a $5\times5$ square window centered on the each boundary pixel, the mean slope of outside the tested flat region is calculated. The positive/negative sign of slopes is assigned along the boundary by the sign of height differences obtained during the estimation of feature #2. The slope indicator is the mean value of signed slopes. Based on the above features, a tested flat region or a cluster is classified as a ground cluster or an object cluster by the rule-based methods showed in Figure 6-5. The feature maps of each cluster and the classification result at the site22 are shown for an example as Figure 6-6. The obtained DTM is shown in Figure 6-7a to compare with the reference DTM in Figure 6-7b created from the ground truth. In addition, the initial DTM and reference DTM for all of the study sites provided by ISPRS are shown in Figure 6-8 through Figure 6-21. Any missing objects will easily be noticed by comparing our initial DTM to the reference DTM since they are higher than the ground surface. By observing these figures, it is obvious that almost all objects are detected and removed in this stage. This grid-based object detection algorithm is developed here to be a pre-filter for the further processing of the statistical bare-Earth extraction algorithm.
Statistical LiDAR Filtering

Chi-Distribution Measurement

For different terrain types such as a city or a forest terrain, the height differences between neighbored pixels could be varied, but the slope differences among various terrain types should be similar and in a small range. Since most objects are removed from the method in the previous section, we are able to calculate the eight slope differences $S_i (i = 1, 2, ..., 8)$ between any pixel and its 8-connected neighbored pixels (labeled by Table 6-3) from the initial terrains.

Each slope difference, $S_i (i = 1, 2, ..., 8)$, is assumed to be a zero-mean normal distribution in the reference DTM from the ground truth. Although the histograms of obtained slope differences from the initial terrain of the site22 (Figure 6-22) look like zero-mean normal distributions, some necessary steps still need to be done to get a more accurate terrain. Under our assumption, the slope variation consisting of 8 slope differences is a 8 degree of freedom Chi distribution described by

$$S_{Chi} = \sqrt{\sum_{i=1}^{8} S_i^2}. \quad (6.5)$$

For a Chi distribution, the degree of freedom $k$ can be expressed in terms of its mean $\mu$ and standard deviation $\sigma$ by $k = \mu^2 + \sigma^2$. When the degree of freedom is equal to 8, we should get $k = \mu^2 + \sigma^2 = 8$. A novel algorithm is proposed as Figure 6-23, where $k = 8$ is the degree of freedom. This algorithm works as follows. First, the sum of squared mean and squared standard deviation is calculated. If it is greater than the degree of freedom, the absolute maximum slope difference of the point cloud is removed, which should be an object point. To separate all object points, these steps are iteratively executed while $\mu^2 + \sigma^2 > k$. Finally, the slope difference threshold $S_r$ between ground and non-ground surface is obtained by finding the absolute maximum from remaining slope differences.

Using the site22 as an example, Figure 6-24 shows its estimated degree of freedom from the assumed Chi distribution of slope differences is decreasing with our iterations. The $S_r$ is obtained and
equal to 3.0 when $\mu^2 + \sigma^2 \approx k$, where $\mu = 2.66$, standard deviation $\sigma = 0.99$, and $k = 8.06$. The histogram of remaining slope differences is showed in Figure 6-25 which tends to be a Chi distribution.

**Adaptive Height Threshold Derivation**

With the obtained slope difference threshold $S_r$, we are able to derive the adaptive height threshold which indicates the allowed growing height $\Delta h = z' - z$ for each pixel. Considering the 8-connected pixels give different allowed growing heights, the $f_{xi}$ and $f_{yi}$ are changed by adding $\Delta h_i$ to

$$f'_{xi} = f_{xi} + a_i \cdot \Delta h_i / 6g$$

$$f'_{yi} = f_{yi} + b_i \cdot \Delta h_i / 6g$$

where $\Delta h_i = z'_i - z_i$ and $a_i$ and $b_i$ are listed in Table 6-4. The $i$ represents the associated position between the center pixel and its 8-connected pixels which can be found in Table 6-1. The variable $c_i$ shown in Table 6-4 is for later reference.

Then, the new slope $S'_i$ is changed to

$$S'_i = \arctan \left( \frac{f'_{xi}^2 + f'_{yi}^2}{f'_{xi} + f'_{yi}} \right)$$

$$= \arctan \left( \frac{(f_{xi} + a_i \cdot \Delta h_i / 6g)^2 + (f_{yi} + b_i \cdot \Delta h_i / 6g)^2}{(f_{xi} + a_i \cdot \Delta h_i / 6g) + (f_{yi} + b_i \cdot \Delta h_i / 6g)} \right)$$

By combining the condition $\Delta h_i = 0$ if $S'_i = S_i = \arctan \sqrt{f_{xi}^2 + f_{yi}^2}$, the allowed growing height $\Delta h_i$ can be solved as

$$\Delta h_{i\pm} = \left\{ \begin{array}{ll}
- d_i + \text{sgn}(d_i) \sqrt{(d_i)^2 - c_i \left( \tan^2 S_i - \tan^2 S'_i \right)} \times 6g / c & \text{, if } \Delta h_{i\pm} \in R \\
0 & \text{, otherwise}
\end{array} \right.$$

where $S'_i = S \pm S_r$, $d_i = a_i f_{xi} + b_i f_{yi}$ and $c_i$ is shown in Table 6-4. In the case where $\Delta h_{i\pm} \not\in R$, there is no space for growing which leads the allowed growing height to be 0. The allowed growing
height threshold for each pixel is limited by its 8-connected slope differences which result in 8-connected
derived height limits. The minimum of these limits could underestimate ground points and restrain the
growing space of ground points to cause the Type I error; while the maximum of these limits could
overestimate ground points to cause the Type II error. Therefore, the adaptive height threshold of each
pixel is determined by the mean function of its 8-connected height limits

\[ T_{\Delta h} = \frac{1}{8} \sum_{i,j,k=5}^{9} \max(\Delta h_i, \Delta h_j, \Delta h_k). \quad (6.10) \]

Accordingly, the points in the ground and object cluster can simply be reclassified by their AGL
values. If a ground point has an AGL that is higher than its corresponding threshold \( T_{\Delta h} \), this point is
reclassified as an object point. Conversely, if an object point has an AGL that is lower than its
corresponding threshold \( T_{\Delta h} \), this point is reclassified as a ground point. This reclassification procedure
needs iteration to get the final terrain. The iterated process ends when the total number of corrected
ground and object points of the current iteration is not reduced from the previous iteration.

**Grid-Based Bared-Earth Extraction Result**

The result from this slope-based statistical approach is shown in Table 6-5. In addition, the method
is also compared to the previous approaches Chang#1 and Chang#2 in Table 6-6. It shows that the
slope-based statistical algorithm (SSA) is better than previous results in average mean and kappa index
of agreement. In addition, the standard deviation of kappa index of agreement is also less than the others
which means its performance is more stable when applied to various sites.

**Summary**

LiDAR is an important modality in terrain and land surveying for many environmental, engineering
and civil applications. Recently, an unsupervised classification algorithm called Skewness Balancing was
developed for object and ground point separation in airborne LiDAR data [76, 77]. Although the main
advantages of their algorithm are threshold-freedom and independence from LiDAR data resolution, they
have to build a prediction model to categorize LiDAR tiles as hilly or moderate terrains. However, not all
LiDAR data can be categorized as either completely hilly or moderate terrain tiles. Once a tile includes both terrain types, their algorithm will face a big challenge.

Our slope-based statistical algorithm is appropriate to any mixing or complicated terrain types. Most objects are removed and initial terrains are obtained in the object detection algorithm. Slope differences are almost similar and assumed to be a zero-mean normal distribution in all kinds of terrains, unlike absolute height information used by Skewness Balancing algorithm. Based on slope difference variations, the Chi distribution measurement is used to decide the adaptive slope threshold. Accordingly, the adaptive growing height threshold of each pixel is derived by 8-connected neighbored pixels. Finally, we demonstrate the performance of this novel algorithm by testing 15 study sites from ISPRS. It shows that this algorithm is better than algorithms Chang#1 and Chang#2 which have outperformed all other twelve algorithms working on the same study sites shown in Chapter 5.
Table 6-1. Associated z positions in the slope formula

<table>
<thead>
<tr>
<th>7</th>
<th>8</th>
<th>9</th>
</tr>
</thead>
<tbody>
<tr>
<td>4</td>
<td>5</td>
<td>6</td>
</tr>
<tr>
<td>1</td>
<td>2</td>
<td>3</td>
</tr>
</tbody>
</table>

Table 6-2. Kernel weights of the smoothing low-pass filter

| 1 | 2 | 1 |
| 2 | 4 | 2 |
| 1 | 2 | 1 |

Table 6-3. Related positions of 8-connected neighbored pixels

| 7 | 6 | 5 |
| 8 | 4 |
| 1 | 2 | 3 |

Table 6-4. Variable list for the 8-connected pixels

<table>
<thead>
<tr>
<th>$i$</th>
<th>$a_i$</th>
<th>$b_i$</th>
<th>$c_i = a_i^2 + b_i^2$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>-1</td>
<td>-1</td>
<td>2</td>
</tr>
<tr>
<td>2</td>
<td>0</td>
<td>-1</td>
<td>1</td>
</tr>
<tr>
<td>3</td>
<td>1</td>
<td>-1</td>
<td>2</td>
</tr>
<tr>
<td>4</td>
<td>-1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>6</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>7</td>
<td>-1</td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>8</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>9</td>
<td>1</td>
<td>1</td>
<td>2</td>
</tr>
</tbody>
</table>

Table 6-5. Total errors and kappa index for 15 study sites by the slope-based statistical algorithm (SSA).

<table>
<thead>
<tr>
<th>Site</th>
<th>TP</th>
<th>TN</th>
<th>FN</th>
<th>FP</th>
<th>Error</th>
<th>Kappa</th>
</tr>
</thead>
<tbody>
<tr>
<td>Samp11</td>
<td>19306</td>
<td>2480</td>
<td>1621</td>
<td>14603</td>
<td>10.8%</td>
<td>78.1%</td>
</tr>
<tr>
<td>Samp12</td>
<td>25833</td>
<td>858</td>
<td>637</td>
<td>24791</td>
<td>2.9%</td>
<td>94.3%</td>
</tr>
<tr>
<td>Samp21</td>
<td>9935</td>
<td>150</td>
<td>78</td>
<td>2797</td>
<td>1.8%</td>
<td>94.9%</td>
</tr>
<tr>
<td>Samp22</td>
<td>21984</td>
<td>520</td>
<td>580</td>
<td>9622</td>
<td>3.4%</td>
<td>92.2%</td>
</tr>
<tr>
<td>Samp23</td>
<td>12607</td>
<td>616</td>
<td>558</td>
<td>11314</td>
<td>4.7%</td>
<td>90.6%</td>
</tr>
<tr>
<td>Samp24</td>
<td>5277</td>
<td>157</td>
<td>235</td>
<td>1823</td>
<td>5.2%</td>
<td>86.7%</td>
</tr>
<tr>
<td>Samp31</td>
<td>15493</td>
<td>63</td>
<td>233</td>
<td>13073</td>
<td>1.0%</td>
<td>97.9%</td>
</tr>
<tr>
<td>Samp32</td>
<td>5451</td>
<td>151</td>
<td>194</td>
<td>5435</td>
<td>3.1%</td>
<td>93.9%</td>
</tr>
<tr>
<td>Samp32</td>
<td>12061</td>
<td>382</td>
<td>210</td>
<td>29817</td>
<td>1.4%</td>
<td>96.6%</td>
</tr>
<tr>
<td>Samp33</td>
<td>13901</td>
<td>49</td>
<td>574</td>
<td>3321</td>
<td>3.5%</td>
<td>89.2%</td>
</tr>
<tr>
<td>Samp34</td>
<td>19894</td>
<td>202</td>
<td>411</td>
<td>1949</td>
<td>2.7%</td>
<td>84.9%</td>
</tr>
<tr>
<td>Samp35</td>
<td>32498</td>
<td>409</td>
<td>405</td>
<td>981</td>
<td>2.4%</td>
<td>69.4%</td>
</tr>
<tr>
<td>Samp36</td>
<td>3832</td>
<td>150</td>
<td>220</td>
<td>4405</td>
<td>4.3%</td>
<td>91.4%</td>
</tr>
<tr>
<td>Samp37</td>
<td>33784</td>
<td>54</td>
<td>338</td>
<td>867</td>
<td>1.1%</td>
<td>81.0%</td>
</tr>
<tr>
<td>Samp38</td>
<td>13797</td>
<td>76</td>
<td>275</td>
<td>1495</td>
<td>2.2%</td>
<td>88.2%</td>
</tr>
<tr>
<td>Method</td>
<td>Error mean</td>
<td>Kappa mean</td>
<td>Kappa Standard Deviation</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>---------</td>
<td>------------</td>
<td>------------</td>
<td>-------------------------</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Chang#1</td>
<td>4.6%</td>
<td>84.5%</td>
<td>12.1%</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Chang#2</td>
<td>3.8%</td>
<td>87.3%</td>
<td>10.2%</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SSA</td>
<td>3.4%</td>
<td>88.6%</td>
<td>7.7%</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Figure 6-1. Standard deviation and confidence interval of a normal distribution.

