Image Analysis Techniques for LiDAR Point Cloud Segmentation and Surface Estimation

Mahmoud S. Awadallah

Dissertation submitted to the Faculty of the
Virginia Polytechnic Institute and State University
in partial fulfillment of the requirements for the degree of

Doctor of Philosophy
in
Computer Engineering

Lynn Abbott, Chair
Devi Parikh
Ross F. Nelson
Michael S. Hsiao
Randolph H. Wynne
Ahmed M. Ghanem

July 25, 2016
Blacksburg, Virginia

Keywords: LiDAR, active contours (snakes), point clouds, superpixel segmentation.

Image Analysis Techniques for LiDAR Point Cloud Segmentation and Surface Estimation

Mahmoud S. Awadallah

ABSTRACT

Light Detection And Ranging (LiDAR), as well as many other applications and sensors, involve segmenting sparse sets of points (point clouds) for which point density is the only discriminating feature. The segmentation of these point clouds is challenging for several reasons, including the fact that the points are not associated with a regular grid. Moreover, the presence of noise, particularly impulsive noise with varying density, can make it difficult to obtain a good segmentation using traditional techniques, including the algorithms that had been developed to process LiDAR data.

This dissertation introduces novel algorithms and frameworks based on statistical techniques and image analysis in order to segment and extract surfaces from sparse noisy point clouds. We introduce an adaptive method for mapping point clouds onto an image grid followed by a contour detection approach that is based on an enhanced version of region-based Active Contours Without Edges (ACWE). We also proposed a noise reduction method using Bayesian approach and incorporated it, along with other noise reduction approaches, into a joint framework that produces robust results. We combined the aforementioned techniques with a statistical surface refinement method to introduce a novel framework to detect ground and canopy surfaces in micropulse photon-counting LiDAR data. The algorithm is fully automatic and uses no prior elevation or geographic information to extract surfaces.

Moreover, we propose a novel segmentation framework for noisy point clouds in the plane based on a Markov random field (MRF) optimization that we call Point Cloud Density-based Segmentation (PCDS). We also developed a large synthetic dataset of in plane point clouds that includes either a set of randomly placed, sized and oriented primitive objects (circle, rectangle and triangle) or an arbitrary shape that forms a simple approximation for the LiDAR point clouds. The experiment performed on a large number of real LiDAR and synthetic point clouds showed that our proposed frameworks and algorithms outperforms the state-of-the-art algorithms in terms of segmentation accuracy and surface RMSE.

This work received support from NASA award NNX13AR46G and Virginia Tech IGEP program.

Image Analysis Techniques for LiDAR Point Cloud Segmentation and Surface Estimation

Mahmoud S. Awadallah

GENERAL AUDIENCE ABSTRACT

The increasing concerns about the global warming have raised the interest about studying and understanding the global ecosystem components including the carbon cycle. The interaction between forests and earth atmosphere is one major component of the global carbon cycle. Thus, quantifying the global forest biomass is an important factor in studying carbon cycle and its dynamics. Therefore repeated large-scale estimates of forest biomass are critically important.

LIDAR (Light Detection and Ranging) is a active remote sensing method that uses light in the form of a pulsed laser to measure ranges and distances based on the time-of-flight concept (similar to radar systems). LiDAR systems can generate precise, three-dimensional information about the shape of the Earth and its surface characteristics. Therefore, LiDAR remote sensing is much more suitable for forest studies than photogrammetry because of the laser's ability to penetrate tree crowns allowing the system to find ground returns under dense canopies. This property allows us to estimate tree heights which is a major factor for estimating the forest biomass.

In order to track forest biomass changes at the global scale, recurring high-altitude observations are needed. Satellite-based LiDAR systems can provide these observations, although no such systems are currently operational. The situation will change with the launch of NASAs ICESat-2, which is planned for July 2017. However, although LiDAR technology allows for rapid and inexpensive measurements over broad geographical areas, ICESat-2 will be equipped with a new sensor known as photon-counting micropulse LiDAR system. This new LiDAR technology is expected to produce measurements that include high levels of noise. The data produced by this sensor will be in the form of a cloud of points in which the signal points are expected to be much more dense than noise points. Analysis of data from the ICESat-2 satellite will therefore need to be robust with respect to noise, as well as fast and automatic because of the large quantity of data that will be generated.

The problem of segmentation in point clouds is challenging for several reasons, including the fact that the points are not associated with a regular grid, as is the case with most image data. Moreover, the presence of noise particularly impulsive noise with varying density, can make it difficult to obtain a good segmentation using traditional techniques, including the algorithms that had been developed to process LiDAR data. This dissertation introduces novel algorithms and approaches based on statistical techniques and image analysis in order to segment sparse noisy point clouds to extract contours and surfaces in order to detect meaningful measurements and information.

Dedication

Dedicated to my daughters, Mariam and Malak and my son Omar, they spent long times without their dad because “he is working towards his PhD.” To my parents, the blessing of my entire life, to Sara Abdelkader, the spark that fueled my energy along the way and to the interdisciplinary graduate education remote sensing community in Virginia Tech.

Acknowledgments

In the name of Allah, the Entirely Merciful, the Especially Merciful, "And whatever you have of favor - it is from Allah." First of all I want to thank Allah for all the blessings, the good things and the favors I had in this life.

I want also to thank my advisors, Dr. A. Lynn Abbott and Dr. Ross Nelson, for their great inspiration, support and guidance over all this work. Their active thoughts and guidance helped in many milestones of my work. I appreciate very much their supervision which gave me a lot of opportunities to explore my research.

My cordial and special thanks should be given to Dr Randolph Wynne, and the IGEP community from which I gained so much ideas, thoughts and support that affected greatly my progress in every stage of my research.

I am also grateful to all the members of my advisory committee: Dr. Devi Parikh, Dr. Michael S. Hsiao, and Dr. Ahmed M. Ghanem for their valuable time and interest in serving on my supervisory committee.

I also express my appreciation to all the IGEP colleagues for their collaboration and valuable discussions during seminars and meetings.

Finally, I express my great thanks for my lovely daughters, Mariam and Malak for their understanding and patience which made this study possible. Also, I am grateful to my parents for their great support for my life.

Contents

List of Figures

List of Tables

List of Acronyms

ACWE Active Contour Without Edges

ATLAS Advanced Topographic Laser Altimeter System

CMF Continous Maximal Flow

DBSCAN Density-Based Spatial Clustering of Applications with Noise

DEM Digital Elevation Models

DSM Digital Surface Models

DTM Digital Terrain Models

GAC Geometric Active Contours

GLAS Geoscience Laser Altimeter System

G-LiHT NASA-Goddard's LiDAR, Hyperspectral, and Thermal Imager

GVF Gradient Vector Flow

ICESat Ice, Cloud and Land Elevation Satellite

LiDAR Light Detection And Ranging

MABEL Multiple-Altimeter Beam Experiment LiDAR

MPL Micro Pulse LiDAR

RADAR Radiowave Detection And Ranging

ROI Region Of Interest

RSE Residual Standard Error

SEEDS Superpixels Extracted via Energy-Driven Sampling

SERC Smithsonian Environmental Research Center

SLIC Simple Linear Iterative Clustering

SODAR Sound Detection And Ranging

TOC Top Of Canopy

TVD Total Variational Denoising

Chapter 1

Introduction

1.1 Objectives and Motivation

The importance of finding efficient ways of quantifying terrestrial carbon stocks at a global scale has increased due to the concerns about global climate change. Exchange of carbon between forests and the atmosphere is a vital component of the global carbon cycle [1]. Recent advances in remote sensing technology have facilitated rapid and inexpensive measurements of topography over large areas [2]. LiDAR technology is an active remote sensing technology which allows measurements of topography, vegetation canopy heights, and buildings over large areas. Laser altimetry has proven successful for a variety of forested terrain applications, such as the estimation of vegetation height and structure [3], elevation measurements [4], estimation of bare-earth [5], [6], segmentation of individual tree canopies [7], and stand-level forest parameter estimation [8]. Moreover, many studies have been published regarding the estimation of above-ground biomass using airborne and/or satellite analog LiDAR systems [9]–[18]. In order to detect change of forest biomass and its dynamics at the global scale, repeated observations are needed [19]. Satellite-based LiDAR platforms are well suited to the task of gathering such measurements at the global scale, and on a recurring basis [3].

NASA's Ice, Cloud and land Elevation Satellite (ICESat) mission (2003 – 2009) pioneered the use of space-borne laser altimeters to measure changes in elevation of the Greenland and Antarctic ice sheets as well as canopy heights in a global scale as part of NASA's Earth Observing System (EOS) of satellites [20]–[25]. ICESat obtained canopy height estimates using the Geoscience Laser Altimeter System (GLAS) sensor, a laser altimeter for which elevation estimates are based on the analysis of the waveform returns [26]. Canopy height estimates were obtained with root-mean-square error (RMSE) of 2 to 6 m [3].

The success of ICESat motivated NASA to undertake the follow-on ICESat-2 mission, with a launch date currently scheduled in 2017. One objective of ICESat-2 is to measure vegetation canopy height in support of estimating biomass and biomass change on a large scale. A

2-year mission is planned. Taken together, ICESat and ICESat-2 will provide data for an extended change assessment that will span more than 16 years.

ICESat-2 will carry a sensor known as ATLAS (Advanced Topographic Laser Altimeter System) [27], which represents a fundamental change in technology from the GLAS waveform LiDAR system employed by the first ICESat mission. Although both are profiling sensors (as opposed to "pushbroom" sensors), ATLAS is a "photon-counting" device that transmits small bursts of photons, and employs avalanche-mode photodetectors to sense the arrival times for individual photons [28]. ATLAS is designed to obtain elevation measurements with a ground track spacing of less than 2 km using green ($\lambda = 532$ nm) laser.

The main advantage of photon-counting LiDAR is that it uses low power laser which means longer sensor life for space missions. Moreover, data from photon-counting systems have a close affinity to small footprint discrete return data from airborne laser scanners, and these data have been used extensively for biophysical parameter estimation for over 15 years [12], [15], [17], [29]–[33]. The previous work implies improved precision of biomass estimation from spaceborne LiDAR. In fact, a recent metaanalysis indicated that biomass estimated using discrete return systems has Residual Standard Error (RSE) that is comparable or better than biomass estimated using waveform LiDAR, radar, optical data, or combinations of LiDAR and other remotely sensed data (Figure 1.1,[20]). By implication, then, reliable estimates of above-ground forest biomass should be possible using ICESat-2.

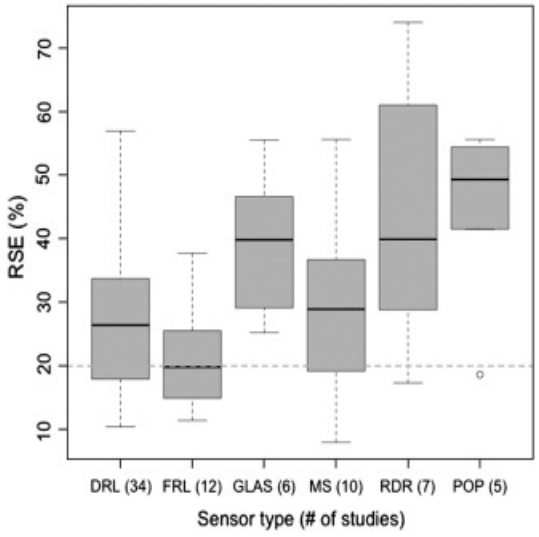


Figure 1.1: Residual standard error from estimates of above-ground biomass compiled from a metaanalysis by Zolkos et al. [20]. RSE(%) (RSE standardized by mean AGB from field measurements) categorized by sensor type, with dotted horizontal line at RSE = 20% of mean aboveground biomass (AGB). DRL = discrete return LiDAR, FRL = full return LiDAR, MS = multi-sensor, RDR = radar, POP = passive optical.

Unfortunately, much of the prior research into the estimation of forest biomass using LiDAR is not directly applicable to ICESat-2, because of the change in technology from the GLAS

analog waveform system to the ATLAS photon counting system. Moreover, a number of applications besides ATLAS sensor produce point clouds in two- or three-dimensional space. In some cases, the *locations* of the individual points represent the only information that is available for segmentation and further analysis. The problem of segmenting such point clouds is challenging for several reasons as illustrated in Figure 1.2 for real LiDAR data and Figure 1.3 for synthetic data:

- The first and most prominent problem is impulsive noise, which results when nonsignal photons are detected by the sensor. Impulsive noise is most prominent during the daylight hours, as photons from the sun are mistaken for signal photons. The noise level will also vary according to surface reflectivity, surface orientation, and local atmospheric conditions. Noise is expected to be present throughout the atmospheric column, within the canopy, and below the ground.
- The second problem is that dense canopies will occlude the ground in some cases, allowing only a small percentage of signal photons to reach the ground and reflect back to the sensor. This problem is related to the use of green laser which is optimized for ice studies and it will be more severe with photon-counting systems, as compared to waveform LiDAR. This effect will be magnified for closed-canopy forest ecosystems.
- A final problem is that gaps in the canopy can be difficult to detect when noise levels are high. Correct detection of these gaps can have a large impact on overall accuracy.

Due to these challenges and the nature of ICESat-2 mission, the analysis of ATLAS data will therefore need to be robust with respect to noise, as well as fast and automatic because of the large quantity of data that will be generated. This research is concerned generally with the segmentation of planar sets of points for which point density is the only discriminating feature. Such segmentation is applicable to a variety of applications. Of particular interest here are the photon-counting LiDAR point clouds. We present novel algorithms and approaches that are based on various statistical analysis and image processing for partitioning these point clouds efficiently and effectively. We also introduce novel algorithms to estimate the ground and canopy surfaces from profiling LiDAR point clouds as a preparation for ICESat-2 mission.

This dissertation is therefore concerned with the fundamental problem of distinguishing foreground (signal) points from background (noise) points within $\mathbb{R}^2$ (2-dimensional Euclidean space), when severe levels of impulsive noise are present. We emphasize that we are not attempting to detect planar shapes within $\mathbb{R}^3$ (3-dimensional Euclidean space), for which regression techniques are well known. We also emphasize that no information is available beyond the coordinates of the points within $\mathbb{R}^2$, such as color or intensity. Furthermore, we do not assume that the points are associated with a regular grid in the plane, such as a pixel array. Therefore, our approach is to perform segmentation with an emphasis on detection of object boundaries.

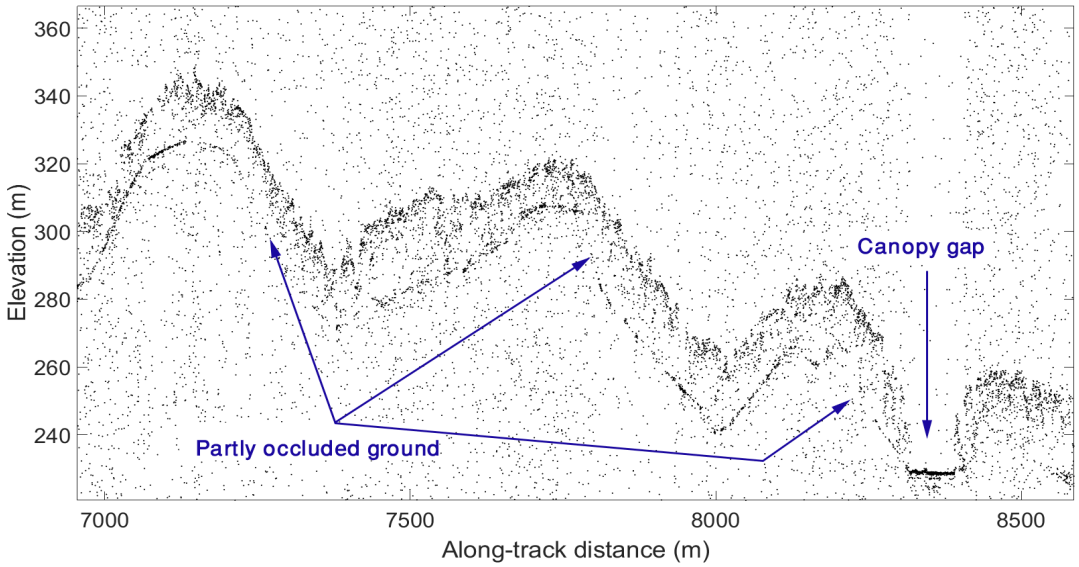


Figure 1.2: A portion of a simulated ATLAS transect, acquired from southwestern Wisconsin, about 5 km from Dodgeville. Impulsive noise is present throughout the data, including locations below ground level. In this example, a road causes a gap in the canopy. Under portions of the canopy that are especially dense, the return signal is weak.

The algorithms introduced in this dissertation were developed and tested on variety of publicly available airborne micropulse laser datasets adopted from three sensors. These include low-altitude photon-counting acquisitions from the Sigma Space Micro Pulse LiDAR (MPL) system [34], high-altitude data from the Multiple Altimeter Beam Experimental LiDAR (MABEL) [35], and the simulated ATLAS data from MABEL (MATLAS) [25]. These sources are consistent with the planned ATLAS sensor on ICESat-2 and they provide insights into the signal qualities that are anticipated for ICESat-2. Estimates of tree heights from these sources are compared with high-density airborne discrete LiDAR data collected for the coincident flight paths using NASA-Goddard's LiDAR, Hyperspectral, and Thermal Imager (G-LiHT).

In addition to LiDAR data, we developed a large dataset of in plane point clouds that includes either a set of randomly placed, sized and oriented primitive objects (circle, rectangle and triangle) or an arbitrary shape that resembles the LiDAR point clouds. We also conducted extensive experiments using these synthetic point clouds.

1.2 Main Contributions

This research presents a novel and adaptive signal-analysis framework for automatic segmentation of sparse, noisy, in plane point clouds. The framework processes these point clouds

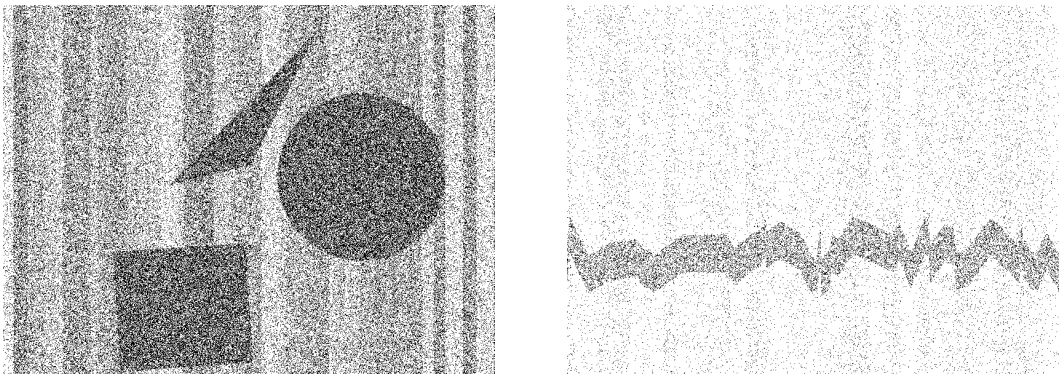


Figure 1.3: Examples of synthetic point clouds in the plane with high levels of impulsive noise representing simple geometric objects and a LiDAR simulation. Segmentation of these cases is difficult with previous techniques.

in both image domain and point cloud domain. In both domains the algorithm uses statistical analysis techniques in order to reduce noise and estimate surfaces. The algorithm also maps these point clouds into a image grid and that enables applying image enhancement and segmentation from an image analysis perspective.

The main contributions of this dissertation can be summarized as follows:

- An adaptive method for mapping point clouds onto an image grid is introduced. In this method the algorithm approximates the point density for signal and noise and decides the size of image grid, along with other algorithm parameters that produce good segmentation results, based on a set of features extracted from the point cloud including the estimated signal-to-noise ratio (SNR).

- A contour detection and segmentation approach for in plane point clouds based on a modified version of region-based Active Contours Without Edges (ACWE) [36]. The ACWE algorithm is based on optimizing an energy function derived from the intensities inside and outside the curve in a grayscale image. In our approach we modified the ACWE energy function in order to incorporate point density into the optimization process.

- One of the major disadvantages for ACWE approach is its sensitivity to noise. The main requirement for a successful segmentation of the point clouds under study in this dissertation is to develop a noise-robust algorithm, so we propose a fully automatic initialization method based on statistical measures of the point density using joint horizontal and vertical projections that initialize the energy of active contours near the Region Of Interest (ROI) in order to enhance the segmentation accuracy, reduce convergence time, and prevent the system from getting trapped by noise and local minima.

- Most optimization techniques (including the one used in ACWE) use a termination criterion (to stop evolution) based on stationary energy state or a predefined maximum number of iterations. In the problem of point cloud segmentation these criteria will fail. Partially because the sparsity of points, which may cause energy function to vanish which means leads the contours to vanish without capturing signal points. The algorithm also might get trapped in an oscillation between two or more states leading to infinite loop of optimization iterations. In this dissertation we propose using a stopping criterion based on density estimation using Delaunay triangulation.

- In this dissertation we also introduce a noise reduction method using Bayesian approach. Based on the assumption that noise and signal point counts are drawn from a Poisson distribution we statistically estimate the Poisson parameter for both signal and noise (λ) and derive a noise reduction algorithm that minimizes the probability of error.

- Due to the amount of noise present in these point clouds, there is no single noise reduction approach that can produce good results in all cases. For this reason we developed a joint noise reduction method that incorporates various noise reduction techniques in one joint framework that produces more robust results.

- In this dissertation we combine the aforementioned techniques to introduce a novel algorithm to detect ground and canopy surfaces in micropulse photon-counting LiDAR data. The algorithm is fully automatic and uses no prior elevation or geographic information to extract surfaces.

- We also propose a novel segmentation approach for noisy point clouds in the plane based on a Markov random field (MRF) optimization that we call Point Cloud Density-based Segmentation (PCDS). The algorithm first performs local grouping of points into over-segmented clusters of nearly uniform density (called superpixels); this initial segmentation is followed by a global segmentation of these groups of points. To our knowledge this is the first superpixel-based segmentation approach that addresses the problem presented here.

- In order to further validate the proposed algorithms, we developed a large synthetic dataset of in plane point clouds that includes either a set of randomly placed, sized and oriented primitive objects (circle, rectangle and triangle) or an arbitrary shape that forms a simple approximation for the LiDAR point clouds.

1.3 Dissertation Organization

This dissertation is organized as follows:

Chapter 2 gives a brief background about the terminologies used in this dissertation including LiDAR and level set active contours.

Chapter 3 introduces the literature review and the study of previous algorithms from surface finding perspective as well as segmentation perspective.

Chapter 4 introduces the datasets and study areas used in this dissertation including the synthetic datasets.

Chapter 5 introduces the proposed framework and algorithms used to estimate forest canopy heights from LiDAR data using the modified ACWE.

Chapter 6 introduces the proposed algorithms for noisy point cloud segmentation using superpixels segmentation and its use to estimate forest canopy heights from LiDAR data.

Chapter 7 introduces the experiments performed and shows the results and comparisons of different proposed algorithms.

Finally, chapter 8 concludes the dissertation and gives a set of ideas to be explored in the future.

Chapter 2

Background

2.1 LiDAR

2.1.1 Overview

Remote sensing can be defined broadly as: the gathering of information at a distance or without direct physical contact with the object being sensed [37]. Recent advances in remote sensing have enabled breakthroughs in many applications such as atmospheric research, environmental research, and military research.

Remote sensing is a complex process that consists of several components, as shown in Figure 2.1. Physical objects (e.g., buildings, vegetation, soil) are the objects under examination. Sensor data are formed when an instrument views a physical object by recording electromagnetic radiation emitted or reflected from it. These data are analyzed and interpreted in order to effectively address practical problems. Finally the applications can combine the remote sensing data with other data to address a specific practical problem such as vegetation cover [37].

Remote sensing can be classified according to the generated radiation source into passive and active remote sensing. In passive remote sensing no self-generated radiation is used in the sensing process while in active remote sensing a self-generated radiation sources are used, such as laser light (LiDAR) or radio-wave (Radiowave Detection And Ranging (RADAR)). Active remote sensing can further be classified according to the frequency of radiation used. Optical remote sensing uses visible frequencies, radio remote sensing uses radio and microwave frequencies, and acoustic remote sensing uses acoustic frequencies [38]

LiDAR (also known as LADAR or laser altimetry) has been studied since the 1960s [39], [40]. It is an active remote sensing technique which is based on transmitting a pulse of laser light into water or air, and the backscattered photons are detected and analyzed to generate range

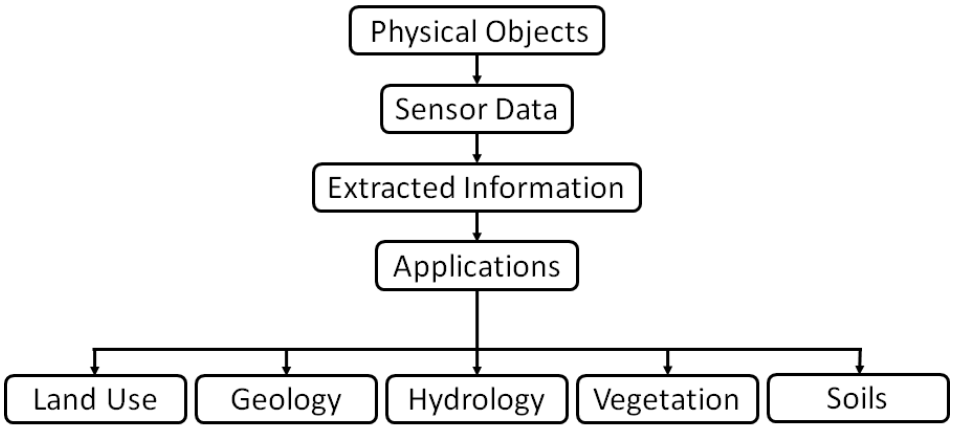


Figure 2.1: Overview of the remote sensing process [37].

information by measuring the time delay between emission and receiving pulses (given that light travels at a known constant velocity). Scattering occurs not only on solid objects but also from the molecules and particulate matter in air and water [37]. The time difference between the emitted laser pulses, the angle of pulse emission, and the absolute location of the sensor, are the three main factors that are used to generate the three-dimensional coordinates of the target objects [41].