Figure 6-2. Outlier points found by a normal distribution with 95% confidence interval at site22: 20 (black) points and 49 (white) points are lower and higher than the interval.

Figure 6-3. Edge detection at the site22: A) Smoothed slope map and B) smoothed tophat filtering at the site22
Figure 6-4. Obtained flat (black) regions at the site.

Figure 6-5. Rule-based methods for flat region classification.

Each flat region

No

Yes

#inside clusters<10

Yes

Height indicator<2.5m

Rule#1: Area size>350 & slope indicator<0°
Rule#2: Area size>100 & slope indicator<-20°
Rule#3: Area size>50 & slope indicator<-30°
Pass any rule?

No

Yes

Ground cluster

Object cluster
Figure 6-6. The associated feature maps and classification result of the site22: A) cluster labeling map B) cluster-wise inside cluster number map C) cluster-wise positive average height map D) cluster-wise signed average slope map E) cluster-wise area size map F) classification result.
Figure 6-7. DTM comparison at the site22: A) initial DTM and B) reference DTM

Figure 6-8. DTM comparison at the site11: A) initial DTM and B) reference DTM

Figure 6-9. DTM comparison at the site12: A) initial DTM and B) reference DTM
Figure 6-10. DTM comparison at the site21: A) initial DTM and B) reference DTM

Figure 6-11. DTM comparison at the site23: A) initial DTM and B) reference DTM

Figure 6-12. DTM comparison at the site24: A) initial DTM and B) reference DTM
Figure 6-13. DTM comparison at the site 31: A) initial DTM and B) reference DTM

Figure 6-14. DTM comparison at the site 41: A) initial DTM and B) reference DTM

Figure 6-15. DTM comparison at the site 42: A) initial DTM and B) reference DTM
Figure 6-16. DTM comparison at the site 51: A) initial DTM and B) reference DTM

Figure 6-17. DTM comparison at the site 52: A) initial DTM and B) reference DTM

Figure 6-18. DTM comparison at the site 53: A) initial DTM and B) reference DTM
Figure 6-19. DTM comparison at the site54: A) initial DTM and B) reference DTM

Figure 6-20. DTM comparison at the site61: A) initial DTM and B) reference DTM

Figure 6-21. DTM comparison at the site71: A) initial DTM and B) reference DTM
Histogram of $S_1$: $\mu=0.15$, $\sigma=3.05$

Histogram of $S_2$: $\mu=0.10$, $\sigma=2.22$

Histogram of $S_3$: $\mu=0.41$, $\sigma=3.90$

Histogram of $S_4$: $\mu=0.22$, $\sigma=2.91$

Histogram of $S_5$: $\mu=0.26$, $\sigma=3.06$

Histogram of $S_6$: $\mu=0.09$, $\sigma=2.26$

Histogram of $S_7$: $\mu=0.26$, $\sigma=3.70$

Histogram of $S_8$: $\mu=0.10$, $\sigma=2.87$

Figure 6-22. Histograms of slope differences from the initial terrain at the site22
Load LiDAR point cloud

\[ \text{while } \mu^2 + \sigma^2 > k \text{ do} \]

Remove absolute maximum slope difference

\[ \text{end while} \]

Get slope difference threshold, \( S_T \), between ground and non-ground surface by finding absolute maximum from remaining slope differences.

Figure 6-23. Adaptive slope difference threshold algorithm based on Chi distribution

Figure 6-24. The change curve of freedom from slope differences with iterations at the site

Figure 6-25. Histogram of remaining slope differences which tends to be a Chi distribution at the site
CHAPTER 7
OCCLUDED VEHICLE DETECTION DESIGN

Vehicle detection has been utilized in the Intelligent Transportation System (ITS), Automatic Vehicle Guidance (AVG), and traffic flow estimation, but has not been exploited in the forested terrain. LiDAR employs an active optical modality or laser ranging that provides primarily geometric information to detect natural surface features and other hard targets that may be spectrally inseparable in multi-spectral passive optical imagery. In addition, airborne LiDAR can provide data in large spatial extents with varying temporal resolution and it can be deployed more or less anywhere and at any time which restricts the use of passive optical imagery including in smoke, haze, fog, and at night. Thus, occluded vehicle detection from airborne LiDAR data in forested terrain can be applied to many fields, more specifically: 1) military surveillance – searching enemy vehicles in a battle area with forest, 2) homeland security – border crossing monitoring for vehicles in forest area, 3) global warming – vehicle hunting for illegal deforestation which is a hidden cause of global warming, 4) disaster rescue – finding vehicles stuck by disrupted roads in forest during natural disaster, 5) emergency road service – locating vehicles involved with general car accidents in forest, and 6) criminal searching – uncovering forest canopy to search suspicious vehicles hiding in mountains.

The occluded objects underneath trees can be revealed though the developed canopy removal algorithm in the beginning. The obtained uncovered LiDAR points are clustered into individual objects by the proposed DTM extraction algorithm and the associated morphological image processing, including consideration of horizontal and vertical orientations. The clustered LiDAR points of each object will be exploited by many theories such as Spin image, non-parameteric Parzen-window estimation, Bayesian decision, and relative entropy, etc. A probabilistic modeling is built up to detect those occluded vehicles in forested terrain from airborne LiDAR point clouds. Finally, we verify our results by examining the Receiver Operating Characteristic (ROC) curves.
Object Clustering

Cluster analysis, or clustering, is the assignment of a set of observations into subsets (called clusters) so that observations in the same cluster are similar in some sense. Clustering is a method of unsupervised learning, and a common technique for statistical data analysis used in many fields, including machine learning, data mining, pattern recognition, image analysis and bioinformatics [78]. In the application of image segmentation, clustering can be used to divide a digital image into distinct regions for border detection or object recognition.

Object recognition in computer vision is the task of finding a given object in an image or video sequence. Humans recognize a multitude of objects in images with little effort, despite the fact that the image of the objects may vary somewhat in different view points, in many different sizes/scale or even when they are translated or rotated. Objects can even be recognized when they are partially obstructed from view. This task is still a challenge for computer vision systems in general [79].

A good approach for object recognition is the scale-invariant feature transform (or SIFT) which is an algorithm in computer vision to detect and describe local features in images. SIFT key points of objects are first extracted from a set of reference images and stored in a database. An object is recognized in a new image by individually comparing each feature from the new image to this database and finding candidate matching features based on Euclidean distance of their feature vectors. From the full set of matches, subsets of key points that agree on the object and its location, scale, and orientation in the new image are identified to filter out good matches. The determination of consistent clusters is performed rapidly by using an efficient hash table implementation of the generalized Hough transform. Each cluster of 3 or more features that agree on an object and its pose is then subject to further detailed model verification and subsequently outliers are discarded. Finally the probability that a particular set of features indicates the presence of an object is computed, given the accuracy of fit and number of probable false matches. Object matches that pass all these tests can be identified as correct with high confidence [80].

In addition, k-means clustering [81] is a method of cluster analysis which aims to partition
observations into \( k \) clusters in which each observation belongs to the cluster with the nearest mean. The k-means clustering algorithm is commonly used in computer vision as a form of image segmentation. The results of the segmentation are used to aid border detection and object recognition. The two key features of k-means which make it efficient are often regarded as its biggest drawbacks: 1) the number of clusters \( k \) is an input parameter and an inappropriate choice of \( k \) may yield poor results; 2) Euclidean distance is used as a metric and variance is used as a measure of cluster scatter.

However, our proposed object clustering method does not have the need to create a database from the training data in the SIFT or to choose the number of clusters \( k \) and use the Euclidean distance as a metric in the k-means clustering. It is important to obtain the whole shape of each object by clustering analysis for identification, but it is a challenge to do when vehicles are mixed with other objects and irregular canopy occlusion will more or less reduce reflected points from underneath objects. Thus, it will make different objects which are close to one another become hardly segmented since the gaps between objects could be occluded such that the number of points reflected from the gaps become impacted. Therefore, we developed a feasible way for the object segmentation (Figure 7-1), which includes horizontal-based and vertical-based morphological filtering, following our canopy removal algorithm and ground point filtering algorithm.

**Horizontal Based Morphological Filtering**

First, the DTM is a regular grid image which can be generated by resampling irregular ground points extracted from our ground point filtering, where the grid resolution is 5 pixels by 5 pixels per square meter which was approximately equal to our average LiDAR density. By referring to the DTM, the individual subtractions with corresponding elevation of DTM from the original height of points are used to obtain the above ground levels (AGL) of all non-canopy points, which were remaining points after removing canopy points above vehicles. Then, the DSM is generated by resampling irregular non-canopy points with their AGL values in the same grid resolution as the DTM. Finally, two kinds of segmentations are followed to cluster individual object points: horizontal-based and vertical-based morphological filtering. Some summary of morphological filtering operations is shown in Table 7-1.
The horizontal-based segmentation is used to separate object points which are not close enough in the horizontal direction or x-y coordinates to form an individual object. The opening operator is performed first by a disk-shaped structuring element (SE) with a radius of two pixels on the DSM. As such, those objects are partially connected and their lower parts can be segmented away. The disk type of structure element is chosen as it is orientation invariant and the orientations of objects could be in any direction.

In the seriously occluded situation, the partial connection between two objects could be larger than the SE of the opening operator to make the above segmentation fail. However, the bot-hat operator is useful for enhancing detail in the presence of local bottoms [61]. Thus, the bot-hat operator using a disk-shaped SE with a radius of ten pixels is applied to find a channel-like structure where the center is lower than its neighbor. Using both opening and bot-hat morphological filters we are able to segment two close objects with short or long partial connections. Therefore, this spacing-based segmentation is very helpful to separate those vehicles which are close to one another in the parking lot.

**Vertical Based Morphological Filtering**

The vertical-based segmentation is applied to separate those points which are close in vertical direction, or z coordinate, but belong to different objects. In the process of removing tree canopy above vehicles, some vegetation points lower than the maximum height of vehicles still remained since they could be parts of potential vehicles and need to be further investigated. Besides, some vehicles could be close to buildings and the thin gaps between them could not reflect any points due to irregular tree canopies. Both cases need segmentation in the vertical direction.

We proposed a simple but effective method consisting of the horizontal-based segmentation described above with two elevation limitations. The first limit of elevation is to filter the DSM by examining and removing those pixels higher than 2 meters, which is referred to as the average heights of sedans, SUVs, and pickup trucks. This modified DSM is then applied to the horizontal-based segmentation with the same opening and bot-hat morphological filters. In this way, those noise points located above vehicles can easily be removed to help with further vehicle identification.
Buses, 18-wheel trucks and military equipment (e.g., tanks) are higher than 2 meters but usually lower than 4 meters. Therefore, if these large vehicles exist in the dataset, then another DSM has to be obtained by removing those pixels higher than the second limit of elevation, 4 meters. Similarly, the spacing-based segmentation above is followed again to achieve the elevation-based segmentation. Accordingly, those points belonging to one object can be clustered by the object segmentation from both spacing-based and elevation-based morphological filtering methods.

**Object Clustering Result**

Figure 7-2, Figure 7-3, and Figure 7-4 show the results of object segmentation in Hogtown parking site, Hogtown forest site, and the residential site, respectively. Comparing these to Figure 4-11b, Figure 4-12b, and Figure 4-13b, those segmented objects achieve a promised accomplishment since nearly all points belonging to vehicles are clustered individually and segmented with other non-vehicles.

When considering the corresponding LiDAR density in these three study sites, it was calculated that the average LiDAR densities after removing canopy points are 15.77, 9.73, and 18.98 points per square meter and the standard deviation is 8.68, 6.45, and 8.41 for Hogtown parking site, Hogtown forest site, and the residential site, respectively. The entire space distribution for LiDAR density in these three study sites can be seen in Figure 4-14a, Figure 4-14b, and Figure 4-14c. However, it is noted that even if the LiDAR points are from the laser scanning swath of only one flight pass, in which the point density could be only one sixth or one seventh of the current density, then this object segmentation method combining spacing-based and elevation-based morphological filtering still can work well as is shown in a later section.

**Object Classification**

After clustering points as segmented objects, the whole shape of individual objects can be formed and their special features can be extracted and analyzed for this object classification: either a vehicle or a non-vehicle class. If vehicles were in an open area which was not occluded by trees, the vehicle/non-vehicle classification could be much easier, such as in typical research for estimating traffic flow. For occluded vehicles underneath trees, two problems emerge for vehicle/non-vehicle classification:
the intra-class problem and inter-class problem. The former problem is that even if the vehicle size is variable due to not only vegetation occlusion but also different types, manufacturing companies, and models, all un-occluded parts of different vehicles have to be recognized as the vehicle class. The latter problem is how to distinguish those whole or partial vehicles with variable size from variable non-vehicle objects. We propose the following approach to solve both intra-class and inter-class problems simultaneously. The block diagram of our object classification is shown in Figure 7-5.

**Principal Component Analysis**

Principal component analysis (PCA) [82] involves a mathematical procedure that transforms a number of possibly correlated variables into a smaller number of uncorrelated variables called principal components. The first principal component accounts for as much of the variability in the data as possible, and each succeeding component accounts for as much of the remaining variability as possible. PCA is the simplest of the true eigenvector-based multivariate analyses. Often, its operation can be thought of as revealing the internal structure of the data in a way which best explains the variance in the data. If a multivariate dataset is visualized as a set of coordinates in a high-dimensional data space (1 axis per variable), PCA supplies the user with a lower-dimensional picture, a “shadow” of this object when viewed from its (in some sense) most informative viewpoint.

Component analysis is an unsupervised approach to finding the “right” features from the data. In PCA [83], we usually represent the d-dimensional data in a lower-dimensional space. This will reduce the degrees of freedom, reducing the space and time complexities. The common goal is to represent data in a space that best describes the variation in a sum-squared error sense. One example of using PCA is to get the best fitting plane from irregular 3-dimensional points. Figure 7-6 shows that the first two principal components are obtained which are orthogonal each other and can define vectors that form a basis for the plane. The third PC is also orthogonal to the first two, and it defines the normal vector of the plane.

We apply PCA to find the length and width of each object. First, each object consists of clustered non-canopy points with their AGL heights. All vehicles can be considered as rectangle shapes when
projecting to x-y coordinates. The length and width of vehicles can be a good pre-filter to remove non-vehicle objects. Since the orientation of every object is arbitrary, PCA [83] is a good tool to get its orientation and find both length and width of any object. Let \( v \) be the \( N \times 2 \) matrix consisted of \( N \) points with x-y coordinate pairs. The 2-D mean vector \( \mu \) and 2×2 covariance matrix \( \Sigma \) are computed for the data set \( v \). Next, the eigenvectors and eigenvalues are computed and sorted according to decreasing eigenvalue. Then, the 2×2 matrix \( A \) is formed whose columns consist of the 2 eigenvectors. The representation of data by principal components consists of projecting the data onto 2-D according to [83]

\[
v' = A' (v - \mu),
\]

where \( v' \) consists of two columns \( x' \) and \( y' \). Therefore, the length, \( L \), and width, \( W \), of an object can be obtained by

\[
L = \max(x') - \min(x') \tag{7.2}
\]

\[
W = \max(y') - \min(y') \tag{7.3}
\]

Taking account of the original vehicle size and acceptable error margin, the length and width of vehicle candidates should be less than 8 meter and 4 meter, respectively. Thus, if the length or width of one object is greater than the corresponding threshold, it will be classified as a non-vehicle object, while those vehicle candidates which satisfied the length and width criterions continue to be examined and their features will be extracted from the Spin image.

**Spin Image**

The spin image [84] is a method of surface matching which is the process that compares surfaces and decides whether they are similar. Surface matching can be used for object recognition by analyzing and comparing the features extracted from object surfaces. Surface matching is difficult because the coordinate system in which to compare two surfaces is undefined, characteristics of sensed data, including clutter, occlusion and sensor noise. Surface matching is further complicated in sparse LiDAR point density situations.

The spin image describes a data level representation of surfaces used for surface matching. In its...
representation, surface shape is described by a collection of oriented points (3-D points with a surface normal). Using a single point basis constructed from an oriented point, the position of other points on the surface can be described by two parameters. The accumulation of these parameters for many points on the surface results in an image at each oriented point. These images, localized descriptions of the global shape of the surface, are invariant to rigid transformations. Through correlation of images, point correspondences between two surfaces can be established. When two surfaces have many point correspondences, they match. Taken together, the oriented points and associated images make up the surface representation. Because the image generation process can be visualized as a sheet spinning about the normal of a point, the images in this kind of representation are called spin-images.

The two coordinates of the basis in spin-images are $\alpha$, the perpendicular distance to the line $L$, and $\beta$, the signed perpendicular distance to the plane $P$. An oriented point basis is a cylindrical coordinate system that is missing the polar angle coordinate because this coordinate cannot be determined using just surface position and normal. Using an oriented point basis $O$, a spin-map $S_o$ is defined as the function in [84] that projects 3-D points $x$ to the 2-D coordinates of a particular basis $(p, n)$ corresponding to oriented point $O$.

\[
S_o : R^3 \rightarrow R^2
\]

Then for each vertex $x$ on the surface of the object, the spin-map coordinates with respect to $O$ are computed and bilinearly interpolated to form a digitalized 2-D array. The pixel value which each point $x$ is spin-mapped would be to increment by one. Once all of the points on the surface of the object have been accumulated, a 2-D array representation of the spin-image is generated.

In our vehicle detection, the oriented plane is chosen as the local ground surface of the mapped object, where its normal vector $n$ is orthogonal to this oriented plane, while the oriented point $p$ is set as the center point of the object on the ground level. Then, the new 2-D coordinates $(\alpha, \beta)$ of the Spin map can be obtained from 3-D points of the object by Equation (7.5). In order to count points of the Spin map
inside each bin of the Spin image, the contribution of each point is bilinearly interpolated to the four surrounding bins in the 2-D array making the array less sensitive to the position of the point, where the bin size is set as 0.2 meter, approximately equal to LiDAR vertical accuracy. Once all resampling points of the object have been accumulated to their corresponding bins, a 2-D array representation of the Spin image is generated. Taking some examples of collected vehicle data with variable occluded degree, their 3-D shapes from irregularly reflected points are generated by the Delaunay triangulation to see how those vehicles were seriously impacted by the occlusion, and their corresponding 2-D Spin images are obtained and shown in Figure 7-7 through Figure 7-16, sorted by the actual collected number of LiDAR points from individual vehicles.

Although irregularly occluded vehicles are variable in length, width, height, and shape, they could be transformed to the Spin images, $SI(\alpha, \beta)$, to analyze and extract some important features. By our observation and analysis on those Spin images of vehicles, two special features, $\beta_\mu$ and $\alpha_\sigma$, are extracted according to

$$\beta_\alpha = \arg \max_\beta (SI(\alpha, \beta)), \quad \alpha = 1, 2, \ldots, n$$

(7.6)

$$\beta_\mu = \frac{1}{n} \sum_{\alpha=1}^{n} \beta_\alpha$$

(7.7)

$$\alpha_\sigma = \sqrt{\frac{1}{\gamma} \sum_{\alpha=1}^{n} \sum_{\beta=1}^{m} (\beta - \beta_\alpha)^2 SI(\alpha, \beta)},$$

(7.8)

where

$$\gamma = \sum_{\alpha=1}^{n} \sum_{\beta=1}^{m} SI(\alpha, \beta),$$

(7.9)

$$n = \max[\alpha : SI(\alpha, \beta) \neq 0],$$

(7.10)

and

$$m = \max[\beta : SI(\alpha, \beta) \neq 0].$$

(7.11)

The individual locations of $\beta_\alpha$ are marked by black circles for illustration in the Spin images of Figure 7-7 to Figure 7-16, where all locations of $\beta_\alpha$ are connected by piecewise lines. By using the spin
image, points are accumulated and counted if they have the same horizontal distance, $\alpha$, and the same vertical distance, $\beta$, away from the oriented point. The spin image benefits the vehicle detection in the following two ways: 1) we can take advantage of the symmetry of a vehicle where the shape of the right side is the same as the left side in the viewpoint of the front or back; 2) the spin image is rotation invariant which avoids the need to detect the orientation of diverse and occluded vehicles. Even though PCA can be helpful to determine the orientation of vehicles, it becomes more difficult when facing the situation with sparser point density. Therefore, we can feasibly apply the spin image to LiDAR data on the vehicle detection in a sparse point density and an arbitrary vehicle orientation.

Using the extracted features $\beta_\mu$ and $\alpha_\sigma$ of vehicle candidates including vehicles and non-vehicles, two corresponding bivariate non-parametric PDFs (Probability Density Function) can be estimated by the Parzen windowing method with a Gaussian kernel function [49] from the training data set. The optimal window size selection can be determined by Silverman’s formula [85],

$$\sigma_{opt} = \sigma \left[ \frac{4}{(2d + 1)N} \right]^{\frac{1}{d+4}},$$

where $d$ is the data dimensionality, $N$ is the sample size of the data, and $\sigma_x = d^{-1} \sum_i x_i x_i$, where $\sum x_x$ are the diagonal elements of the sample covariance matrix. Accordingly, the bivariate PDF of vehicles and non-vehicles constructed from two features $\beta_\mu$ and $\alpha_\sigma$ is obtained and shown in Figure 7-17 and Figure 7-18, respectively.

**Bayesian Decision**

Bayesian decision theory is a fundamental statistical approach to the problem of pattern classification by using probability and the costs associated to decisions. The Bayesian decision rule [83] is to decide $\omega_1$ if

$$\frac{p(x|\omega_1)}{p(x|\omega_2)} > \frac{(\lambda_{12} - \lambda_{22})p(\omega_2)}{(\lambda_{21} - \lambda_{11})p(\omega_1)}$$

(7.13)
where \( \lambda_{ij} = \lambda(\omega_i | \omega_j) \) is the loss incurred for deciding \( \omega_i \) when the true state of nature is \( \omega_j \), \( p(x|\omega_j) \) is a function of \( \omega_j \) (i.e., the likelihood function) and \( p(\omega_j) \) is a priori probability of \( \omega_j \). Thus the Bayesian decision rule can be interpreted as calling for deciding \( \omega_i \) if the likelihood ratio, 
\[
p(x|\omega_i)/p(x|\omega_2),
\]
exceeds a threshold value represented by the term in the right hand side of Equation (7.13) which is independent of the observation \( x \).

In our study, \( \omega_1 \) and \( \omega_2 \) is the class for vehicle and non-vehicle, respectively. Given a test object with two special features \( \beta_\mu \) and \( \alpha_\sigma \), the probability in the vehicle/non-vehicle category can be found from the obtained bivariate PDF of vehicles/non-vehicles. If the probability in the vehicle category is higher than in the non-vehicle category or the ratio of the former over the latter is greater than 1, then the test object is classified as a vehicle; otherwise, it is classified as a non-vehicle.

**Relative Entropy**

The relative entropy or Kullback-Leibler distance is a measure tool to calculate the distance between two distributions. It is strongly related to information divergence and information for discrimination which discrete version is defined as [83]

\[
D_{KL}(p(x), q(x)) = \sum_x q(x) \ln \frac{q(x)}{p(x)}
\]

(7.14)

where \( p(x) \) and \( q(x) \) are two discrete distributions over the same variable \( x \). Its continuous version [83] is

\[
D_{KL}(p(x), q(x)) = \int_{-\infty}^{\infty} q(x) \ln \frac{q(x)}{p(x)} dx
\]

(7.15)

It is noted that the \( D_{KL}(p(.), q(.)) \geq 0 \) and \( D_{KL}(p(.), q(.)) = 0 \) if and only if \( p(.) = q(.) \).

In our study, we apply this relative entropy to the Bayesian decision. First, the right hand side of the Bayesian decision equation can be regarded as a Bayesian threshold (BT).

\[
\frac{p(x|\omega_1)}{p(x|\omega_2)} > \frac{(\lambda_{12} - \lambda_{22})P(\omega_2)}{(\lambda_{21} - \lambda_{11})P(\omega_1)} = BT
\]

(7.16)
Since we want to get optimal decision for vehicles, the PDFs between referenced vehicles and classified vehicles should be as similar as possible. The relative entropy can be used to measure the distance between the distributions of referenced vehicles and classified vehicles. With varying the $BT$ in a smart way, such as where one tests the large scale first to find a good threshold and obtain small range for the small scale trial, the various relative entropy measurements can be obtained for vehicle detection. Of course, the smaller the relative entropy is, the better the $BT$ is.

However, the Bayesian decision is based on quantifying the tradeoffs between various classification decisions using probability. Similarly, our approach cannot ignore the relative entropy for non-vehicle detection in order to achieve the optimal decision for considering both vehicles and non-vehicles. Therefore, the best trade off, or the optimal $BT$ value, is achieved by considering two relative entropies between vehicle and non-vehicle detection.

In our case, the relative entropy between two distributions of vehicles and non-vehicles based on extracted features $\beta_\mu$ and $\alpha_\sigma$ is fixed. But, the distribution of detected vehicles/non-vehicles can be changed with the adjustable variable $BT$ such that the relative entropy between two distributions of reference vehicles and detected vehicles is varied, and so is that between reference non-vehicles and detected non-vehicles. where these two situations are shown as the curves for vehicles and non-vehicles, respectively, in Figure 7-19. Therefore, the relative entropy can help to examine and evaluate which $BT$ can make our detected distribution of vehicle/non-vehicle be as close as possible to the true distribution of vehicle/non-vehicle.