LiDAR can provide 10x more accurate elevation values in areas with tree coverage that may cause ground occlusion, as compared to previous techniques [41]. However, photogrammetric Digital Elevation Models (DEM) provide ground points with a higher level of confidence in bare ground areas [41]. In its early stages, LiDAR was used in profiling measurements in which laser aimed directly underneath the aircraft in the nadir direction. The result of each transect is a single elevation profile of the narrow region immediately beneath the aircraft (Figure 2.2) allowing researchers to investigate topography, vegetation structure, and atmospheric studies [37]. More recent systems involve a scanning mechanism (widely called push-broom sensors) providing across-track sampling for the generation of 3D datasets rather than simple 2D profiles [42]. Spaceborne missions (e.g. ICESat and ICESat-2) usually employ profiling LiDAR due to the limited resources in space.

2.1.2 How LiDAR Systems Work

In general, a LiDAR sensor consists of five main components: (1) a transmitter, in all practical cases a laser, but not necessarily pulsed; (2) transmitter optics; (3) receiver optics; (4) a detector; and (5) an electronic system for data acquisition, processing, evaluation, display, and storage (Figure 2.3) [38].

If the speed of light is denoted by c, then the delay t between the transmitted and backscat-

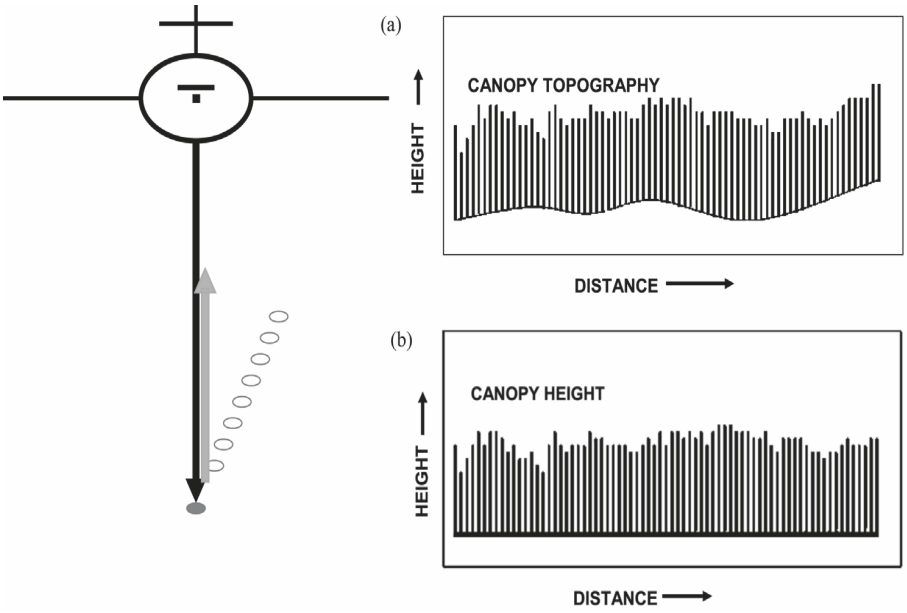


Figure 2.2: Schematic representation of an airborne laser profiler. (a) Acquisition of laser profiles. (b) Sample data gathered by a laser profiler, illustrating extraction of canopy height from the raw profile data [37].

tered pulses from an object at distance x is given by

$$t = 2\frac{x}{c} \tag{2.1}$$

The distance between the instrument and the object can be obtained easily from this equation.

LiDAR sensors for vegetation applications usually operate at near infrared wavelengths (e.g. 1.064 μm) because of two reasons: (1) the backscattering from green leaves will be strong (2), and the atmospheric loss of signal from will be relatively low [42].

LiDAR systems can be classified in several ways: by wavelength (IR, VIS, UV), by technology (discrete return, waveform, photon-counting), by type of platform (terrestrial, mobile, airborne, spaceborne) [38]. The most widely used types of LiDAR systems for vegetation applications are discrete return LiDAR and waveform LiDAR. In waveform LiDAR, the full waveform of return energy per unit of time is digitized. However, in the discrete return LiDAR (maybe single-return or multiple-return systems) packets of photons which comprise bright, significant returns from a single pulse are recorded [37]. Photon-counting LiDAR has a lower energy, but a higher repetition rate, than the analog waveform system. It sends a large number of shots in a short time and records the returns from individual photons with high sensitivity producing a cloud of discrete points [3].

Airborne LiDAR for topographic data collection was developed in the 1980s [43], [44], and commercial airborne LiDAR systems have been operational since the mid-1990s [45]. There

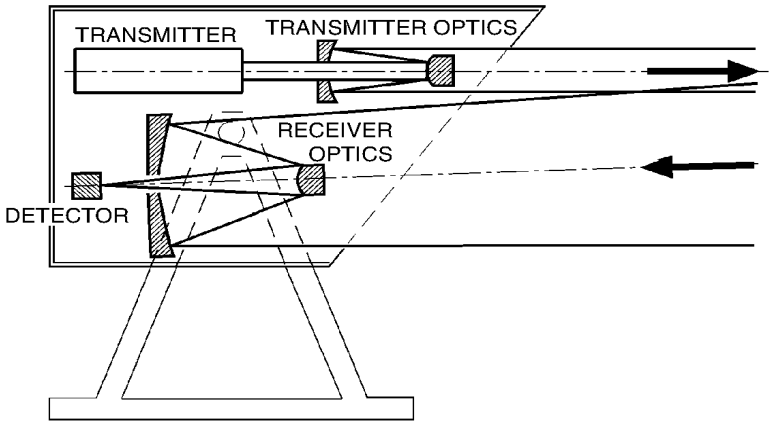


Figure 2.3: Essential optical components of a LiDAR system [38].

has been a significant increase in the use of LiDAR data for DEM generation over the last decade as LiDAR systems have become more reliable, more accurate, and less expensive [46]. Kraus and Pfeifer [47] concluded that the accuracy of LiDAR-derived DEMs for forested areas is comparable to that of DEMs for open areas obtained using photogrammetric techniques. Various filtering methods have been developed to classify or separate raw LiDAR data into ground and non-ground data. However, automated filtering techniques are not perfectly accurate [48], and manual editing of the filtering results is often needed [49], [50].

2.1.3 LiDAR for Global Vegetation Applications

Understanding environmental patterns on the global scale and their changes over long intervals requires examination of movements and interchanges between their varied components. Global carbon cycle which is one of the essential components of living organisms is one example for these patterns. It is of key interest in any consideration of global cycling within the biosphere (Figure 2.4). The atmospheric CO_2 plays a significant rule in the Earth's energy balance. Carbon released by the human activities (e.g., burning fossil fuels for electrical power generation, and to propel automobiles and aircraft) significantly affects the Earth's climate. However photosynthesis plays an important rule in this cycle. LiDARs can provide estimates of the areas covered by plants to assist scientists in understanding the carbon cycle by estimating the time period for which the plants are photosynthetically active.

Mobile and terrestrial LiDAR systems perform a good job in small areas but airborne LiDAR systems expand the geographical range of studies by flying to high altitudes and to remote locations. However, collecting airborne LiDAR data places some constraints (e.g., size, mass, power, and receiver aperture) on the LiDAR system design and performance. LiDAR systems mounted to an aircraft should have the ability to operate in an environment with high- and low-frequency vibrations and temperature and cabin pressure variations [38].

However, airborne LiDAR is not suitable for global scale applications such as climate and carbon cycles because of the large areas required to be covered by the aircraft. These applications require observations from satellite platforms. However, space LiDAR has many design limitations including environmental requirements, surviving the mechanical vibration of launch and operating in the space, and reliability to operate without the need for repair or adjustment [38].

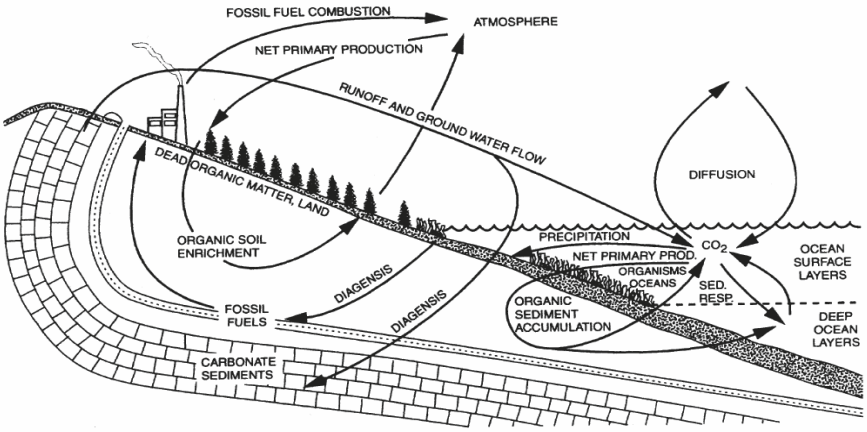


Figure 2.4: Carbon cycle. Adapted from [51].

In effort to shed light on this important cycle, NASA is planning to lunch ICESat-2 as a successor to Ice, Cloud and Land Elevation Satellite (ICESat) mission, which acquired data near-globally during the period from 2003 to 2009.

2.2 The Advanced Topographic Laser Altimeter System (ATLAS) sensor

ATLAS will be the sole instrument aboard ICESat-2. ATLAS is a profiling "photon-counting" space LiDAR sensor that transmits small bursts of photons, and employs avalanche-mode photodetectors (see [28]) to sense the arrival times for individual photons. ATLAS is designed to transmit a pulse every ∼0.7m along ground tracks and will obtain elevation measurements with a ground track spacing of less than 2 km using green ($\lambda = 532$ nm) laser. ATLAS will transmit laser pulses at 10kHz with a footprint diameter of ∼14m. All received noise and signal photons by ATLAS will be projected to the center of each footprint. Each laser pulse contains more than 200 trillion photons, but only very few of them will return to the telescope, where they are sent via optical fibers to the instrument's detectors. Each returned photon will have a geolocated ellipsoidal height. These heights will be corrected for several geophysical phenomena (e.g. atmospheric refraction, tides) and will be classified as likely signal photon or likely background photons. Each laser pulse is split into 6 individual

beams, arranged in three pairs (see Figure 2.5). The beams within each pair have different transmit energies ('weak' and 'strong', with an energy ratio between them of approximately 1:4) and are separated by 90 m in the across-track direction. The beam pairs are separated by 3 km in the across-track direction, and the strong and weak beams are separated by 2.5 km in the along-track direction. ICESat-2 is expected to have RGTs (Reference Ground Tracks) repeat rate of 91 days and it will be able to only characterize the lowest 15 km of the earth's atmosphere [3], [25], [27].

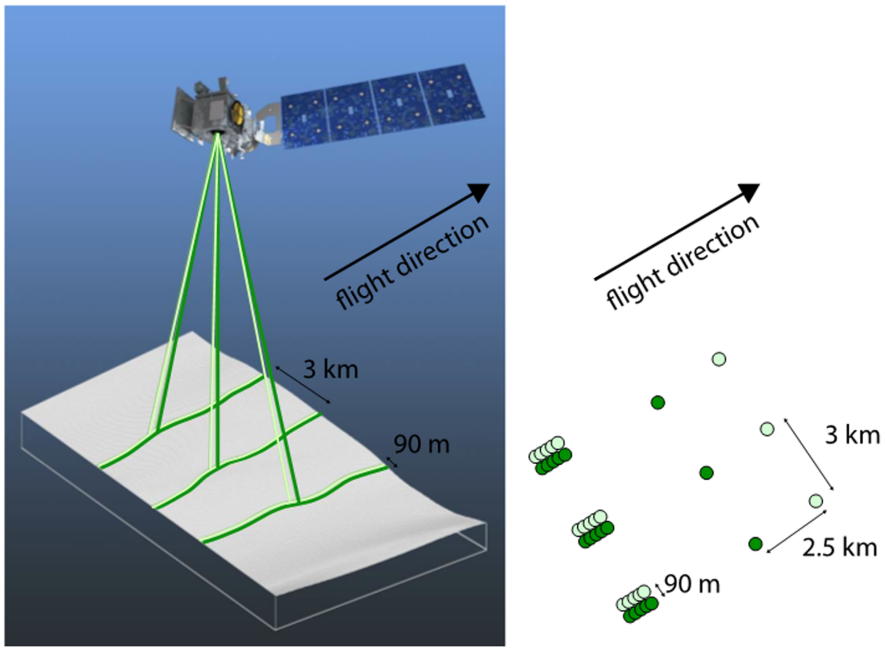


Figure 2.5: A visual representation of the sampling structure for the 6-beams of ATLAS LiDAR as it becomes in orbit [25].

2.3 Active contour models ("snakes")

2.3.1 Introduction

In this thesis we employ a variation of active contour model in order to segment in-plane point clouds. In this section we introduce the basic terminology and concept of this class of segmentation algorithms.

Active contour model is an energy minimizing curve used as a technique to segment objects from a possibly noisy 2D image. It is typically implemented using an energy minimization process [52]. Given an initial curve v in the image, the curve is transformed through repeated

steps that attempt to minimize an energy functional E_{snake}. The name "snake" results from the appearance of the curve v as it changes during the iterative process.