After measuring the $D_{KL}$ information divergences between reference and detection in both vehicles and non-vehicles, it is observed in Figure 7-19 that two curves, $D_{KL}$ functions of the $BT$, for vehicles and non-vehicles approach each other initially and diverge gradually in the sense of ascending $BT$. The detectability for non-vehicles gradually becomes worse but it turns better for vehicle detection and then turns worse. The optimal $BT$ here is one that achieves the best overall detection for both vehicles and non-vehicles. If the $BT$ is solved by finding the lowest average of summing two $D_{KL}$ curves,
it cannot be guaranteed as the best solution because the averaging is equal to giving the same weighting coefficients while the prior probabilities of vehicles and non-vehicles are different and probably unknown. Instead of the summing method, the subtracting method is proposed. From Figure 7-19, it is observed that the detectability for non-vehicles is always stronger than the detectability for vehicles. Thus, the best $BT$ should be able to suppress the detectability for non-vehicles and boost the detectability for vehicles simultaneously, which inspires the following equation.

$$BT_{opt} = \arg \min_{BT} \left( D_{KL,\text{Vehicle}}(BT) - D_{KL,\text{NonVehicle}}(BT) \right)$$ (7.17)

Actually, the above difference of $D_{KL}$ values between the vehicle and non-vehicle class is the indicator of detection bias. The larger difference means the higher preference of discriminability to non-vehicles. For example, a $BT$ smaller than 0.4 makes the detection of non-vehicles pretty good since its corresponding $D_{KL}$ is almost 0, while the detection of vehicles becomes very poor which are observed by those large corresponding $D_{KL}$ values. Equation (7.17) can also be interpreted to give better detection for vehicles and balance the detection bias between vehicles and non-vehicles. Therefore, the optimal $BT$ of 0.7 can be found from Equation (7.17). The bivariate PDF of detected vehicles and non-vehicles with setting $BT$ as 0.7 is shown in Figure 7-20 and Figure 7-21, respectively.

**Occluded Vehicle Detection Result and Evaluation**

By using the Bayesian decision rule in Equation (7.16) and the optimal $BT$ in Equation (7.17) determined by the associated information divergence, each segmented object with its extracted features $\beta_\mu$ and $\alpha_\sigma$ can be classified as either a vehicle or a non-vehicle based on generated bivariate distributions $p(\alpha_\sigma, \beta_\mu | \omega_1)$ and $p(\alpha_\sigma, \beta_\mu | \omega_2)$ from the reference vehicle class in Figure 7-17 and reference non-vehicle class in Figure 7-18, respectively. The reference vehicle/non-vehicle bivariate distributions are trained by 1658 objects at Hogtown parking site and Hogtown forest site from individual and overlapped laser scans. The test site, Hogtown residential site, includes 416 objects from individual and overlapped laser scans which are never used for training and have no contribution to the true
vehicle/non-vehicle bivariate distributions. However, all of the 2074 objects are tested and classified by the Bayesian decision rule with our optimal $BT$. Our detected vehicle/non-vehicle objects are compared to true vehicle/non-vehicle objects by the cross matrix in Table 7-2, where the False Negative (FN) case, or Type I error, is for falsely rejecting vehicles, and the False Positive (FP) case, or Type II error, is for incorrectly accepting non-vehicles. The remaining True Positive (TP) and True Negative (TN) cases are for the correct classifications of vehicles and non-vehicles, respectively. The detection accuracy [4] is obtained by Equation (7.18) composed of the TP, FN, FP, and TN.

$$\text{Accuracy} = \frac{TP + TN}{TP + FN + FP + TN} \times 100\%$$  \hspace{1cm} (7.18)

For Hogtown parking site, the result of vehicle detection for overlapped laser scans is shown in Figure 7-22a, where the white lines and black lines represent detected vehicles and non-vehicles, respectively. Similarly, the result of vehicle detection for a single scan (scan #6) is shown in Figure 7-22b. Comparing to Figure 4-11b, the TP, FN, FP, and TN can be counted and the accuracy can be obtained by Equation (7.18). Accordingly, the overlapped and single-scanned accuracy are 95.4% and 85.5%, respectively. It is noted that the average under canopy LiDAR density of a single scan for this site is 2.25 points per square meter with a standard deviation of 1.24 points per square meter. Even though those vehicles are occluded by irregular tree structures and reflected by different laser scans, our vehicle detection accuracy for single scans still range from 83.8% to 88.2%. Besides, the overall accuracy by testing 947 objects from overlapped and single scans can reach 87.9% (Table 7-3).

For Hogtown forest site, the result of vehicle detection for its overlapped laser scans and single scan (scan #6) are shown in Figure 7-23. Similarly, this result can be compared to Figure 4-12b and TP, FN, FP, and TN can be counted to obtain the detection accuracy. The accuracies are 80.5% and 83.2% for overlapped scans and the single scan, respectively (Table 7-4). The average under canopy LiDAR density of single scan for this site is 1.62 points per square meter with a standard deviation of 1.07 points per square meter.

Due to the seriously occluded scenario in this forest site, where its occluded rate of 80.59% is
much higher than the 63.51% in the parking site, both available LiDAR density and detected accuracy for underneath objects are affected and reduced. In addition, since there is only one tarpaulin settled as a mimic vehicle source in this site, the obtained information for detecting vehicles under this heavy occlusion should be insufficient no matter how many laser scans were rendered. Therefore, if more vehicle sources were set up and collected under this forest, the bivariate distribution of true vehicles can be improved and so can the vehicle detectability. Otherwise, even if all single scan data are overlapped for detecting, its performance could be worse than that from only single scan. It is because those detailed vehicle-like objects could be incorrectly classified as vehicles if those counterparts of detailed true vehicles are missing in the statistical based decision. However, our overall accuracy by testing 711 objects from overlapped and single scans can still get 81.0% (Table 7-4) due to the large amount of training data from 13 LiDAR scans.

For Hogtown residential site, the vehicle detection results for overlapped scans and single scan (scan #3) are shown in Figure 7-24, where the accuracies are 94.7% and 83.3%, respectively (Table 7-5). Obviously, this performance is much better than that in the Hogtown forest site. It is the degree of occlusion that makes the major difference of detection, where the average under canopy LiDAR density of single scan for this site is 3.16 points per square meter with a standard deviation of 1.40 points per square meter. Although the average under-canopy LiDAR density is a little higher than that in Hogtown parking site, the performance does not become better. A major reason is that all the objects in this site have never been used for learning in the bivariate probability density functions of vehicle class and non-vehicle class. It also explains why the overall accuracy for this site drops to 80.8% (Table 7-5). In addition, its standard deviation of under-canopy LiDAR density is larger than the other sites which can explain the larger dynamic range of the detection accuracy from 71.2% to 94.7% (Table 7-5).

On the other hand, the average detection accuracy is examined based on the number of collected LiDAR points from each object. In this way, the quality of detection performance can be observed by the quantity of LiDAR points. First, all objects are recorded with their numbers of collected LiDAR points. Instead of mixing all objects in one site to count TP, FN, FP, and TN values, they are separated into
different groups to count those values, where each group consists of only one specific number of collected LiDAR points. Figure 7-25 shows that the relationship between the numbers of collected LiDAR points from each object and the vehicle detection performance including TP, FN, FP, and TN in the case of a Bayesian threshold equal to 0.7. It can be observed that the TP and TN increase with increasing abscissa while FN and FP decrease with increasing abscissa, which demonstrates that a higher number of collected LiDAR points from a test object will give the better system detectability.

Accordingly, another relationship between the number of collected LiDAR points from each object and the vehicle detection accuracy, hit ratio, and false alarm in the case of Bayesian threshold equal to 0.7 can be obtained and is showed in Figure 7-26, where the hit ratio and false alarm formula can be derived from [49] as the following equations.

$$\text{Hit Ratio} = \frac{TP}{TP + FN} \times 100\%$$ (7.19)

$$\text{False Alarm} = \frac{FP}{FP + TN} \times 100\%$$ (7.20)

Obviously, the accuracy curve in Figure 7-26 is a monotonic increasing function since the hit ratio and false alarm generally increase and decrease with increasing abscissa, respectively. It is noted that the detection accuracy for different collected LiDAR points per object is always beyond 80%, even though the collected point number of a test object is below 5 points (Figure 7-26). It is because not only the points reflected from a testing object but also its neighboring ground points are imposed to extract associated features.

Finally, the optimal Bayesian threshold is examined by the receiver operating characteristic (ROC) curves [49], where the system performance is presented by the false alarm and hit ratio simultaneously. The ROC (Receiver Operating Characteristics) curves of our detection performance are shown for 5, 10, 15, and 20 sampling points on one object and 0 sampling points for reference in Figure 7-27. For each curve excluding the reference curve, the Bayesian thresholds 0.5, 0.6, 0.7, 0.8, 0.9 and 1.5 are marked in ascending order. It is observed that the Bayesian threshold 0.7, or the third mark counted from the bottom...
up of each curve consisting of 7 piece-wise lines, is the turn point, where the slopes of lines before this point is greater than 45° while the slopes of lines after this point is smaller than 45°. This phenomenon explains that the optimal Bayesian threshold is 0.7. Although this ROC curve can examine the system performance to get the optimal Bayesian threshold, the procedure to obtain those TP, FN, FP, and TN values for hit ratio and false alarm is time consuming since each classified object has to be verified as the TP, FN, FP, or TN case. However, using the proposed method of relative entropy is an efficient and effective way to analyze and get the optimal Bayesian threshold since the vehicle/non-vehicle distribution can be created directly from the classified objects without having the need to check the recognition result of each classified object.

**Summary**

We demonstrate that the state-of-the-art airborne LiDAR system can provide valuable data which can effectively support the occluded vehicle detection in forest terrain. The proposed system is a probabilistic model built through the canopy removal algorithm, DTM extraction algorithm, morphological image processing, PCA, Spin image, Parzen-window estimation, Bayesian decision, and relative entropy, along with verification of ROC curves. Based on the statistics for the number of collected LiDAR points from each object, the average vehicle detection accuracy is always over 80%, even though there are only less than 5 points reflected from the testing object. In addition, the probabilistic-based system performance could be easily promoted if the amount of vehicle sources and the variety of occluded scenarios could be increased in the learning phase. The potential applications for this work include many fields such as military surveillance, homeland security, global warming, disaster rescue, emergency road service, and criminal searching.
Table 7-1. Summary of some morphological filtering operations.

<table>
<thead>
<tr>
<th>Operation</th>
<th>Equation</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>Translation</td>
<td>$B_z = {w</td>
<td>w = a + z, \text{ for } a \in B}$</td>
</tr>
<tr>
<td>Reflection</td>
<td>$\hat{B} = {w</td>
<td>w = -b, \text{ for } b \in B}$</td>
</tr>
<tr>
<td>Dilation</td>
<td>$A \ominus B = {z</td>
<td>(\hat{B})_z \cap A \neq \emptyset}$</td>
</tr>
<tr>
<td>Erosion</td>
<td>$A \oplus B = {z</td>
<td>(B)_z \subseteq A}$</td>
</tr>
<tr>
<td>Opening</td>
<td>$A \circ B = (A \ominus B) \oplus B$</td>
<td>Smoothes contours, breaks narrow isthmuses, and eliminates small islands and sharp peaks.</td>
</tr>
<tr>
<td>Closing</td>
<td>$A \bullet B = (A \ominus B) \ominus B$</td>
<td>Smoothes contours, fuses narrow breaks and long thin gulfs, and eliminates small holes.</td>
</tr>
</tbody>
</table>

Table 7-2. Vehicle detection cross matrix

<table>
<thead>
<tr>
<th>Classified Vehicles</th>
<th>Classified Clutters</th>
</tr>
</thead>
<tbody>
<tr>
<td>Reference Vehicles</td>
<td>True Positive (TP)</td>
</tr>
<tr>
<td>Reference Clutters</td>
<td>False Positive (FP)/Type II Error</td>
</tr>
<tr>
<td></td>
<td>False Negative (FN)/Type I Error</td>
</tr>
<tr>
<td></td>
<td>True Negative (TN)</td>
</tr>
</tbody>
</table>

Table 7-3. Vehicle detection accuracy for Hogtown parking site with different LiDAR scans and overlapped all LiDAR scans.

<table>
<thead>
<tr>
<th>LiDAR</th>
<th>TP</th>
<th>FN</th>
<th>FP</th>
<th>TN</th>
<th>Sum</th>
<th>Accuracy</th>
</tr>
</thead>
<tbody>
<tr>
<td>Scan #1</td>
<td>6</td>
<td>2</td>
<td>15</td>
<td>91</td>
<td>114</td>
<td>85.1%</td>
</tr>
<tr>
<td>Scan #2</td>
<td>6</td>
<td>2</td>
<td>17</td>
<td>92</td>
<td>117</td>
<td>83.8%</td>
</tr>
<tr>
<td>Scan #3</td>
<td>6</td>
<td>2</td>
<td>14</td>
<td>89</td>
<td>111</td>
<td>85.6%</td>
</tr>
<tr>
<td>Scan #4</td>
<td>5</td>
<td>3</td>
<td>9</td>
<td>77</td>
<td>94</td>
<td>87.2%</td>
</tr>
<tr>
<td>Scan #5</td>
<td>7</td>
<td>1</td>
<td>15</td>
<td>113</td>
<td>136</td>
<td>88.2%</td>
</tr>
<tr>
<td>Scan #6</td>
<td>7</td>
<td>1</td>
<td>16</td>
<td>93</td>
<td>117</td>
<td>85.5%</td>
</tr>
<tr>
<td>Scan #7</td>
<td>7</td>
<td>1</td>
<td>9</td>
<td>68</td>
<td>85</td>
<td>88.2%</td>
</tr>
<tr>
<td>All Scans</td>
<td>8</td>
<td>0</td>
<td>8</td>
<td>157</td>
<td>173</td>
<td>95.4%</td>
</tr>
<tr>
<td>Overall</td>
<td>52</td>
<td>12</td>
<td>103</td>
<td>780</td>
<td>947</td>
<td>87.9%</td>
</tr>
</tbody>
</table>

Table 7-4. Vehicle detection accuracy for Hogtown forest site with different LiDAR scans and overlapped all LiDAR scans.

<table>
<thead>
<tr>
<th>LiDAR</th>
<th>TP</th>
<th>FN</th>
<th>FP</th>
<th>TN</th>
<th>Sum</th>
<th>Accuracy</th>
</tr>
</thead>
<tbody>
<tr>
<td>Scan #1</td>
<td>0</td>
<td>1</td>
<td>24</td>
<td>76</td>
<td>101</td>
<td>75.3%</td>
</tr>
<tr>
<td>Scan #2</td>
<td>0</td>
<td>1</td>
<td>13</td>
<td>83</td>
<td>97</td>
<td>85.6%</td>
</tr>
<tr>
<td>Scan #3</td>
<td>0</td>
<td>1</td>
<td>18</td>
<td>70</td>
<td>89</td>
<td>78.7%</td>
</tr>
<tr>
<td>Scan #4</td>
<td>0</td>
<td>1</td>
<td>11</td>
<td>84</td>
<td>96</td>
<td>87.5%</td>
</tr>
<tr>
<td>Scan #5</td>
<td>0</td>
<td>1</td>
<td>19</td>
<td>64</td>
<td>84</td>
<td>76.2%</td>
</tr>
<tr>
<td>Scan #6</td>
<td>1</td>
<td>0</td>
<td>16</td>
<td>78</td>
<td>95</td>
<td>83.2%</td>
</tr>
<tr>
<td>All Scans</td>
<td>1</td>
<td>0</td>
<td>29</td>
<td>119</td>
<td>149</td>
<td>80.5%</td>
</tr>
<tr>
<td>Overall</td>
<td>2</td>
<td>5</td>
<td>130</td>
<td>574</td>
<td>711</td>
<td>81.0%</td>
</tr>
</tbody>
</table>
Table 7-5. Vehicle detection accuracy for Hogtown residential site with different LiDAR scans and overlapped all LiDAR scans.

<table>
<thead>
<tr>
<th>LiDAR</th>
<th>TP</th>
<th>FN</th>
<th>FP</th>
<th>TN</th>
<th>Sum</th>
<th>Accuracy</th>
</tr>
</thead>
<tbody>
<tr>
<td>Scan #1</td>
<td>5</td>
<td>1</td>
<td>9</td>
<td>29</td>
<td>44</td>
<td>77.3%</td>
</tr>
<tr>
<td>Scan #2</td>
<td>5</td>
<td>1</td>
<td>10</td>
<td>43</td>
<td>59</td>
<td>81.4%</td>
</tr>
<tr>
<td>Scan #3</td>
<td>4</td>
<td>2</td>
<td>7</td>
<td>41</td>
<td>54</td>
<td>83.3%</td>
</tr>
<tr>
<td>Scan #4</td>
<td>3</td>
<td>3</td>
<td>12</td>
<td>34</td>
<td>52</td>
<td>71.2%</td>
</tr>
<tr>
<td>Scan #5</td>
<td>3</td>
<td>3</td>
<td>14</td>
<td>48</td>
<td>68</td>
<td>75.0%</td>
</tr>
<tr>
<td>Scan #6</td>
<td>3</td>
<td>3</td>
<td>11</td>
<td>47</td>
<td>64</td>
<td>78.1%</td>
</tr>
<tr>
<td>All Scans</td>
<td>6</td>
<td>0</td>
<td>4</td>
<td>65</td>
<td>75</td>
<td>94.7%</td>
</tr>
<tr>
<td>Overall</td>
<td>29</td>
<td>13</td>
<td>67</td>
<td>307</td>
<td>416</td>
<td>80.8%</td>
</tr>
</tbody>
</table>
Figure 7-1. Object segmentation flowchart.

Figure 7-2. Object segmentation result in Hogtown parking site.

Figure 7-3. Object segmentation result in Hogtown forest site.
Figure 7-4. Object segmentation result in the residential site.

Figure 7-5. Object classification flowchart.

1. Clustered Object Points
2. Principal Component Analysis
3. Spin Image
4. Relative Entropy
5. Bayesian Decision
6. Target Object
7. Clutter Object
Figure 7-6. Using PCA to get the best fitting plane where the first two principal components define vectors that form a basis for the plane and the third principal component is orthogonal to the first two, and it defines the normal vector of the plane.

Figure 7-7. An unoccluded vehicle consisting of 207 LiDAR points represented in the A) 3-D shape and B) 2-D Spin image.

Figure 7-8. An occluded vehicle consisting of 82 LiDAR points represented in the A) 3-D shape and B) 2-D Spin image.
Figure 7-9. An occluded vehicle consisting of 73 LiDAR points represented in the A) 3-D shape and B) 2-D Spin image.

Figure 7-10. An occluded vehicle consisting of 41 LiDAR points represented in the A) 3-D shape and B) 2-D Spin image.

Figure 7-11. An occluded vehicle consisting of 23 LiDAR points represented in the A) 3-D shape and B) 2-D Spin image.

Figure 7-12. An occluded vehicle consisting of 14 LiDAR points represented in the A) 3-D shape and B) 2-D Spin image.
Figure 7-13. An occluded vehicle consisting of 12 LiDAR points represented in the A) 3-D shape and B) 2-D Spin image.

Figure 7-14. An occluded vehicle consisting of 9 LiDAR points represented in the A) 3-D shape and B) 2-D Spin image.

Figure 7-15. An occluded vehicle consisting of 8 LiDAR points represented in the A) 3-D shape and B) 2-D Spin image.

Figure 7-16. An occluded vehicle consisting of 7 LiDAR points represented in the A) 3-D shape and B) 2-D Spin image.
Figure 7-17. Bivariate PDF of reference vehicles

Figure 7-18. Bivariate PDF of reference non-vehicles

Figure 7-19. The information divergence $D_{KL}$ for vehicles and non-vehicles
Figure 7-20. Bivariate PDF of detected vehicles with $BT=0.7$

Figure 7-21. Bivariate PDF of detected non-vehicles with $BT=0.7$

Figure 7-22. Occluded vehicle detection results in the Hogtown parking site from A) overlapped-scanned LiDAR data B) single-scanned LiDAR data, where the white/black lines represent detected vehicles/non-vehicles.
Figure 7-23. Occluded vehicle detection results in the Hogtown forest site from A) overlapped-scanned LiDAR data B) single-scanned LiDAR data, where the white/black lines represent detected vehicles/non-vehicles.

Figure 7-24. Occluded vehicle detection results in the Hogtown residential site from A) overlapped-scanned LiDAR data B) single-scanned LiDAR data, where the white/black lines represent detected vehicles/non-vehicles.

Figure 7-25. Vehicle detection performance, TP, FN, FP, and TN, vs. the number of collected LiDAR points from each object with the Bayesian threshold equal to 0.7.
Figure 7-26. Vehicle detection accuracy, hit ratio and false alarm, vs. the number of collected LiDAR points from each object with the Bayesian threshold equal to 0.7.

Figure 7-27. ROC curves of our detection performance for 0, 5, 10, 15, and 20 sampling points on an object respectively. Those 6 Bayesian thresholds 0.5, 0.6, 0.7, 0.8, 0.9 and 1.5 of each curve are marked in ascending order.
CHAPTER 8
DOWNSAMPLED VEHICLE DETECTION SIMULATION

Vehicle and Clutter Dataset

There are 580 independent vehicles in open area which are extracted from UF campus site. The point density histogram of vehicles is shown as Figure 8-1. The maximum length, width, and height of vehicles are about 6.3m, 2.5m and 2.3m, respectively. An envelope box is defined as a simple classifier whose size is the same as the maximum length, width, and height of vehicles. In order to increase the difficulty of the vehicle detection and balance the total number of vehicles and clutters, the other 580 independent non-vehicle objects are selected by extracting LiDAR points inside the envelope box randomly moving in this study site. The confusion table of this envelope box classifier is in Table 8-1, which shows a simple classifier is unable to give good vehicle detection in this case since its recognition rate and Kappa index of agreement are 50% and 0%, respectively.

Support Vector Machine

The optimal Bayesian classifier is based on the estimation of the PDF functions describing the data distribution in each class. General speaking, it is a difficult task to get a accurate distribution estimation, especially in high-dimensional spaces. Alternatively, one may make the problem easier by designing a decision surface which separates the classes from the training data set such as support vector machine (SVM) [86, 87] and minimum mean square errors (MMSE), without having to deduce it from the PDFs. Although the solution may not correspond to the optimal Bayesian classifier, it usually turns out to result in better performance compared to that of the Bayes classifier which employs estimates of the involved PDFs where the size of the available training data set is limited. The general mathematical formulation of SVMs is briefly recalled as follows.

Linear SVM

Given some training data $D$ and label space $Y$ (e.g., $D = \mathbb{R}^n$, $Y = \{-1, 1\}$ in a two-class problem). The classification is carried out using a linear discriminant function $\omega (D \rightarrow Y)$. Each $x_i \in D$ is a
n-dimensional real vector which is an available training sample with a label $y_i$, where $i \in [1, N]$. The theoretical aim of supervised classification is to find the maximum-margin hyperplane that divides the points having $y_i = 1$ from those having $y_i = -1$. For a linear classifier, $\phi(x) = w \cdot x + w_0$, where $w \in \mathcal{D}$ is the normal vector to the hyperplane and $w_0$ is the bias. We aim at finding the classifier parameters $(w, w_0)$ which verify:

$$\forall (x_i, y_i) \in \mathcal{D}, \quad y_i \times (w \cdot x_i + w_0) > 0$$

(8.1)

Since the SVM method searches the best classifier (i.e., the largest margin), we impose:

$$\forall (x_i, y_i) \in \mathcal{D}, \quad y_i \times (w \cdot x_i + w_0) \geq 1$$

(8.2)

The support vectors lie on two hyperplanes $w \cdot x + w_0 = \pm 1$ which are parallel and equidistant to the optimal linear separable hyperplane. Finally, the optimal hyperplane has to maximize the margin (i.e., the Euclidian distance between both hyperplanes, defined as $2/\|w\|$) under the constraints defined in Equation (8.2). Unfortunately, in most cases, such quadratic optimization problems are unsolvable: we cannot find a linear classifier consistent with the training set. The classification problem is not linearly separable.

Consequently, slack variables or margin errors $\xi_i$, where a slack variable is a nonnegative variable that turns an inequality into an equality constraint, are introduced to cope with misclassified samples and prevent Equation (8.2) from being violated. Another reason is the avoidance of over-fitting the classifier to the training samples, which would result in poor performance. It becomes:

$$\forall (x_i, y_i) \in \mathcal{D}, \quad y_i \times (w \cdot x_i + w_0) > 1 - \xi_i,$$

(8.3)

where $i \in [1, N]$, $\xi_i \geq 0$. The final optimization problem is subsequently:

$$\min \left[ \frac{\|w\|^2}{2} + C \sum_{i=1}^{N} \xi_i \right] \text{ subject to Equation (8.3)}$$

(8.4)

$C$ is a constant which determines the trade-off between margin maximization and training error.
minimization. It turns out that the solution is given as a weighted average of the training points:

$$w = \sum_{i=1}^{N} \lambda_i y_i x_i$$  \hspace{1cm} (8.5)

The coefficients $\lambda_i$ are the Lagrange multipliers of the optimization task and they are zero for all points outside the margin and on the correct side of the classifier. These points therefore do not contribute to the formation of the direction of the classifier. The rest of the points, with nonzero $\lambda_i$’s, which contribute to the buildup of $w$, are called support vectors.

**Multi-Class SVM**

SVMs are designed to solve binary problems. In the cases of having more than two classes of interest, the dominating approach for doing so is to reduce the single multiclass problem into multiple binary classification problems. For such pairwise classification, $n \times (n - 1)/2$ binary classifiers are computed on each pair of classes. Each of the problems yields a binary classifier, which is assumed to produce an output function that gives relatively large values for examples from the positive class and relatively small values for examples belonging to the negative class. Each sample is assigned to the class getting the highest number of votes. A vote for a given class is defined as a classifier assigning the sample to that class.