The energy functional is commonly defined using several additive terms, such as the following:

$$E_{snake} = \int_0^1 [E_{int}(v(s)) + E_{image}(v(s)) + E_{con}(v(s))]\, ds \tag{2.2}$$

An internal energy term E_{int} affects the degree to which the contour can bend and stretch. An external energy term E_{image} incorporates effects from image properties, such as magnitudes of intensity edges. A second external energy term E_{con} may be used to specify additional constraints. In this way, a snake represents a compromise between the curve's own properties (such as curvature) and image properties (such as contrast or texture).

In most snakes implementations, the internal curve energy is defined as follows:

$$E_{int} = \alpha(s) \left| \frac{dv(s)}{ds} \right|^2 + \beta(s) \left| \frac{d^2v(s)}{ds^2} \right|^2 \tag{2.3}$$

The first-order term represents the energy due to stretching (elastic energy), and the second-order term measures the energy due to bending (curvature energy). The variables $\alpha(s)$ and $\beta(s)$ are weights that control the significance of each term. The image energy attracts the snake to low-level features in the image. The traditional snakes formulation suggested that lines (E_{line}), edges (E_{edge}) and terminations (E_{term}) could contribute to the energy function, as follows:

$$E_{image} = w_{line}E_{line} + w_{edge}E_{edge} + w_{term}E_{term} \tag{2.4}$$

However, this formulation has limitations when applied to images altered by impulsive noise, or images in which there is no clear intensity edges (such as images of point clouds). In the next subsection we will introduces another more recent formulation of snakes that is not based on image intensity edges.

2.3.2 Active Contours Without Edges (ACWE)

Besides the dependence of traditional snakes model on gradient information which may not be useful in point cloud images, this model also suffers from the drawback of small capturing range (so that the curve should be initialized near to the required area of interest).

Chan-Vese [36] developed a non-parametric active contour model that does not depend on edge information, by using an energy function of the form:

$$F(c_1, c_2, C) = \lambda_1 \int_{inside(C)} |u_0(x,y) - c_1|^2 dxdy + \lambda_2 \int_{outside(C)} |u_0(x,y) - c_2|^2 dxdy \tag{2.5}$$

where c_1 is the mean of pixel intensities inside the curve and c_2 is the mean of pixel intensities outside the curve. Regularity terms including the length of the curve, and/or the area of the region inside the contour are added to the energy function. The contour curve denoted by C, is represented by the zero level-set such that:

$$\begin{cases} C \ = \ \{(x,\ y) \mid \phi(x,y) \ = \ 0\} \\ inside\,(C) = \{(x,\ y) \mid \ \phi\,(\ x,\ y) > \ 0\} \\ outside\ \ (C) \ \ = \ \{(x,\ y) \mid \ \phi\,(\ x,\ y) < \ 0\} \end{cases} \tag{2.6}$$

where $\phi(x, y)$ is the level-set function. The evolution equation is given by:

$$\frac{\partial \phi}{\partial t} = \delta_\epsilon(\phi)[\mu \ \mathrm{div}(\frac{\nabla \phi}{|\nabla \phi|}) - \upsilon - (\lambda_1(u_0 - c_1)^2 + \lambda_2(u_0 - c_2)^2)] = 0 \tag{2.7}$$

where $\upsilon \geq 0,\ \lambda_1 \geq 0,\ \lambda_2 \geq 0$ are constant parameters to be chosen suitably, and $\delta_\epsilon(\phi)$ is the regularized version of Heaviside function H such that (ϵ is a suitably chosen value) $\delta_\epsilon(z) = H^-(z) = \frac{1}{\pi}.\frac{\epsilon}{\epsilon^2+z^2}$. The main advantage of this approach is that the re-initialization step is optional [36], [53], [54]. However, finding appropriate initial contours $\phi_0(x, y)$ is an essential component in obtaining an efficient segmentation [55]. This algorithm is more robust to noise than the original snakes formulations explained in the previous subsection [52] but it still suffers from the essential need for a good initialization and a stopping criterion that makes it suitable for noisy point cloud segmentation.

2.3.3 Superpixel segmentation

Image pixel is an area of illumination from which an image is composed. Many existing computer vision algorithms are based on a computational model that uses image pixels as the basic representation. Markov random fields and other stochastic models of images are often defined on this grid of pixels. However, this grid of pixels is not a natural representation of visual scenes, it is rather, an "artifact" of a digital imaging process [56]. It would be more efficient to apply complex computer vision algorithms with perceptually meaningful entities obtained from a low-level grouping process [56]. This low-level grouping of pixels into much larger entities that has some common characteristics, is called superpixel segmentation. Figure 2.6 shows an illustration of a superpixel segmentation for an image. Such a superpixels has many desired properties:

- It is computationally efficient: it reduces the complexity of images from hundreds of thousands of pixels to only a few hundred superpixels.
- A single superpixel should include pixels from only one object in the image.

- The superpixels boundaries need to conform with the boundaries of objects present in the image.
- The algorithm that produces superpixel segmentation should be computationally efficient because it is actually a preprocessing step to speed up later-stage complex visual processing.
- The superpixels need to be regular in shape in order to facilitate the neighboring superpixels processing.

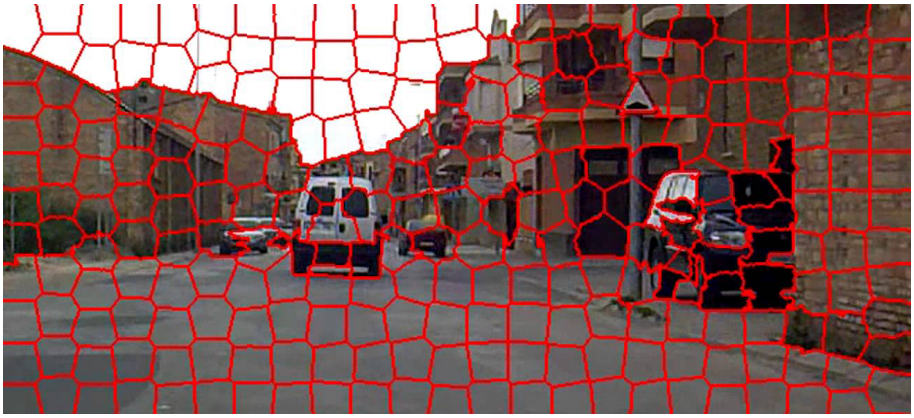


Figure 2.6: An illustration for superpixels obtained for an image [57].

Chapter 3

Literature Review

Due to the interdisciplinary nature of this dissertation, different research areas need to be reviewed. This includes LiDAR data processing and surface detection, image segmentation using active contour models and image segmentation using superpixels. The related work of these areas is introduced in the following subsections.

3.1 LiDAR Data Processing

LiDAR is one of the most effective methods to generate Digital Terrain Models (DTM) and Digital Surface Models (DSM). Many processing algorithms have been developed to generate DTM using raw airborne LiDAR data [5], [40]. An advanced DTM generation algorithm was proposed [58] based on eliminating nonground points (vegetation and buildings). This algorithm is mainly based on the slope of points (sudden change in elevation). After removing the nonground points, the ground points are interpolated to extract a continuous ground curve. The main drawback of this method is that it tends to produce a smoothed version of the DTM. A morphological filtering technique was used to extract bare ground [2]. Filter window size is increased gradually and an elevation difference threshold is used to remove nonground points. The main problem with this approach is that it assumes constant slope which is not true for rough terrains when ground slopes changes rapidly. Another morphological filtering approach is proposed [50] to overcome this drawback. It is based on the fact that the elevation changes suddenly at the boundaries of buildings and other objects above the ground.

Neuenschwander et al. [59] proposed a method for LiDAR data classification into ground, vegetation or man-made features based on a adaptive lower envelope follower (ALEF) which can be used successfully for areas of high topographic relief. The ALEF has problems eliminating all non-ground points, so a post-classification editing process is required in order to have a clean DTM. It also requires user interaction in some cases. The terrain extraction

and segmentation (TEXAS) approach [60] was introduced as an improvement over ALEF. TEXAS uses the third-order spatial derivative to determine the curvature of the terrain rather than relying only on the slope. Another algorithm for DEM generation in urban areas from raw LiDAR data was developed [61]. It is based on the computational results along one-dimensional LiDAR profile using the slope and elevation difference between two consecutive points in that profile.

More recently, terrestrial and mobile LiDAR systems were used [62] to automatically extract road edges using a combination of GVF and balloon [63] parametric active contour models. The algorithm proposed in this work divides the road data into a set of equal length blocks and applies the algorithm on each block separately then combines the results together in the final step. Sweeping 2D laser scanners were used for intelligent vehicles [64] to allow them to recognize and identify its obstacles.

All the previously mentioned algorithms assume that the LiDAR data is obtained by a scanning LiDAR "pushbroom". Unlike ATLAS, scanning LiDAR produces a dense 3D point cloud that represents a whole scanned area with a very small amount of background noise. ATLAS is a space-borne profiling LiDAR which produces sparse, noisy point clouds which represents an elevation profile that may or may not include vegetation. Thus, all the previously mentioned algorithms are not suitable for processing ATLAS data.

Relatively very little research has been performed on space-borne photon-counting LiDAR. Magruder et al. [65] proposed an algorithm to filter noise from photon-counting LiDAR data using Canny edge detection, pdf-based signal extraction, and localized statistical analysis. Herzfeld et al. [3] proposed an approach for detection of ground and canopy cover in Sigma Space MPL data using statistical parameters. However, in this algorithm some empirical constraints and weights were used, which make this algorithm more data dependent and restricts its general applicability for ATLAS data. Horan and Kerekes [66] described a statistical method to reduce noise in photon-counting LiDAR data using elevation bins. Moussavi et al. [67] introduced an automatic multi-step algorithm based on mathematical and statistical data analysis for surface detection in Sigma Space MPL dataset. Zhang et al. [68] proposed a density clustering based algorithm to detect ground from two MABEL examples. Gwenzi and Lefsky [69] introduced a histogram-based approach to detect ground and canopy surfaces from MABEL flights over Savanna ecosystem. Finally, from biomass estimation perspective, Montesano et al. [70] introduced a study of the uncertainty of biomass estimation from ICESat-2 modeled data using the Forest Light (FLIGHT) radiative transfer model for the globally important ecological region of northern Siberia.

All the mentioned previous research is either dedicated to reducing noise without introducing a full surface finding algorithm or is customized to work on a specific dataset without extensive performance analysis of the algorithm for ATLAS data (especially MABEL and MATLAS). From surface finding perspective, to our knowledge, this is the first study for ground and canopy surface estimation using statistical analysis and image processing techniques which is applied to this wide variety of ICESat-2 data. From image segmentation

perspective, this dissertation addresses a unique problem from three aspects. First, and most prominently, the presence of a large amount of noise confuses most segmentation algorithms. Second, the density of signal and/or noise varies significantly within the same point cloud and from one point cloud to another. Third, the proposed algorithm has to be fully automatic.

The previously developed related algorithms that address the same problem ([3], [67]) employ elevation histograms combined with point density analysis in order to extract surfaces. The major drawbacks of these algorithms is the reliance on histograms which is very sensitive to parameters such as bin size. These algorithms also use some data driven empirical parameters that makes algorithms sensitive to the data under study. Moreover, these algorithms have been tested only on one type of simulated ATLAS datasets (MPL).

3.2 Segmentation by Deformable Models

Segmentation by deformable models [71] has been used successfully for many cases involving images, and for some cases involving point clouds that have been mapped onto two-dimensional (2D) image domains [72]. These techniques use internal image forces and external constraints to guide the evolution of continuous contours. Active contours ("snakes") were first introduced by [52], and have been broadly classified into two categories: edge-based and region-based.

Edge-based snakes [52], [73] use gradient information to identify object boundaries in images. This type is quite sensitive to noise, and is heavily dependent on the initialization step due to its small capture range. Gradient Vector Flow (GVF) contour [73] is an example of the gradient-based approaches. GVF snakes have relatively large capturing range but the technique is still sensitive to noise because it is mainly based on image derivatives. In one case [72], this technique was used to segment 2D slices of 3D point clouds, assuming that there was no noise present in the model.

More recently, region-based active contours have been used for image segmentation. These are more robust than edge-based methods to initial contour misplacement and to image noise. Geometric Active Contours (GACs) [74] and Active Contours without Edges (ACWE) [36] are examples of this category. In these models the curve is represented implicitly as zero level-sets of an unknown n-dimensional function. This approach has the advantage that it can handle topological changes automatically. Variational level-set methods are a type of region-based active contour that are capable of producing more robust results because of the ability to incorporate region-based information as well as shape-prior information [75], into energy functional that is directly formulated within the level-set domain. Many other enhancements have been developed for this model to remove the need for re-initialization [53], and for incorporating a local energy term [55]. Active contours were applied using image gradient and GVF to unstructured point clouds [76] to allow curve points to move on an

unstructured mesh rather than the usual mesh-structured points. To solve the initialization problem, Weber et al. [77] proposed an initialization method for active contours which is based on geometric shapes. However, this technique does not accommodate noisy, sparse and arbitrarily-shaped point clouds. ACWE combines curve evolution, the Mumford-Shah functional, and level-set methods to segment an image into a number of disjoint regions that represent foreground objects segmented from the background. ACWE have been widely used in image segmentation [53], [78]–[80], but to our knowledge we are the first to apply a modified ACWE approach to the problem of noisy point cloud segmentation [81], [82].