**Nonlinear SVM**

When the classification problem is not linearly separable, one solution consists in changing the feature space. We can create nonlinear classifiers by applying the kernel trick to maximum-margin hyperplanes. The resulting algorithm is formally similar, except that every dot product is replaced by a nonlinear kernel function $K(x_i, x_j) = \langle \Phi(x_i), \Phi(x_j) \rangle$ where $\langle \cdot, \cdot \rangle$ denotes the inner product operation in a higher dimension space. This allows the algorithm to fit the maximum-margin hyperplane in a transformed feature space. Thus, to solve a linear problem in the high-dimensional space, all we have to do is replace the inner products with the corresponding kernel evaluations. Typical examples of kernel functions are polynomial function and the radial basis function (RBF), defined as
\[ K(x, y) = (x^T y + \beta)^n \]  \hspace{1cm} (8.6)

and

\[ K(x, y) = \exp \left(-\frac{\|x - y\|^2}{\sigma^2}\right), \]  \hspace{1cm} (8.7)

respectively, where \( \beta \) and \( n \) are user-defined parameters in Equation (8.6) and \( \sigma \) is a user-defined parameter that specifies the rate of decay of \( K(x, y) \) toward zero, as \( y \) moves away from \( x \), in Equation (8.7). Note that solving a linear problem in the high-dimensional space is equivalent to solving a nonlinear problem in the original space. As in Equation (8.5), the hyperplane computed by the SVM method in the high-dimensional space is

\[ w = \sum_{i=1}^{N} \lambda_i y_i \Phi(x_i) \]  \hspace{1cm} (8.8)

Given an \( x \), we first map it to \( \Phi(x) \) and then test whether the following is less than or greater than zero:

\[ g(x) \equiv < w, \Phi(x) > + w_0 = \sum_{i=1}^{N} \lambda_i y_i < \Phi(x), \Phi(x_i) > + w_0 = \sum_{i=1}^{N} \lambda_i y_i K(x, x_i) > + w_0 \]  \hspace{1cm} (8.9)

From the previous relation, it becomes clear that the explicit form of the mapping function \( \Phi(\cdot) \) is not required; all we have to know is the kernel function since data appear only in inner products. Observe that the resulting discriminant function, \( g(x) \), is nonlinear because of the nonlinearity of the kernel function. For the kernel function selection, the radial basis function was selected in this study since its detection accuracy is little better than the polynomial function.

**Novel Feature Extraction**

**Spin Image Features**

Two Spin image features \( \beta_\mu \) and \( \alpha_\sigma \) developed in Chapter 7 are used here again. One vehicle
DSM and one clutter DSM with their LiDAR points are showed in Figure 8-2, for illustration. The obtained Spin images for the vehicle and clutter example are shown in Figure 8-3. It can be found that the highest counting positions in each column of the vehicle are similar and less varied than those of the clutter which give the ability to discriminate between vehicles and clutters.

If we only use these two Spin image features $\beta_\mu$ and $\alpha_\sigma$, as the whole feature space in the SVM classifier, the classification result can be obtained as Figure 8-4 and Table 8-2. The recognition rate and Kappa index of agreement by using $\beta_\mu$ and $\alpha_\sigma$ in the SVM are 99.66% and 99.31%, respectively, which are almost perfect and much better than the simple envelope box classifier.

**Principal Component Features**

Another piece of information that is useful to distinguish vehicles from clutter is the blocking LiDAR area. A vehicle is a solid object which cannot be penetrated by LiDAR scanning, so no LiDAR points exist in its underneath area. For a non-solid clutter such as a tree, some ground points could be found inside the area it occupies. The downsampling situation makes this feature more significant since the blocking LiDAR area will not be reduced for a solid object. However, if we only use collected LiDAR points from an object, its occupied area could be reduced a lot in a very sparse point density situation. Therefore, not only object points but also ground points are considered for computing a blocking area. The blocking area by a vehicle tends to be a rectangular shape whose length and width can be estimated by principal component analysis in Chapter 7.

The length and width estimation for the vehicle and clutter example is showed in Figure 8-5. If we only use these two features blocking length and width as the whole feature space in the SVM classifier, the classification result can be obtained as Figure 8-6 and Table 8-3. Its recognition rate and Kappa index of agreement are 91.98% and 83.97%, respectively, which are worse than the spin image features but still much better than the simple envelope box classifier.

**Surface Intensity Feature**

Return intensity or simply intensity is an attribute that describes the strength of the beam
backscatter pertaining to the return in question. It depends on the reflectance properties of the target, and hence it can potentially be used in target discrimination. Its utility for object classification is often reduced because of its dependence on bidirectional reflectance distribution function effects, the distance (range) to the laser instrument, the total number of returns identified in the parent beam, the rank of the return (first, second, etc.) in the parent beam, and the receiver’s gain factor [88]. In this study, we still try to take advantage of intensity information to extract a useful feature. The intensity map is generated by gridding x, y, and intensity data with triangle-based linear interpolation (Figure 8-7). It is observed that intensity of vehicles circled by white lines varied not only between different vehicles but also on individual vehicles. It seems that the LiDAR intensity needs to be normalized before using. So, we also study a paper [89] for how to get LiDAR intensity normalization.

Concerning full-waveform laser data for each single beam, the total number of detected backscattered pulses is known and is assigned to the corresponding echoes. Each echo is described by a point with its 3D coordinate, signal amplitude a, and signal width w at full-width-at-half-maximum (FWHM) derived from the Gaussian approximation (such as Gaussian Mixture Model: GMM). Additionally the 3D coordinate of the sensor position is available. The shape of the received waveform depends on the illuminated surface area, especially on the material, reflectance of the surface and the inclination angle between the surface normal and the laser beam direction. For all points with high planarity, the measured intensity can be normalized by \( I = I_r / \cos(\vartheta) \) where \( I_r \) is recorded intensity and \( \cos(\vartheta) \) is the incidence angle [89].

The normalized intensity can be obtained if 1) the flight position corresponding to each LiDAR point is given. 2) LiDAR points must be located on high planarity to get an accurate normal vector of the reflected surface of a target, which depends on the relative roughness of the surface. When the LiDAR point density is high enough, the normal vector can be calculated accurately even if the size of the flat plane part is small. But, if the LiDAR point density is too low, the normal vector will not be accurate if the size of the flat plane part is not large enough, such as a vehicle.

Instead of normalized intensity, I come up with a novel idea, surface intensity index (SII). First, we
have to find the 3D convex hull from those points belonging to one object \( S_i \), \( i \in [1, n] \). The total point number of \( S_i \) is represented by \( N(S_i) \). It is assumed that those points, \( p_j : j \in [1, N(S_i)] \), of \( S_i \) on the convex hull reflect all measured energy or intensity, \( I_{ij} \), to its surface. The rest points inside the convex hull can only reflect partial measured energy to its surface which is defined by the power propagation rule from wireless communications

\[
I'_{ij} = I_{ij} \left( \frac{d_{ij}}{d_0} \right)^{-n}
\]

(8.10)

where \( d_0 \) is a reference distance, \( d_{ij} \) is the shortest distance from the point \( p_j \) to the convex hull of \( S_i \) and \( n \) is the path loss exponent. In this study, I assume \( d_0 = 0.3 \) m and \( n = 2 \) if \( d_{ij} > d_0 \), otherwise \( I'_{ij} = I_{ij} \). The SII of \( S_i \) is between 0 and 1, which is defined as

\[
\text{SII}_i = \frac{\sum_{j=1}^{N(S_i)} I'_{ij}}{\sum_{j=1}^{N(S_i)} I_{ij}}
\]

(8.11)

The SIIs for the vehicle and clutter example are shown in Figure 8-8. If we use this feature combined with blocking area of an object in the SVM classifier, the classification result can be obtained as Figure 8-9 and Table 8-4. Its recognition rate and Kappa index of agreement are 87.59% and 75.17%, respectively, which are worse than our previous features but still much better than the simple envelope box classifier.

**Vehicle Detection Methods**

**Vehicle Recognition #1 Method**

In [90], the authors used a six-parameter representation that includes the vehicle length, width, and four vehicle parameters (average height values, h1-h4, computed over the four equally size regions) as shown in Figure 8-10. 72 vehicles were chosen and processed in an interactive way, the regions containing vehicles were selected by an operator and the vehicles were automatically extracted by the height threshold method. Their vehicles were parameterized and then categorized into three main groups:
passenger cars, multi-purpose vehicles such as SUVs, minivans, light trucks, and trucks/eighteen-wheelers.

To reduce the dimensionality of the parameter space, PCA was then performed. PCA is an effective tool for handling data representation or classification problems where there is a significant correlation among the parameters describing the object patterns. By training the datasets, the correlation can be determined and a reduced parameter set can be defined that can both represent the information in a more compact way and can support an efficient classification in the reduced feature space. The clear advantage of the method is that it does not require any physical modeling of the data; of course, the selection of the input parameters has some importance. Provided that a rich set of input parameters is defined, the method will effectively identify the redundancy and thus usually results in a quite reduced parameter representation.

In their investigations, the 72 vehicles provided a statistically meaningful dataset for the PCA process. Although the six-parameter representation in our 580 vehicles and 580 clutters do not provide a dataset as meaningful as their investigations, the first two principal components of this feature space still represent about 85% of the information (Figure 8-11).

In order to convey as much information as possible through the PCA process to the classifier, we choose those features which represent at least 99% of the information in the PCA. Therefore, all six principal components are chosen and used in the SVM classifier. Its recognition rate and Kappa index of agreement are 87.59% and 75.17%, respectively, shown in Table 8-5.

**Vehicle Recognition #2 Method**

Applying more features could improve the performance of PCA-based classification. Since the LiDAR dataset also contains intensity values, the authors in [91] use them as additional parameters. They supposed different vehicle categories produce different reflection intensity and extend the six-parameter representation to ten-parameter representation by adding four mean intensity values (int1-int4) corresponding to four mean heights (h1-h4). As opposed to the strikingly positive results of the PCA method which is based on geometric parameters, this enhanced algorithm did not result in well
distinguished categories; the deviation between the intensity values turned out to be too high.

In this study, we still try to use this ten-parameter representation to see its performance. The PCA process reduces its redundancy from 10 features to 4 principal components which can represent over 99% of the information of original features (Figure 8-12). However, using these 4 principal components in the SVM classifier does not result in a satisfactory outcome. Its recognition rate and Kappa index of agreement are only 67.50% and 35.00%, respectively, shown in Table 8-6.

**SVM with SPI (Spin image, PCA, and Intensity) Features**

A Spin image, PCA, and surface Intensity index (SPI) method is proposed here, which is a five-parameter representation including the aforementioned $\beta_{\mu}$, $\alpha_{\sigma}$, blocking length, blocking width, and SII. The PCA process reduces its redundancy from 5 features to 4 principal components which can represent about 99.9% of the information of original features (Figure 8-13). Applying these 4 principal components to the SVM classifier, it results in a very satisfactory performance. Its recognition rate and Kappa index of agreement are only 99.91% and 99.83%, respectively, shown in Table 8-7.

**Downsampled Vehicle Detection Test**

For those 580 unoccluded vehicles, we generate the test datasets by using downsampling rates from 0.1 to 0.9 with an interval of 0.1. Since the other 580 clutters are from diverse objects occluded by trees irregularly, they are not downsampled again in the test datasets. For each downsampling rate, there are 10 simulations in which vehicle points are randomly sampled. The true positive and false negative values are averaged by 10 simulations to get the average recognition rate and Kappa index of agreement in each downsampling rate. Tables 8-8 to 8-10 show the downsampled vehicle detection simulation performances for the vehicle recognition #1, vehicle recognition #2, and SPI methods. Figure 8-14 and Figure 8-15 show the comparison of the envelope box method and the above three methods in average recognition rate and average Kappa index of agreement for the downsampled vehicle detection simulation.

It is obvious by looking at Figure 8-15 that our proposed SPI method is the best one compared to the other three methods, especially in low sampling rates. That is because we try to mitigate the impact
of sparse samples on vehicle detection during the feature extraction stage. Figure 8-15 also verifies that the five features in the SPI method can allow more serious under-sampling situations and tolerate more shape distortion errors than others.

**Testing SVM with SPI Features on Hogtown forest sites**

The above occluded vehicle detection results are obtained from downsampling simulations for vehicles in open areas, a scenario which could be different from the real-world situation. Thus, we apply our proposed SPI method to three forested sites in Hogtown area as well, where the training data are still from Hogtown parking site and Hogtown forest site. The five features extracted from Spin image, PCA, and LiDAR intensity in Hogtown area still can be reduced to four principal components (Figure 8-16) even though the variation percentages of them are different to UF campus area. The occluded vehicle detection results for Hogtown parking site, Hogtown forest site, and Hogtown residential site are shown in Tables 8-11, 8-12 and 8-13, respectively.

Comparing the detection results between the methods based on Bayesian decision with Spin image features and SVM classifier with SPI features by Tables 7-3, 7-4, 7-5, 8-11, 8-12 and 8-13, shows that the latter method improves the performance of the former method. The overall performances of the former one for Hogtown parking site, Hogtown forest site, and Hogtown residential site are 87.9%, 81.0%, and 80.8%, respectively, while they are promoted by the latter method to 97.15%, 99.16%, and 93.03%, respectively. It is noted that the vehicle missing errors in these sites become larger because the SVM classifier focuses on the overall performance. If someone needs to balance the missing error and false alarm of vehicle detection, our Bayesian decision considering with relative entropy values for the vehicle class and non-vehicle class is a good choice since it will give a tradeoff between vehicle and non-vehicle decisions. However, the vehicle missing errors in the SVM classifier with SPI feature can be easily decreased if the number of independent vehicles in occluded situations can be increased.

**Summary**

The proposed SPI method is a new vehicle detection approach which combines five features extracted from Spin image, PCA, and LiDAR intensity and applies them to the SVM classifier. The main
advantage for these features is that they can mitigate the impact of sparse samples and tolerate more shape distortion errors. By using the independent 580 vehicles and 580 non-vehicle objects in the dataset, it is verified that this SPI method outperforms the other three methods for vehicle detection, especially in low sampling rates. In addition, we also apply this method to three forested sites in Hogtown area. It shows that the overall performances for those sites in Bayesian decision are improved significantly where the average accuracy of the three sites is promoted from 83.23% to 96.45%.
Table 8-1. Confusion table of the envelope box classifier

<table>
<thead>
<tr>
<th>Envelope Box</th>
<th>Classified Vehicles</th>
<th>Classified Clutters</th>
</tr>
</thead>
<tbody>
<tr>
<td>Reference Vehicles</td>
<td>580</td>
<td>0</td>
</tr>
<tr>
<td>Reference Clutters</td>
<td>580</td>
<td>0</td>
</tr>
</tbody>
</table>

(Recognition rate, Kappa index of agreement) = (50%, 0%)

Table 8-2. Confusion table of the SVM classification by spin-image features

<table>
<thead>
<tr>
<th>Spin Image (β, α)</th>
<th>Classified Vehicles</th>
<th>Classified Clutters</th>
</tr>
</thead>
<tbody>
<tr>
<td>Reference Vehicles</td>
<td>579</td>
<td>1</td>
</tr>
<tr>
<td>Reference Clutters</td>
<td>3</td>
<td>577</td>
</tr>
</tbody>
</table>

(Recognition rate, Kappa index of agreement) = (99.66%, 99.31%)

Table 8-3. Confusion table of the SVM classification by features (blocking length, blocking width)

<table>
<thead>
<tr>
<th>PCA (Length, Width)</th>
<th>Classified Vehicles</th>
<th>Classified Clutters</th>
</tr>
</thead>
<tbody>
<tr>
<td>Reference Vehicles</td>
<td>555</td>
<td>25</td>
</tr>
<tr>
<td>Reference Clutters</td>
<td>68</td>
<td>512</td>
</tr>
</tbody>
</table>

(Recognition rate, Kappa index of agreement) = (91.98%, 83.97%)

Table 8-4. Confusion table of the SVM classification by features (SII, blocking area)

<table>
<thead>
<tr>
<th>(SII, blocking area)</th>
<th>Classified Vehicles</th>
<th>Classified Clutters</th>
</tr>
</thead>
<tbody>
<tr>
<td>Reference Vehicles</td>
<td>550</td>
<td>30</td>
</tr>
<tr>
<td>Reference Clutters</td>
<td>114</td>
<td>466</td>
</tr>
</tbody>
</table>

(Recognition rate, Kappa index of agreement) = (87.59%, 75.17%)

Table 8-5. Confusion table of the SVM classification by features of the vehicle recognition #1

<table>
<thead>
<tr>
<th>Vehicle Recognition #1</th>
<th>Classified Vehicles</th>
<th>Classified Clutters</th>
</tr>
</thead>
<tbody>
<tr>
<td>Reference Vehicles</td>
<td>572</td>
<td>8</td>
</tr>
<tr>
<td>Reference Clutters</td>
<td>12</td>
<td>568</td>
</tr>
</tbody>
</table>

(Recognition rate, Kappa index of agreement) = (98.28%, 96.55%)

Table 8-6. Confusion table of the SVM classification by features of the vehicle recognition #2

<table>
<thead>
<tr>
<th>Vehicle Recognition #2</th>
<th>Classified Vehicles</th>
<th>Classified Clutters</th>
</tr>
</thead>
<tbody>
<tr>
<td>Reference Vehicles</td>
<td>346</td>
<td>234</td>
</tr>
<tr>
<td>Reference Clutters</td>
<td>143</td>
<td>437</td>
</tr>
</tbody>
</table>

(Recognition rate, Kappa index of agreement) = (67.50%, 35.00%)

Table 8-7. Confusion table of the SVM classification by features of SPI method

<table>
<thead>
<tr>
<th>SPI</th>
<th>Classified Vehicles</th>
<th>Classified Clutters</th>
</tr>
</thead>
<tbody>
<tr>
<td>Reference Vehicles</td>
<td>580</td>
<td>0</td>
</tr>
<tr>
<td>Reference Clutters</td>
<td>1</td>
<td>579</td>
</tr>
</tbody>
</table>

(Recognition rate, Kappa index of agreement) = (99.91%, 99.83%)
Table 8-8. Average vehicle detection performance of Vehicle Recognition #1 method with 10 simulations for each downsample rate

<table>
<thead>
<tr>
<th>Downsample Rate</th>
<th>True Positive</th>
<th>False Negative</th>
<th>Recognition Rate</th>
<th>Kappa Index</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.1</td>
<td>43.80</td>
<td>536.20</td>
<td>52.74%</td>
<td>5.48%</td>
</tr>
<tr>
<td>0.2</td>
<td>248.10</td>
<td>331.90</td>
<td>70.35%</td>
<td>40.71%</td>
</tr>
<tr>
<td>0.3</td>
<td>398.00</td>
<td>182.00</td>
<td>83.28%</td>
<td>66.55%</td>
</tr>
<tr>
<td>0.4</td>
<td>474.60</td>
<td>105.40</td>
<td>89.88%</td>
<td>79.76%</td>
</tr>
<tr>
<td>0.5</td>
<td>511.00</td>
<td>69.00</td>
<td>93.02%</td>
<td>86.03%</td>
</tr>
<tr>
<td>0.6</td>
<td>528.00</td>
<td>52.00</td>
<td>94.48%</td>
<td>88.97%</td>
</tr>
<tr>
<td>0.7</td>
<td>537.70</td>
<td>42.30</td>
<td>95.32%</td>
<td>90.64%</td>
</tr>
<tr>
<td>0.8</td>
<td>545.10</td>
<td>34.90</td>
<td>95.96%</td>
<td>91.91%</td>
</tr>
<tr>
<td>0.9</td>
<td>548.10</td>
<td>31.90</td>
<td>96.22%</td>
<td>92.43%</td>
</tr>
<tr>
<td>1.0</td>
<td>572.00</td>
<td>8.00</td>
<td>98.28%</td>
<td>96.55%</td>
</tr>
</tbody>
</table>

(False Positive, True Negative) = (12, 568)

Table 8-9. Average vehicle detection performance of Vehicle Recognition #2 method with 10 simulations for each downsample rate

<table>
<thead>
<tr>
<th>Downsample Rate</th>
<th>True Positive</th>
<th>False Negative</th>
<th>Recognition Rate</th>
<th>Kappa Index</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.1</td>
<td>232.50</td>
<td>347.50</td>
<td>57.72%</td>
<td>15.43%</td>
</tr>
<tr>
<td>0.2</td>
<td>253.20</td>
<td>326.80</td>
<td>59.50%</td>
<td>19.00%</td>
</tr>
<tr>
<td>0.3</td>
<td>265.60</td>
<td>314.40</td>
<td>60.57%</td>
<td>21.14%</td>
</tr>
<tr>
<td>0.4</td>
<td>277.40</td>
<td>302.60</td>
<td>61.59%</td>
<td>23.17%</td>
</tr>
<tr>
<td>0.5</td>
<td>300.70</td>
<td>279.30</td>
<td>63.59%</td>
<td>27.19%</td>
</tr>
<tr>
<td>0.6</td>
<td>306.10</td>
<td>273.90</td>
<td>64.06%</td>
<td>28.12%</td>
</tr>
<tr>
<td>0.7</td>
<td>315.60</td>
<td>264.40</td>
<td>64.88%</td>
<td>29.76%</td>
</tr>
<tr>
<td>0.8</td>
<td>315.20</td>
<td>264.80</td>
<td>64.84%</td>
<td>29.69%</td>
</tr>
<tr>
<td>0.9</td>
<td>315.90</td>
<td>264.10</td>
<td>64.91%</td>
<td>29.81%</td>
</tr>
<tr>
<td>1.0</td>
<td>346.00</td>
<td>234.00</td>
<td>67.50%</td>
<td>35.00%</td>
</tr>
</tbody>
</table>

(False Positive, True Negative) = (143, 437)

Table 8-10. Average vehicle detection performance of SPI method with 10 simulations for each downsample rate

<table>
<thead>
<tr>
<th>Downsample Rate</th>
<th>True Positive</th>
<th>False Negative</th>
<th>Recognition Rate</th>
<th>Kappa Index</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.1</td>
<td>449.20</td>
<td>130.80</td>
<td>88.64%</td>
<td>77.28%</td>
</tr>
<tr>
<td>0.2</td>
<td>529.40</td>
<td>50.60</td>
<td>95.55%</td>
<td>91.10%</td>
</tr>
<tr>
<td>0.3</td>
<td>557.60</td>
<td>22.40</td>
<td>97.98%</td>
<td>95.97%</td>
</tr>
<tr>
<td>0.4</td>
<td>565.60</td>
<td>14.40</td>
<td>98.67%</td>
<td>97.34%</td>
</tr>
<tr>
<td>0.5</td>
<td>571.20</td>
<td>8.80</td>
<td>99.16%</td>
<td>98.31%</td>
</tr>
<tr>
<td>0.6</td>
<td>572.00</td>
<td>8.00</td>
<td>99.22%</td>
<td>98.45%</td>
</tr>
<tr>
<td>0.7</td>
<td>572.80</td>
<td>7.20</td>
<td>99.29%</td>
<td>98.59%</td>
</tr>
<tr>
<td>0.8</td>
<td>573.30</td>
<td>6.70</td>
<td>99.34%</td>
<td>98.67%</td>
</tr>
<tr>
<td>0.9</td>
<td>573.70</td>
<td>6.30</td>
<td>99.37%</td>
<td>98.74%</td>
</tr>
<tr>
<td>1.0</td>
<td>580.00</td>
<td>0.00</td>
<td>99.91%</td>
<td>99.83%</td>
</tr>
</tbody>
</table>

(False Positive, True Negative) = (1, 579)
Table 8-11. Vehicle detection accuracy for Hogtown parking site with different LiDAR scans and overlapped all LiDAR scans based on SVM with SPI features.

<table>
<thead>
<tr>
<th>LiDAR</th>
<th>TP</th>
<th>FN</th>
<th>FP</th>
<th>TN</th>
<th>Sum</th>
<th>Accuracy</th>
</tr>
</thead>
<tbody>
<tr>
<td>Scan #1</td>
<td>2</td>
<td>6</td>
<td>0</td>
<td>106</td>
<td>114</td>
<td>94.74%</td>
</tr>
<tr>
<td>Scan #2</td>
<td>3</td>
<td>5</td>
<td>0</td>
<td>109</td>
<td>117</td>
<td>95.73%</td>
</tr>
<tr>
<td>Scan #3</td>
<td>3</td>
<td>5</td>
<td>0</td>
<td>103</td>
<td>111</td>
<td>95.50%</td>
</tr>
<tr>
<td>Scan #4</td>
<td>5</td>
<td>3</td>
<td>0</td>
<td>86</td>
<td>94</td>
<td>96.81%</td>
</tr>
<tr>
<td>Scan #5</td>
<td>6</td>
<td>2</td>
<td>0</td>
<td>128</td>
<td>136</td>
<td>98.53%</td>
</tr>
<tr>
<td>Scan #6</td>
<td>5</td>
<td>3</td>
<td>0</td>
<td>109</td>
<td>117</td>
<td>97.44%</td>
</tr>
<tr>
<td>Scan #7</td>
<td>8</td>
<td>0</td>
<td>0</td>
<td>77</td>
<td>85</td>
<td>100.00%</td>
</tr>
<tr>
<td>All Scans</td>
<td>5</td>
<td>3</td>
<td>0</td>
<td>165</td>
<td>173</td>
<td>98.27%</td>
</tr>
<tr>
<td>Overall</td>
<td>37</td>
<td>27</td>
<td>0</td>
<td>883</td>
<td>947</td>
<td>97.15%</td>
</tr>
</tbody>
</table>

Table 8-12. Vehicle detection accuracy for Hogtown forest site with different LiDAR scans and overlapped all LiDAR scans based on SVM with SPI features.

<table>
<thead>
<tr>
<th>LiDAR</th>
<th>TP</th>
<th>FN</th>
<th>FP</th>
<th>TN</th>
<th>Sum</th>
<th>Accuracy</th>
</tr>
</thead>
<tbody>
<tr>
<td>Scan #1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>100</td>
<td>101</td>
<td>99.01%</td>
</tr>
<tr>
<td>Scan #2</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>96</td>
<td>97</td>
<td>98.97%</td>
</tr>
<tr>
<td>Scan #3</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>88</td>
<td>89</td>
<td>98.88%</td>
</tr>
<tr>
<td>Scan #4</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>95</td>
<td>96</td>
<td>100.00%</td>
</tr>
<tr>
<td>Scan #5</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>83</td>
<td>84</td>
<td>98.81%</td>
</tr>
<tr>
<td>Scan #6</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>94</td>
<td>95</td>
<td>98.95%</td>
</tr>
<tr>
<td>All Scans</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>148</td>
<td>149</td>
<td>99.33%</td>
</tr>
<tr>
<td>Overall</td>
<td>1</td>
<td>6</td>
<td>0</td>
<td>704</td>
<td>711</td>
<td>99.16%</td>
</tr>
</tbody>
</table>

Table 8-13. Vehicle detection accuracy for Hogtown residential site with different LiDAR scans and overlapped all LiDAR scans.