3.3 Segmentation by Superpixels

Superpixel segmentation approaches can be classified into three broad categories, based on the method used to update superpixels during energy optimization. The first approach finds superpixels by applying graph cuts based on the similarities between neighboring pixels [83]–[85]. The second approach starts with an initial set of small superpixels, and then the size of each superpixel gradually increases based on similarity measures. The most popular examples of this category are watersheds [86], Turbopixels [87] and SLIC [57]. More recently, a third approach has been proposed which starts from a regular grid and gradually changes the superpixels' boundaries to optimize an objective function [88], [89]. The major drawback of the first two approaches is that a large number of pixel similarities need to be calculated in order to grow superpixels from some estimated centers. The third approach is much more computationally efficient. Two examples of this approach are considered in this dissertation: SEEDS [88] and the algorithm of Yao et al. [89]. Both of these use a coarse-to-fine optimization approach to reduce computational complexity, but they differ in the objective function being optimized and in the handling of regularization terms.

Unlike the work presented in this dissertation, all of the algorithms described above assume the presence of intensity or color information for every point in the point cloud. If applied to the problem of interest here, those algorithms typically exhibit poor performance. Examples of superpixel segmentations that result from three algorithms under consideration(Yao et al. [89], SLIC [57], and SEEDS [88]) are shown later in the dissertation. Without color or even monochromatic intensity, the resulting superpixel boundaries tend to "weave" between the points in an effort to optimize the energy function. The resulting superpixel boundaries do not align well with the object boundaries.

The "superpixel" concept easily extends to the case of 3D point clouds. A supervoxel is a collection of 3D voxels, and many of the existing superpixel methods (e.g., [57], [84]) can be extended to 3D. In other work, Papon et al. [90] introduced a connectivity approach for 3D point cloud over-segmentation. This approach assumes the availability of RGB+D data for each point in the point cloud. A constrained planar-cut algorithm was proposed in [91] to partition a 3D point cloud into a set of functional parts using a geometric approach. In [92] the authors proposed a superpixel based segmentation approach to deal with indoor scene

labeling using range data along with color information captured from the Kinect sensor [93]. The research in [94] introduced a region-growing algorithm for surface segmentation of 3D lidar point clouds of urban environments using a coarse-to-fine approach on an octree-based voxelized representation of the input point cloud. In [95] the authors proposed an algorithm to build a Simultaneous Location And Mapping (SLAM) system using the RGB+D data provided by a Kinect sensor. These techniques were considered, but are not a good match to the work presented in this dissertation because they assume the presence of color and/or dense 3D point clouds.

One algorithm that does not rely on color is described in [96]. In that work, the goal is to find topologically persistent holes in a sparse, noisy point cloud in the plane. A major difference between that work and ours lies in the definition of "noise." In [96], noise refers to the deviation of observed points from their true locations in Cartesian space, whereas we are concerned with impulsive (spurious) points. The approach of [96] cannot tolerate any amount of impulsive noise.

Despite the fact that a large number of algorithms have been proposed for segmentation of 3D point clouds, surprisingly few of them have addressed the problem of large amounts of non-uniform noise. Most of the previous 3D segmentation approaches assume that registered color images are also available. Color similarity information is used explicitly in their objective functions in order to find superpixels. In this dissertation we are proposing a superpixel segmentation algorithm for noisy point clouds in the plane. There is no color information associated with points, and the noise density is variable.

We also considered traditional segmentation approaches such as N-cut [83], CMF [97], [98], TVD [99], DBSCAN [100] and revised DBSCAN [101]. Because of our noise characteristics, these techniques also produced very poor results for the point clouds that are of interest here.

The proposed superpixel segmentation algorithm uses a density-based objective function capable of finding good superpixel-type groupings for a noisy point cloud. As compared to previous techniques, we have replaced the appearance/color-similarity terms with another term extracted from some local density measures calculated adaptively from the point cloud. Moreover, we introduce another regularization term that prevents "trivial" superpixels (superpixels that do not contain any point from the point cloud). Our energy function is formulated in such a general way that it can enhance existing superpixel segmentation algorithms as described in the next sections. The results obtained from our algorithm are compared to the original algorithms, and have produced very good results qualitatively and quantitatively in terms of foreground/background classification and conformity to object boundaries (more details are given in Chapter 7). This grouping of points is advantageous for applying any consequent processing for the point clouds (e.g. graph-based approaches) because it reduces the number of nodes required to be considered.

Chapter 4

Datasets and Study Area

In this chapter we introduce the datasets we used in the development and validation stages of the proposed algorithms. Two types of point cloud datasets were used in this dissertation. The first type of dataset is based on real LiDAR datasets provided by NASA [25] for algorithm development and testing. The second type of dataset is a synthetic point clouds with various SNR values. The point clouds represent a sampling of different primitive shapes (circle, triangle, rectangle or arbitrary polygon) formed by a set of points with higher density than noise. These datasets are introduced in more detail in the following subsections.

4.1 ICESat-2 Datasets

NASA has produced variety of publicly available airborne micropulse laser datasets for algorithm development purposes. These datasets were collected using photon-counting LiDAR systems similar to ATLAS. Scientists at NASA applied different simulation parameters to these data in order to incorporate different design cases that simulate photon-counting data expected from ICESat-2 ATLAS sensor. Experiments in this dissertation were performed using three different datasets in several locations and design cases. These include low-altitude photon-counting acquisitions from the Sigma Space Micro Pulse Lidar (MPL) system [34], high-altitude data from the Multiple Altimeter Beam Experimental Lidar (MABEL) [35], and the simulated ATLAS data from MABEL (MATLAS) [25]. This dataset includes transects from diverse locations in USA including profiles for different echo systems, different acquisition times, and multiple simulations to match the expected ATLAS data stream. Figure 4.1 shows the study area of the transects considered in this dissertation. In this figure, the red lines represent the flight lines from MABEL transects, the blue lines represent the MPL transects and the yellow lines represent MATLAS transects.

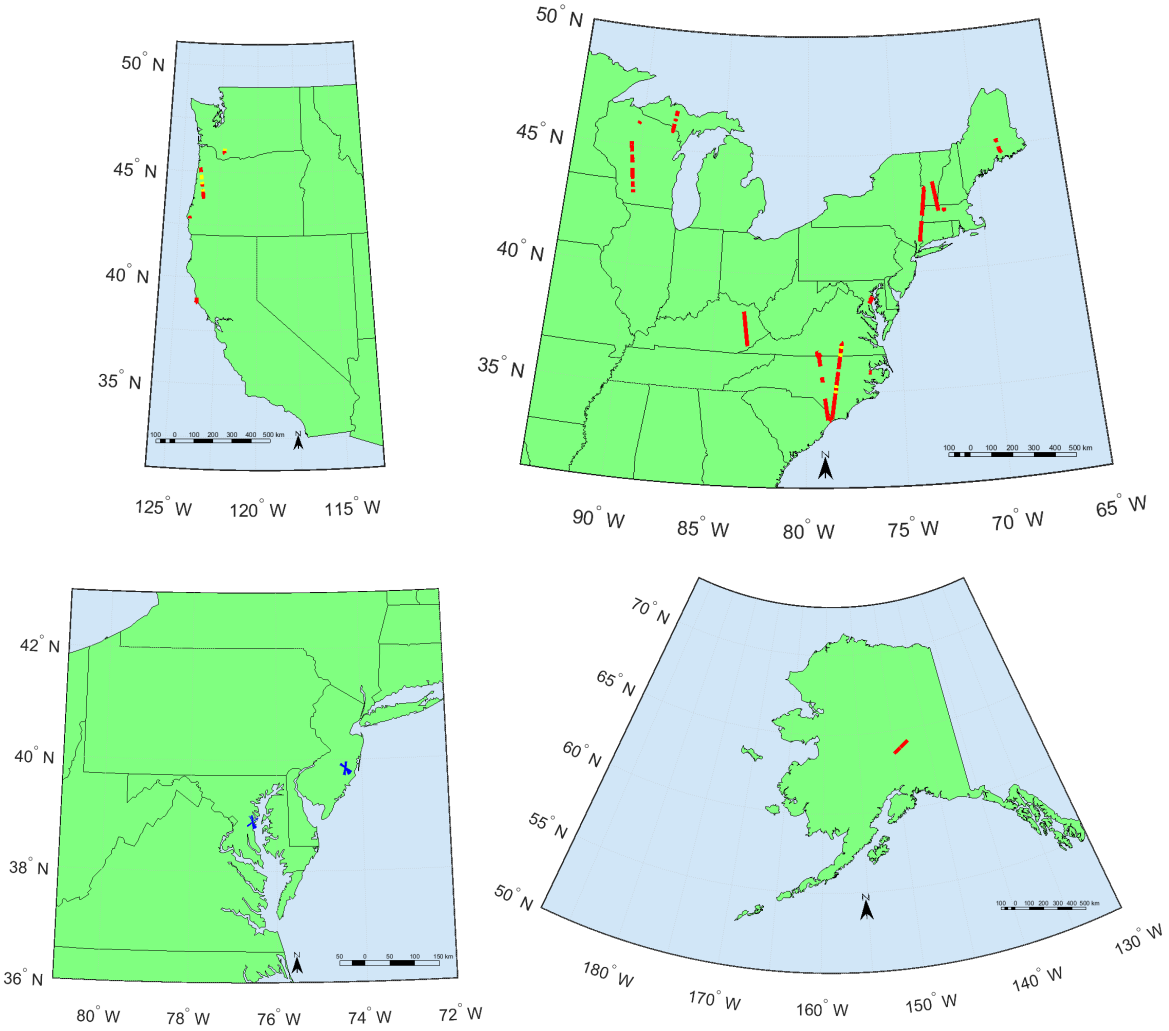


Figure 4.1: Study areas for MABEL (red), MPL (blue), and MATLAS (yellow) transects.

4.1.1 Sigma Space Micro Pulse Lidar (MPL)

The Sigma Space MPL sensor operates at 532 nm wavelength (the same wavelength will be used for ATLAS). This sensor acquired low altitude (∼600m AGL) photon-counting data on October 8, 2009, over two mid-Atlantic, eastern US sites between sunset and 2200 hrs local time. As a result, there was very limited atmospheric noise in the original dataset. To simulate possible ICESat-2 data under varying sunlit atmospheric conditions, scientists at NASA's Goddard Space Flight Center performed reduction, resampling, and simulation to generate multiple noise and beam strengths [34].

Two locations of MPL data were used in the algorithm development. The first was obtained from south of Annapolis, Maryland, USA on the Smithsonian Environmental Research Center (SERC). In this area of gently rolling hills, the dense (>95% canopy closure), tall (25 – 35m) canopy consists mainly of hardwood species, including oak, hickory, maple, and tulip poplar. Given the ±50 – 100m topography and dense overstory, the SERC data represents one of the more challenging types of forest ecosystems in the eastern US especially for ground detection (due to the heavy canopy cover). The second location centers on two flux tower sites, the Silas Little and Cedar Bridge towers, in the Pine Barrens of southern New Jersey, just northeast of Atlantic City, and south of Fort Dix, NJ. The Pine Barrens is a large (∼500km^2), flat, sandy area comprising mainly pine/oak, oak/pine, and pitch pine/scrub oak. The forest canopy closure is roughly 75 – 80%, with evident gaps. From the standpoint of vegetation measurements made using laser altimetry, it is an ideal natural target [34].

This dataset comprises 15 files representing 5 different flight transects (labeled Cedar-2, Cedar-4, SERC-1, SERC-3 and SERC-5), each on the order of 1.5km to 4km in length. For each transect, ambient noise was added at 3 different levels, with the noise varying based on target reflectivity. The files designated as "0.5 MHz" simulate nighttime acquisitions, in which ambient noise is at the lowest level. The other files are designated "2 MHz" and "5 MHz" to represent late daytime acquisitions with clear sky, and daytime acquisitions with a hazy atmosphere, respectively [34].

This dataset is available in ASCII files in which all locations are given in (x, y) UTM coordinates. Each photon is represented in the ASCII file by four variables. *x-value* (m) of footprint location relative to UTM Easting reference, *y-value* (m) of footprint location relative to UTM Northing reference, *z elevation* (m) associated with the photon in the WGS-84 reference system, and *flag* which is 1 for signal photons and 0 for noise photons [34].

The general file naming is *region_flight_uzx.txt*, where *region* represents the name of the region, *flight* designating the flight in the region, and *noise levels* [uz0.5 (lowest), uz2(middle), uz5(highest)] represent the density of random noise added in the simulations to mimic different atmospheric conditions [34]. In order to facilitate the use of MPL datasets in the experiments, we assign a file index for each LiDAR transect to be used in figures. Table 4.1 shows file index, location and noise level for each MPL file.

Table 4.1: MPL file index and their locations and noise levels.

File index	Location	Noise level
1	Cedar-2	0.5 MHz
2	Cedar-2	2 MHz
3	Cedar-2	5 MHz
4	Cedar-4	0.5 MHz
5	Cedar-4	2 MHz
6	Cedar-4	5 MHz
7	SERC-1	0.5 MHz
8	SERC-1	2 MHz
9	SERC-1	5 MHz
10	SERC-3	0.5 MHz
11	SERC-3	2 MHz
12	SERC-3	5 MHz
13	SERC-5	0.5 MHz
14	SERC-5	2 MHz
15	SERC-5	5 MHz

4.1.2 Multiple Altimeter Beam Experimental Lidar (MABEL)

MABEL is a dual-wavelength (532 nm and 1064 nm) high-altitude system that was specifically developed as demonstrator and validation tool for ICESat-2. It was mounted on NASA's ER-2 aircraft which flies over ∼65,000ft (∼20km) and thus is above 95% of the Earth's atmosphere to provide satellite validation and simulation. Moreover, flying at high altitude allows the sensor to collect data over different types of clouds, and allows experimenting with the possible ambiguity that results from separating the laser beam into multiple pulses and lets them all propagate through the atmosphere simultaneously [35].

Two different groups of MABEL datasets were used in this dissertation. The first group represents the MABEL flight lines that were collected during the 20th and 21st of September, 2012. These MABEL data collection campaigns resulted in several files in which every file (usually called a "granule") contains approximately 60 sec of data collection (approximately 12km – 14km length). The second group represents several granules from selected vegetation flights of the MABEL 2014 Alaska campaign.

All MABEL files are stored in the Hierarchical Data Format (HDF) version 5. Its general file naming is *mabel_LLL_YYYMMDDTHHMMSS_RRR_V.h5* where, *LLL* represents the level of the product (i.e: L1A, L1B), *YYYMMDD* represents the year, month, day of requested granule start time, *HHMMSS* represents the hours, minutes, seconds of requested granule start time, *RRR* represents the release number, and *V* represents the version number. A

typical example of a MABEL file name is "mabel_l1a_20101211T004500_010_1.h5" [25]. Figure 4.2 shows an example of one of the MABEL point clouds acquired from southwestern Wisconsin, 5km from Dodgeville. In order to facilitate the use of MABEL datasets in the experiments, we assign a file index for each transect to be used in figures. Table 4.2 shows file index and the actual file name (including acquisition time).

Table 4.2: MABEL 2012 file index and their actual file names along with the latitude and longitude for each file.

File index	File name	Latitude	Longitude	File index	File name	Latitude	Longitude
1	21t231300	34.1836	281.03	26	14t225100	42.9928	286.6618
2	21t231400	34.2996	281.0104	27	14t230100	41.9057	286.4499
3	21t231600	34.5275	280.9716	28	14t230500	41.4695	286.3666
4	21t231700	34.6422	280.9524	29	15t215700	38.8598	283.4998
5	21t233100	36.2289	280.6768	30	20t231600	36.9716	281.817
6	21t233400	36.5714	280.6156	31	20t231800	36.7448	281.7794
7	26t221300	43.0175	269.8525	32	20t231900	36.6314	281.7588
8	26t221500	43.2435	269.8086	33	20t232100	36.4132	281.7195
9	26t221600	43.3565	269.7862	34	20t232200	36.3014	281.6999
10	26t221800	43.5846	269.7409	35	20t232300	36.1912	281.6808
11	26t221900	43.6961	269.7187	36	20t235900	35.447	280.8145
12	26t222000	43.8104	269.6964	37	21t000700	36.35	280.6557
13	26t222100	43.924	269.6732	38	21t225000	35.5434	281.5692
14	26t222400	44.2612	269.6046	39	21t225100	35.4299	281.5495
15	26t222500	44.3744	269.5817	40	21t225200	35.3176	281.5302
16	26t222600	44.4868	269.558	41	21t225300	35.2044	281.5105
17	26t222800	44.7087	269.5128	42	21t225400	35.091	281.4913
18	26t222900	44.8214	269.4897	43	21t225500	34.9782	281.4719
19	26t223000	44.9338	269.4662	44	21t225600	34.8647	281.453
20	26t223100	45.0441	269.4434	45	21t225700	34.7495	281.433
21	26t224900	45.7205	271.8594	46	21t225800	34.6363	281.4138
22	26t225000	45.8498	271.8216	47	21t225900	34.5243	281.395
23	26t225500	46.411	271.941	48	21t230200	34.1888	281.3385
24	21t231500	34.4126	280.9915	49	21t231100	33.9559	281.0685
25	14t224700	43.4224	286.7491	50	21t231200	34.0699	281.049

4.1.3 MATLAS data

The purpose of MATLAS data is to simulate the expected performance of the ATLAS instrument. It is mainly produced from MABEL granules by adjusting the detected number of signal and noise photons to agree with the predicted photon counts using ATLAS instrument model design cases. However, there are some important differences between ATLAS and MABEL sensors. On one hand, ATLAS is expected to transmit green (532nm) laser pulses at 10kHz with a footprint diameter of 14m on the ground. On the other hand, MABEL data collection was operated on a nominal 20km operating altitude which corresponds to a laser footprint on the ground of 2m diameter [35]. MABEL laser pulses are transmitted in the red (532nm) and near infrared (1064nm) wavelengths, and a pulse is emitted every 4cm along the track. The laser at the two wavelengths is split into 16 channels; 8 near infrared (1064nm) and 8 red (532nm) beams.

In order to overcome these differences and produce a dataset that best represents what is expected from ATLAS altimeter, scientists at NASA applied a channel combining mechanism. In this mechanism a group of adjacent MABEL channels with 2m footprints were combined to produce a composite channel to approximate ATLAS's footprint. The photons from several channels were projected to a straight line. This straight line was produced by fitting through all the photons from all channels. The photons were then projected perpendicular to this line. The signal photons are then subsampled to represent the MATLAS 14m footprints spaced 0.7m apart and are assigned the latitude and longitude of the line at the footprint center. The MATLAS data uses file naming conventions and the HDF5 file format similar to the MABEL data Release 10. In order to facilitate the use of MATLAS datasets in the experiments, we assign a file index for each transect to be used in figures. Table 4.3 shows file index and the actual file name (including acquisition time).

4.2 Synthetic Dataset

One of the major drawbacks of the real LiDAR datasets is the unavailability of reliable ground truth. The available ground truth for the real LiDAR datasets is the elevation of ground and canopy surfaces extracted from another higher resolution sensor flown over the same areas. In many cases the difference of the data collection time and the misalignment of both flights causes the ground truth to show surfaces that are different from the "actual" surfaces. A significant part of that misalignment problem is due to the fact that MABEL and MATLAS data sensors had GPS with a relatively low accuracy pointing knowledge. MABEL had pointing knowledge on the order of 30-40 m and GLiHT pointing knowledge is on the order of 2-4 m. Moreover, this ground truth is only suitable for assessing the accuracy of the estimated ground and canopy surfaces which is suitable only for remote sensing applications. In order to assess the algorithm performance from the point cloud segmentation (foreground vs background) perspective, another type of ground truth is required.

Table 4.3: MATLAS file index and their actual file names along with the latitude and longitude for each file.

File index	File name	Latitude	Longitude
1	20t231600_s1_strong	36.9697	-78.1829
2	20t231600_s1_weak	36.9697	-78.1829
3	20t231600_s2_strong	36.9697	-78.1829
4	20t231600_s2_weak	36.9697	-78.1829
5	20t231900_s1_strong	36.6336	-78.2412
6	20t231900_s1_weak	36.6336	-78.2412
7	20t231900_s2_strong	36.6336	-78.2412
8	20t231900_s2_weak	36.6336	-78.2412
9	21t224000_s1_strong	36.6631	-78.235
10	21t224000_s1_weak	36.6631	-78.235
11	21t224000_s2_strong	36.6631	-78.235
12	21t224000_s2_weak	36.6631	-78.235
13	21t225200_s1_strong	35.3184	-78.4697
14	21t225200_s1_weak	35.3184	-78.4697
15	21t225200_s2_strong	35.3184	-78.4697
16	21t225200_s2_weak	35.3184	-78.4697
17	21t225400_s1_strong	35.0914	-78.5086
18	21t225400_s1_weak	35.0914	-78.5086
19	21t225400_s2_strong	35.0914	-78.5086
20	21t225400_s2_weak	35.0914	-78.5086

In order to further assess the accuracy and robustness of the algorithms proposed in this dissertation in terms of segmentation accuracy and robustness for noise, we developed a large set of synthetic point clouds that simulates the characteristics of ATLAS LiDAR point clouds. This dataset consists of two types of point clouds. The primitive shapes point clouds represent a sampling of three different primitive shapes (circle, triangle, rectangle) formed by a set of points distributed randomly with higher density than noise. The point density and noise distribution are randomly chosen based on the available LiDAR point clouds. The location, orientation and size of the shapes are randomly chosen. The "LiDAR-like" point clouds represent a sampling of an arbitrary polygon that was chosen to simulate vegetated areas in LiDAR point clouds. The shape of the polygon is random except that it was restricted to resemble the LiDAR profiles obtained from MABEL. Figure 4.3 shows example point clouds for both types of the synthetic dataset. This dataset along with the MATLAB scripts are publicly available for research purposes.

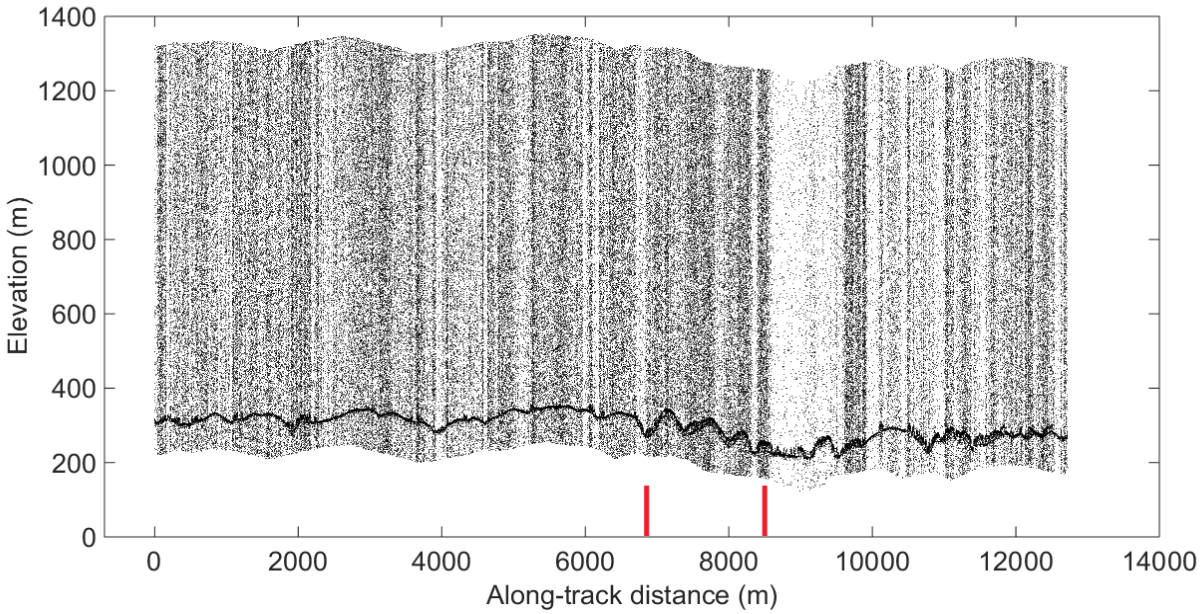
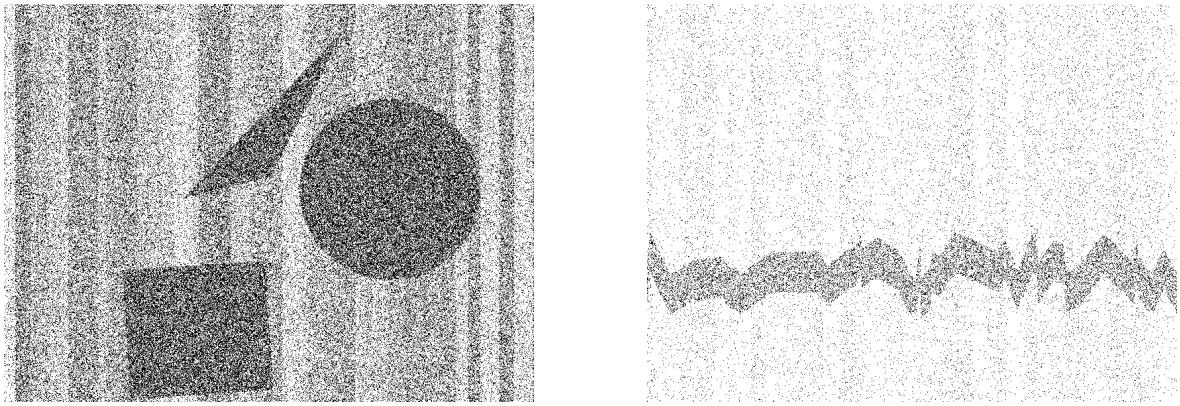


Figure 4.2: An example of one of the MABEL transects acquired from south western Wisconsin, 5km from Dodgeville. The two red bars indicate the zoomed area shown in Figure 1.2.