<table>
<thead>
<tr>
<th>LiDAR</th>
<th>TP</th>
<th>FN</th>
<th>FP</th>
<th>TN</th>
<th>Sum</th>
<th>Accuracy</th>
</tr>
</thead>
<tbody>
<tr>
<td>Scan #1</td>
<td>2</td>
<td>4</td>
<td>0</td>
<td>38</td>
<td>44</td>
<td>90.91%</td>
</tr>
<tr>
<td>Scan #2</td>
<td>1</td>
<td>5</td>
<td>0</td>
<td>53</td>
<td>59</td>
<td>91.53%</td>
</tr>
<tr>
<td>Scan #3</td>
<td>1</td>
<td>5</td>
<td>0</td>
<td>48</td>
<td>54</td>
<td>90.74%</td>
</tr>
<tr>
<td>Scan #4</td>
<td>2</td>
<td>4</td>
<td>0</td>
<td>46</td>
<td>52</td>
<td>92.31%</td>
</tr>
<tr>
<td>Scan #5</td>
<td>2</td>
<td>4</td>
<td>0</td>
<td>62</td>
<td>68</td>
<td>94.12%</td>
</tr>
<tr>
<td>Scan #6</td>
<td>2</td>
<td>4</td>
<td>1</td>
<td>57</td>
<td>64</td>
<td>92.19%</td>
</tr>
<tr>
<td>All Scans</td>
<td>4</td>
<td>2</td>
<td>0</td>
<td>69</td>
<td>75</td>
<td>97.33%</td>
</tr>
<tr>
<td>Overall</td>
<td>14</td>
<td>28</td>
<td>1</td>
<td>373</td>
<td>416</td>
<td>93.03%</td>
</tr>
</tbody>
</table>
Figure 8-1. LiDAR point density histogram of reference vehicles in the UF campus site, where $\mu = 4.5$ and $\sigma = 0.6$.

Figure 8-2. A DSM and LiDAR point map example for A) a vehicle and B) a clutter.

Figure 8-3. Spin image of A) a vehicle and B) a clutter in Figure 8-2a and 8-2b, respectively.
Figure 8-4. SVM classification result by Spin-image features $(\beta, \alpha)$. 

Figure 8-5. Length and width estimation by PCA for A) a vehicle and B) a clutter in Figure 8-2a and 8-2b, respectively.
Figure 8-6. SVM classification result by blocking length and blocking width features.

Figure 8-7. Intensity Map gridded from LiDAR data (x,y,int) at one of the UF campus site.

Figure 8-8. Point relationship with the convex hull for A) a vehicle and B) a clutter in Figure 8-2, where their surface intensity index are 100% and 94%, respectively.
Figure 8-9. SVM classification result by surface intensity index and blocking area features.

Figure 8-10. The six-parameter representation (W, L, H1, H2, H3, H4) of the vehicle recognition #1 method.
Figure 8-11. Contained information of PCA of the vehicle recognition #1 method.

Figure 8-12. Contained information of PCA of the vehicle recognition #2 method.

Figure 8-13. Contained information of PCA of the novel features from Spin image, PCA, and LiDAR intensity (SPI).
Figure 8-14. Average recognition rate of vehicle detection vs. downsample rate comparison among envelop box, vehicle recognition #1, vehicle recognition #2, and SPI algorithm.

Figure 8-15. Average Kappa index of agreement of vehicle detection vs. downsample rate comparison among envelop box, vehicle recognition #1, vehicle recognition #2, and SPI algorithm.
Figure 8-16. Contained information of principal components of the SPI method applied to Hogtown forest sites.
CHAPTER 9
CONCLUSIONS AND CONTRIBUTIONS

Conclusions

Tree Canopy Removal

ISPRS Working Group III/3 conducted a test and found that all bare ground point extraction filter algorithms perform well on LiDAR point clouds from smooth rural landscapes, but all produce errors in rough terrain with vegetation canopy. We develop this canopy removal algorithm to help detect obscure targets underneath forest canopy as well as mitigate the vegetation problem for those filters.

In this algorithm, the multiple-return characteristic of LiDAR data is analyzed and accordingly laser shots are classified as single-return or multiple-return shots. The major challenge of removing canopy is that some foliage will unexpectedly reflect single-return shots rather than normal multiple-return shots when they are very dense. This challenge can be solved by using our developed algorithms such as analyzing distance relationships between foliage, applying morphological filters to process the canopy/non-canopy image and creating rough digital terrain models to calculate above ground levels of points, etc. The unique feature in this algorithm is that no parameter tweaking is required. Both of the city and forest sites are tested where the data are from ISPRS and UF, respectively. It shows that all tree or forest canopy points have been removed such that all obscure vehicles or buildings underneath canopy can now be easily seen.

Bare-Earth Extraction

A DTM, commonly used interchangeably with a DEM, is a digital representation consisting of terrain elevations for ground positions. A DTM is also called a bare-Earth model since it excludes features on the Earth such as tall vegetation, buildings, and bridges. Thus, a DTM can be used to generate graphics displaying terrain slope, direction of slope, and terrain profiles. As such it is typically applied to flood risk assessment, drainage modeling, multitemporal land use and land cover changes, landslide and mudslide monitoring, as well as urban planning and management.
So far, more prevalent methods of LiDAR filtering can be categorized into three groups: morphological filtering, segmentation modeling, and surface modeling. In this work, we develop a novel bare-earth extraction algorithm consisting of segmentation modeling and surface modeling based on the tree canopy removal algorithm including the morphological filtering. The proposed segmentation modeling is built on a triangulated irregular network and composed of triangle assimilation, edge clustering, and point classification to achieve better discrimination of objects and preserve terrain discontinuities. The surface modeling is proposed to iteratively correct both Type I and Type II errors through estimating roughness of digital surface/terrain models, detecting bridges and sharp ridges, etc.

This proposed method is compared with twelve other filters working on the same fifteen study sites provided by ISPRS. Our average error and kappa index of agreement in the automated process are 4.6% and 84.5%, respectively, which outperform all the other twelve proposed filters. Our kappa index, 84.5%, can be interpreted as almost perfect agreement. In addition, applying this work with optimized parameters further improves performance.

Recently, an unsupervised classification algorithm called Skewness Balancing was developed for object and ground point separation in airborne LiDAR data. Although the main advantages of their algorithm are threshold-freedom and independence from LiDAR data resolution, they have to build a prediction model to categorize LiDAR tiles as hilly or moderate terrains. However, not all LiDAR data can be categorized as either completely hilly or moderate terrain tiles. Once a tile includes both terrain types, their algorithm will face a big challenge. Inspired by their algorithm, we develop a novel slope-based statistical algorithm which is appropriate to any mixed or complicated terrain types. Initially, most objects are removed and initial terrains can be obtained in our object detection algorithm. Slope differences can be assumed to be a zero-mean normal distribution in all kinds of terrains, unlike absolute height information used by the Skewness Balancing algorithm. Based on slope difference variations, the Chi distribution measurement is used to decide the adaptive slope threshold. Accordingly, the adaptive growing height threshold of each pixel is derived by 8-connected neighbor pixels which can be used to iteratively correct classified points in the initial terrain. The testing results show that this algorithm is
even better than our previous algorithms, Chang#1 and Chang#2, which have outperformed all twelve other algorithms working on the same study sites.

**Occluded Vehicle Detection**

Vehicle detection research is devoted to the Intelligent Transportation System (ITS) and Automatic Vehicle Guidance (AVG), but is seldom exploited in the forested terrain. The state-of-the-art airborne LiDAR can provide data in large spatial extents with varying temporal resolution and can be deployed more or less anywhere and at any time which can be potentially applied to forested terrain for military surveillance, homeland security, global warming, disaster rescue, emergency road service, and criminal searching. In this work, under-canopy LiDAR points are obtained and clustered by the canopy removal algorithm, bare-Earth extraction, and morphological image processing. Clustered objects are classified into the vehicle or non-vehicle class by Bayesian decision based on Spin images forming the feature space and information divergences determining the optimal Bayesian threshold. The vehicle detection results are demonstrated, discussed, and verified by Receiver Operating Characteristic curves, where diverse-scanned accuracies of training and testing sites range between 73% and 95%.

In addition, we propose the SPI method, a novel occluded vehicle detection approach, which combines five features extracted from Spin image, PCA, and LiDAR intensity and applies them to the SVM classifier. This SPI method is compared to a simple method and two other vehicle detection methods proposed by other authors’ papers published in ISPRS. With ten simulations in each different downsampling rate testing on independent 580 vehicles and 580 non-vehicle objects, our experiments show that this SPI method outperforms all other methods, especially in low sampling rates.

**Contributions**

Although the vehicle detection has been devoted in the Intelligent Transportation System (ITS), Automatic Vehicle Guidance (AVG), and traffic flow estimation in the LiDAR applications, it has not been exploited in the forested terrain with cluttered environment. In this study, we finished a system whose goal is to detect underneath vehicles in forested terrain from LiDAR point clouds. However, this system can contribute in three aspects.
In the first aspect, our canopy removal algorithm can help 1) detect obscure targets underneath forest canopy and 2) mitigate the vegetation problem for those DTM extraction algorithms. As a matter of fact, the thermal imaging cameras can see the heat signature of people, boats, and vehicles in total darkness as well as through smoke, haze, and light fog, but not through the forest canopy. Whether in a city or a forest site, the vegetation area can be correctly detected and canopy points are successfully removed. All obscure vehicles or buildings underneath tree canopy are revealed as we demonstrated. Accordingly, the occluded rate of forest canopy can be obtained. Furthermore, the detailed x-y distribution of the remaining point density can be found as well which will be very useful for predicting the performance of occluded target detection with respect to various object locations.

In the second aspect, our automatic and robust ground filtering is important in LiDAR applications where classified ground and object points can be used for DTM generation and further reconstruction of topographic features. Many methods have been proposed to extract bare-Earth points from LiDAR data. However, most filters perform well in flat and uncomplicated landscapes, while the landscapes containing steep slopes and discontinuities are still a problem which has not been fully solved. The performance of our point-based bare-Earth extraction algorithm has been compared to twelve proposed filters and evaluated by working on the same fifteen study sites. The average total errors and kappa index of agreement of this work in the automated process are 4.6% and 84.5%, respectively, which outperforms all twelve other filters and such kappa index is interpreted as almost perfect agreement. This algorithm applied with optimized parameters performs even better. In addition, another developed grid-based bare-Earth extraction is a slope-based statistical algorithm which can be adaptive with site-wide and local height variations by the Chi distribution measurement and the derived flexible height threshold. This novel algorithm further improves the ground point extraction performance, with average total errors and kappa index of agreement of 3.4% and 88.6%, respectively.

In the last aspect, we demonstrate that the state-of-the-art airborne LiDAR system can provide valuable data which can effectively support the occluded vehicle detection in forest terrain. Based on the actual number of collected LiDAR points from each object, the average vehicle detection accuracy is
always over 80% by our proposed Bayesian framework with Spin image features, even though there are only less than 5 points reflected from the testing object. This probabilistic-based system performance could be easily promoted if the amount of vehicle sources and the variety of occluded scenarios could be increased in the learning phase. In addition, we also propose SPI method which combines five features extracted from Spin image, PCA, and LiDAR intensity and applies them to the SVM classifier. The main advantage for these features is that they can mitigate the impact of sparse samples and tolerate more shape distortion errors. By using the independent 580 vehicles and 580 non-vehicle objects in the dataset, it is verified that this SPI method outperforms the other three vehicle detection methods, especially in low sampling rates. The potential applications for this work include many fields such as military surveillance, homeland security, global warming, disaster rescue, emergency road service, and criminal searching for those vehicles occluded in forested terrain.
LIST OF REFERENCES


[31] JAMAR Technologies, Inc., Catalog Number 7, Horsham, PA.


BIOGRAPHICAL SKETCH

Li-Der Chang was born November 1971 in Taiwan and received the Bachelor of Science and Master of Science degree in electronic engineering from the Chung Cheng Institute of Technology, Taoyuan, Taiwan, R.O.C., in 1993 and 1999, respectively. In 1993-2006 (except two years while pursuing the Master of Science degree), he was a senior research engineer in the telecommunication research and development department of the communication development office and the scientific research office in Taiwan. Since 2006, he has been working toward the Ph.D. degree in the Adaptive Signal Processing Laboratory of the Electrical and Computer Engineering Department, University of Florida. His research interests include remote sensing application, pattern recognition, algorithm development, target detection, image processing, and adaptive signal processing.