(a) **(b)**

Figure 4.3: Examples of the synthetic point clouds with (a) primitive shapes and (b) "LiDAR-like" shape

Chapter 5

Estimating Forest Canopy Heights from LiDAR Point Clouds Using ACWE

5.1 Overview

This chapter introduces an algorithm that automatically estimates both top of canopy and ground surfaces from a profiling spaceborne photon-counting LiDAR point clouds without using prior information related to elevation. The algorithm starts by adaptively mapping the LiDAR point cloud onto a discrete two-dimensional grid. Each cell (pixel) in that grid (image) contains a count of the number of LiDAR returns. The algorithm next performs noise reduction using a novel approach that combines Bayesian denoising with horizontal projection. Horizontal projection performs elevation histogramming in the image domain, which is much more efficient than traditional histogram methods. The algorithm then employs Delaunay triangulation combined with a modified level set model in order to make initial estimates of surface contours for both ground and top of canopy. The algorithm finally uses locally weighted scatterplot smoothing (LOWESS) regression method in order to identify vegetated areas and distinguish them from bare ground areas. Based on this distinction between vegetation and bare ground, the algorithm refines the initial surface estimates using LOWESS regression and curve fitting methods. Finally the algorithm smooths the ground curve using a spline fit technique. To our knowledge, this is the first fully automatic algorithm for detecting canopy height for ATLAS sensor that has been tested on all simulated ATLAS data without using prior elevation information. The framework of the algorithm is summarized in Figure 5.1.

Previously developed related algorithms [3], [67] employ elevation histograms combined with point density analysis in order to extract surfaces. The major drawback of these algorithms is

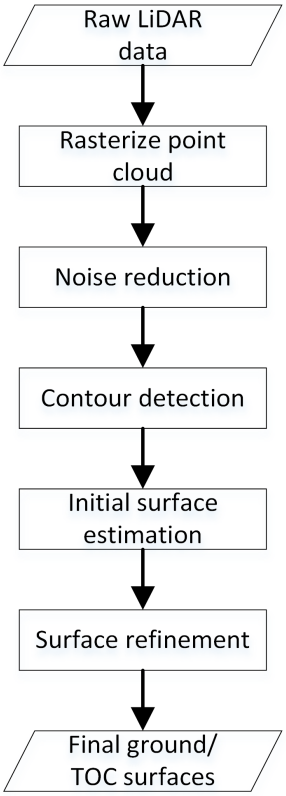


Figure 5.1: Overview of system. The goal is to detect curves that correspond to the ground and to the top of canopy.

the full reliance on histograms, which is very sensitive to parameters such as bin size. These algorithms also use some data driven empirical parameters that make algorithms sensitive to the data under study. Moreover, these algorithms have been tested only on one type of simulated ATLAS dataset (MPL).

Each of the following subsections explains the steps of the algorithm in detail. Figure 5.2 shows a detailed flowchart for the algorithm.

5.2 Mapping point clouds onto a 2D grid

The first stage of the algorithm introduces a 2D quantization grid that encodes the spatial locations of points. Given a point cloud $D = \{p_i \mid i = 1, \ldots, N\}$, $p_i \in \mathbb{R}^2$, in the plane defined by the normal vector $\overrightarrow{n}$, the aim of this step is to map D into a 2D quantization plane I. This plane is subdivided into a grid in which every grid cell $I_{i,j}$ represents a pixel at row i and column j, where $I_{i,j} = |\{P_w\}|$ is the number of points projected inside a

window w of size $h \times h$. The window w is moved through the point cloud perpendicular to $\overrightarrow{n}$ in both dimensions (x, y) with step size t (normally $t = h$). A 2D map therefore results, containing values that represent the number of points in each window. The generated grid has the size (n, m),

$$n = \left(\frac{\max(x) - \min(x) - h}{t}\right) + 1, \quad m = \left(\frac{\max(y) - \min(y) - h}{t}\right) + 1 \tag{5.1}$$

where $\max(x)$, $\max(y)$, $\min(x)$, $\min(y)$ are the maximum and the minimum values of x and y coordinates in the given point cloud.

The quantization of points into this planar grid has two advantages. First, with this grid, virtually any image processing operation can be performed directly over the point clouds. Second, this grid preserves the spatial relations between points, which adds a computational efficiency of this projection plane as a data structure compared to other point cloud representation data structures such as the k-d tree. For example, the average complexity to find the nearest neighbor to a point is $O(log\ N)$ in k-d tree, where N is the number of points. However, using this gridded data structure the complexity is $O(1)$ in almost all cases.

It is also obvious that there is a tradeoff between the window size h and the sparsity of points in the gridded plane. As h decreases, the noise density decreases relative to the image I, but the signal point density also decreases, causing segmentation algorithms to fail in capturing the signal points because the points become so sparse. On the other hand, as h increases, the density of points (both noise and signal) increases, causing the segmentation algorithms to be confused by high-density groups of noise points. It is clear that a good method is required to select an effective value for h that facilitates the segmentation process. In section 5.5.5 we introduce an adaptive method to calculate the image size (values of m and n) that produce better segmentation results. Moreover, the sensitivity of our algorithm for this parameter is analyzed in the experiments section.

5.3 Altitude and Signal-to-Noise Ratio (SNR) estimation

After mapping the point cloud into a 2D grid, the algorithm needs to estimate two important features. The first feature is the altitude of the sensor that collected the point cloud. This feature can be estimated using the difference between the highest and the lowest observations in a given transect. If the point cloud was collected at low altitude (e.g., MPL dataset) the algorithm needs to adjust its parameters such that the noise reduction mechanism should be more conservative because the canopy portion of the point cloud appears sparser in the 2D image. If the point cloud was collected at high altitude (e.g., MABEL and MATLAS datasets) the canopy will appear as a small dense portion of the point cloud so the noise reduction mechanism need to be more sophisticated.

The second feature extracted from the point cloud is the signal-to-noise ratio (SNR) in order to select the consequent steps which produce the best surface estimate. The SNR can be estimated using a slice-based horizontal projection, which is an image processing technique equivalent to the binning elevation histogram (see section 5.4.1 for more details). If the SNR is high, the initial ground and TOC surfaces can be estimated using an edge detector called Combination of Receptive Fields (CORF) [102] because the strong signal will produce a strong edge that can be detected with the CORF edge detection. Otherwise, a modified version of ACWE is used to estimate initial surfaces.

5.4 Noise reduction

The amount of noise introduced in photon-counting point clouds is one of the major challenges for any surface finding algorithm. As a result, noise reduction is considered one of the vital components in our algorithm. In this algorithm we did not use any global DTM products to "clip" unwanted noise because any misalignment between the global DTM and ICESat-2 product will cause poor results. In this algorithm we propose a novel framework for noise reduction that is based on combining different heterogeneous techniques in a successful collaborative framework as depicted in the following subsections.

5.4.1 Noise reduction using slice-based horizontal projections

This technique is based primarily on the assumption that the point density for the signal return is more than the density of noise points. It was first employed in [81] as a technique for level set initialization. This method is an image processing implementation of the elevation histogram, but it offers more flexibility and efficiency than direct histogramming using k-d tree. In this method we first divide the image into a set of vertical slices and calculate the horizontal projection for each slice. The horizontal projection is the summation of each row. This summation gives a count of the number of points in each row (rows represent elevation in this case).

These horizontal projections are then processed to find the rows in each slice with high gradient (1D edges) as they indicate the rows at which there is a large change in point density which leads, roughly, to the locations of the signal photons within the slice. The gradient direction is then calculated for each of these locations, and the area between every two successive increasing and decreasing edges is considered to be most likely to include the signal. These areas are kept and all other areas are removed. The drawback of this method is that it may not be able to detect signal in high slope areas. Another drawback is that sometimes it might mis-classify the ground points under dense canopy as noise because these points usually have lower density. Figure 5.3 shows an example of the result of this technique applied on the transect shown in Figure 4.2

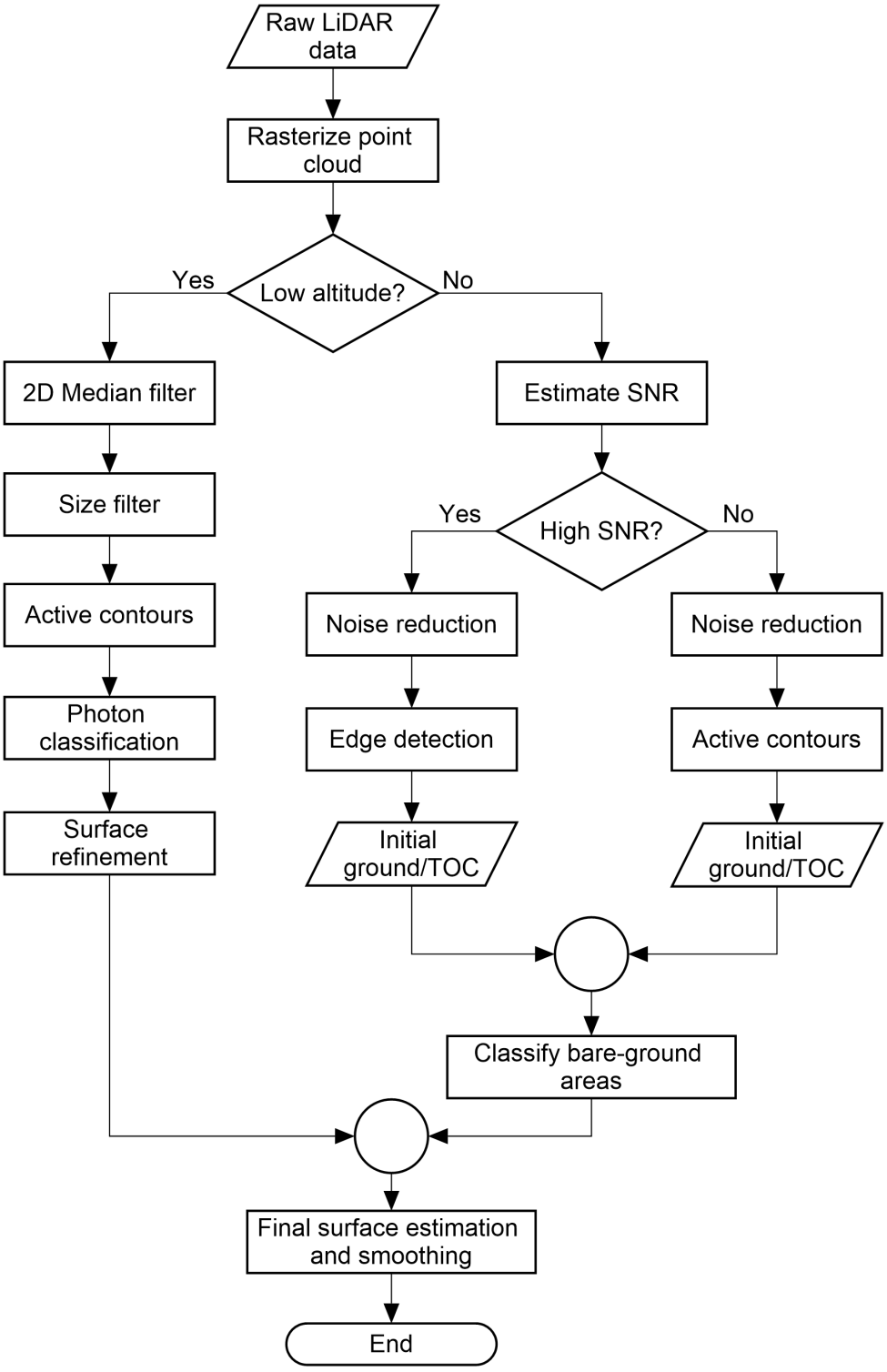


Figure 5.2: Flowchart for the proposed algorithm. The output is two curves that correspond to the ground and the top of canopy.

Mathematically this can be expressed as follows. Let $H_i(x) = \sum_{y=a}^{b} I(x,y)$ represent the horizontal projection of the vertical slice i which extends from column a to column b of the image $I(x,y)$. Let the set $\theta_H^i = \{[e^+, e^-] \mid e^+ = \nabla H^+ \text{ and } e^- = \nabla H^-\}$ represent all intervals $[e^+, e^-]$ between each two successive positive edge (∇H^+) with its corresponding negative (∇H^-) edge in horizontal slice i. Then the initial arbitrary region Ω_0 defined in the image Ω is defined as:

$$\Omega_0(x,y) = \begin{cases} 1, & \text{if } x \in \bigcup_i \theta_H^i \\ 0, & \text{otherwise} \end{cases} \tag{5.2}$$

The region Ω_0 is used to initialize the level set by the initial signed distance function as follows:

$$\phi_0(x,y) = \begin{cases} \rho, & \text{if } (x,y) \in \Omega_0 \\ -\rho, & \text{otherwise} \end{cases} \tag{5.3}$$

5.4.2 Bayesian noise reduction method

We propose another novel method to reduce noise from noisy point clouds using Bayesian decision theory. This method assumes that the noise and signal densities follow two different Poisson distributions with parameters λ_1 and λ_2 (see Figure 5.4). The key idea is to find the decision boundary between the two distributions that minimizes the probability of error. This can be achieved using the Bayesian decision theory, where:

$$P(X = k) = \frac{\lambda^k e^{-\lambda}}{k!} \tag{5.4}$$

where k is the number of points observed in a specific window w, and λ is the average number of points in the given window. The problem of noise removal in this context can be characterized as a binary classification problem assuming that class ω_1 represents noise and class ω_2 represents signal. The Bayesian decision theory can be used to find a threshold that minimizes the probability of error as follows:

$$P(\omega_i|k) = \frac{P(k|\omega_i)P(\omega_i)}{P(k)}, i = 1, 2 \tag{5.5}$$

$$\frac{P(k|\omega_2)P(\omega_2)}{P(k)} \underset{\omega_2}{\overset{\omega_1}{\lessgtr}} \frac{P(k|\omega_1)P(\omega_1)}{P(k)} \tag{5.6}$$

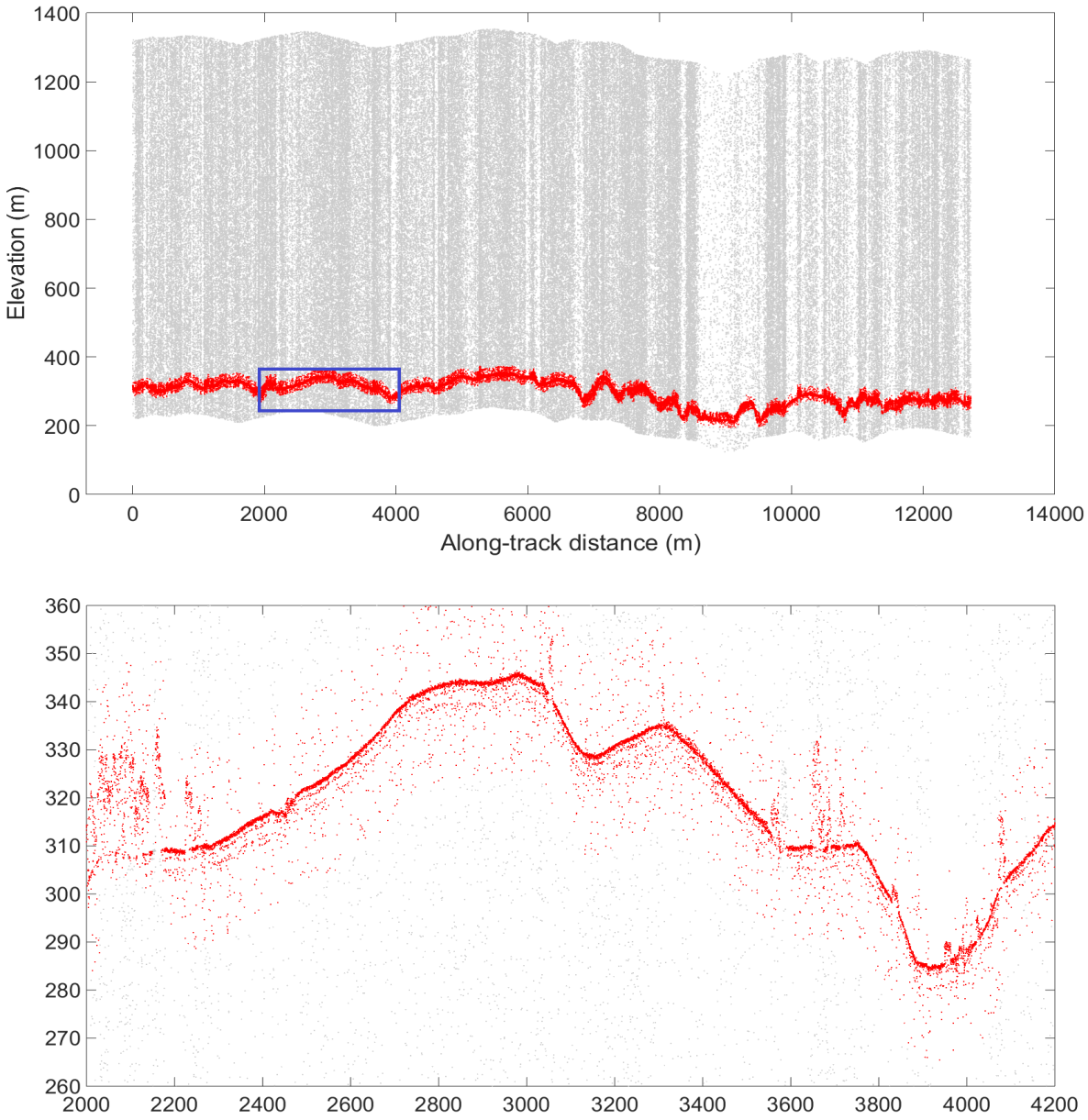


Figure 5.3: An example result of horizontal projection noise reduction applied on the image shown in Figure 4.2. Red points indicate signal photons and gray points indicate noise photons. A close up for the blue rectangle is shown at the bottom.

Assuming that the priors satisfy $P(\omega_i) = 0.5$, the need for $P(\omega_i)$ can be removed. Removing $P(k)$ from both sides simplifies the problem into:

$$P(k|\omega_2) \underset{\omega_2}{\overset{\omega_1}{\lessgtr}} P(k|\omega_1) \tag{5.7}$$

Substituting 5.4 in 5.7 we get:

$$\frac{\lambda_2^k e^{-\lambda_2}}{k!} \underset{\omega_2}{\overset{\omega_1}{\lessgtr}} \frac{\lambda_1^k e^{-\lambda_1}}{k!} \tag{5.8}$$

$$\lambda_2^k e^{-\lambda_2} \underset{\omega_2}{\overset{\omega_1}{\lessgtr}} \lambda_1^k e^{-\lambda_1} \tag{5.9}$$

From this expression a likelihood ratio l can be extracted as follows:

$$l = \frac{\lambda_2^k e^{-\lambda_2}}{\lambda_1^k e^{-\lambda_1}} \tag{5.10}$$

where λ_1 and λ_2 represent the average point density for classes ω_1 and ω_2 respectively. The classification decision ω is taken as follows:

$$\omega = \begin{cases} \omega_1, & \text{if likelihood ratio} \geq 1 \\ \omega_2, & \text{otherwise} \end{cases} \tag{5.11}$$

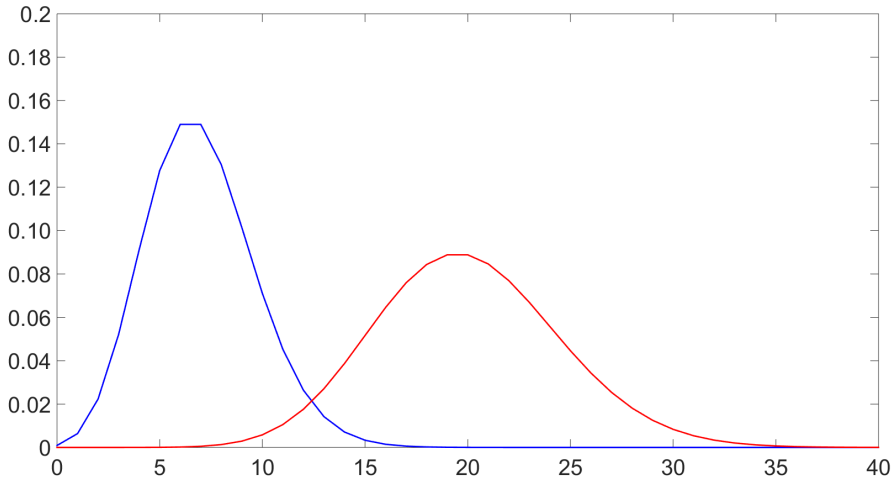


Figure 5.4: Example of two Poisson distributions representing noise (blue) and signal (red).

In order to calculate the likelihood ratio shown in Equation 5.10 we need to estimate the unknowns λ_1 and λ_2. In order to perform this estimation we use the slice-based horizontal projection technique which was explained earlier. After estimating the signal and noise parts, the mean number of points per meter is calculated for both noise and signal as estimates for λ_1 and λ_2, respectively. This method removes a significant amount of noise. Figure 5.5 shows an example of the result of this technique applied on the example transect shown in Figure 4.2.

5.4.3 Noise reduction using Wiener filter

The third noise reduction method is applying an adaptive Wiener filter [103] on the image. This filter uses a pixelwise adaptive Wiener method based on statistics estimated from a local neighborhood of each pixel. It estimates the local mean μ and variance σ around each pixel where,

$$\mu = \frac{1}{NM} \sum_{n_1,n_2 \in \eta} I(n_1, n_2) \tag{5.12}$$

$$\sigma^2 = \frac{1}{NM} \sum_{n_1,n_2 \in \eta} I^2(n_1, n_2) - \mu^2 \tag{5.13}$$

where η is the N-by-M local neighborhood of each pixel in the image I. The adaptive Wiener filter is then created using these estimates,

$$b(n_1, n_2) = \mu + \frac{\sigma^2 - \upsilon^2}{\sigma^2}(I(n_1, n_2) - \mu) \tag{5.14}$$

where υ^2 is the noise variance. The main drawback of this method is that it cannot deal with the noise of variable densities because it assumes a specific distribution for the noise in the whole transect. Figure 5.6 shows an example of the result of this technique applied on the example transect shown in Figure 4.2.

5.4.4 Noise reduction using morphological reconstruction and median filters

After mapping the point cloud onto a 2D grid, it is straightforward to apply many image processing techniques including 2D median filtering and morphological operations such as opening and closing. Using these two methods we can reduce a significant amount of noise, but these methods always affect the signal, especially under dense canopy. Figure 5.7 and 5.8

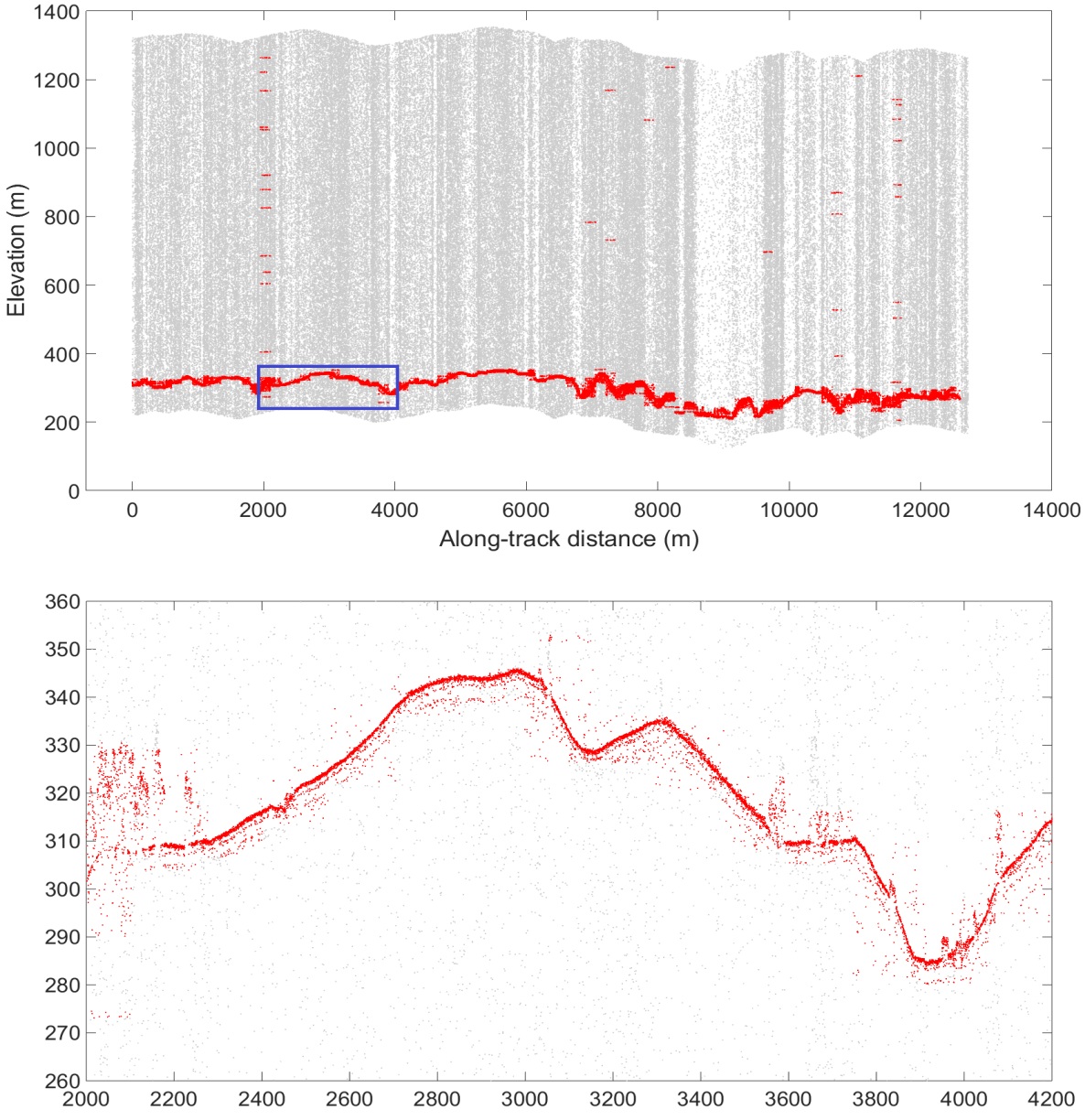


Figure 5.5: An example result of Bayesian noise reduction applied on the image shown in Figure 4.2. A close up for the blue rectangle is shown at the bottom.

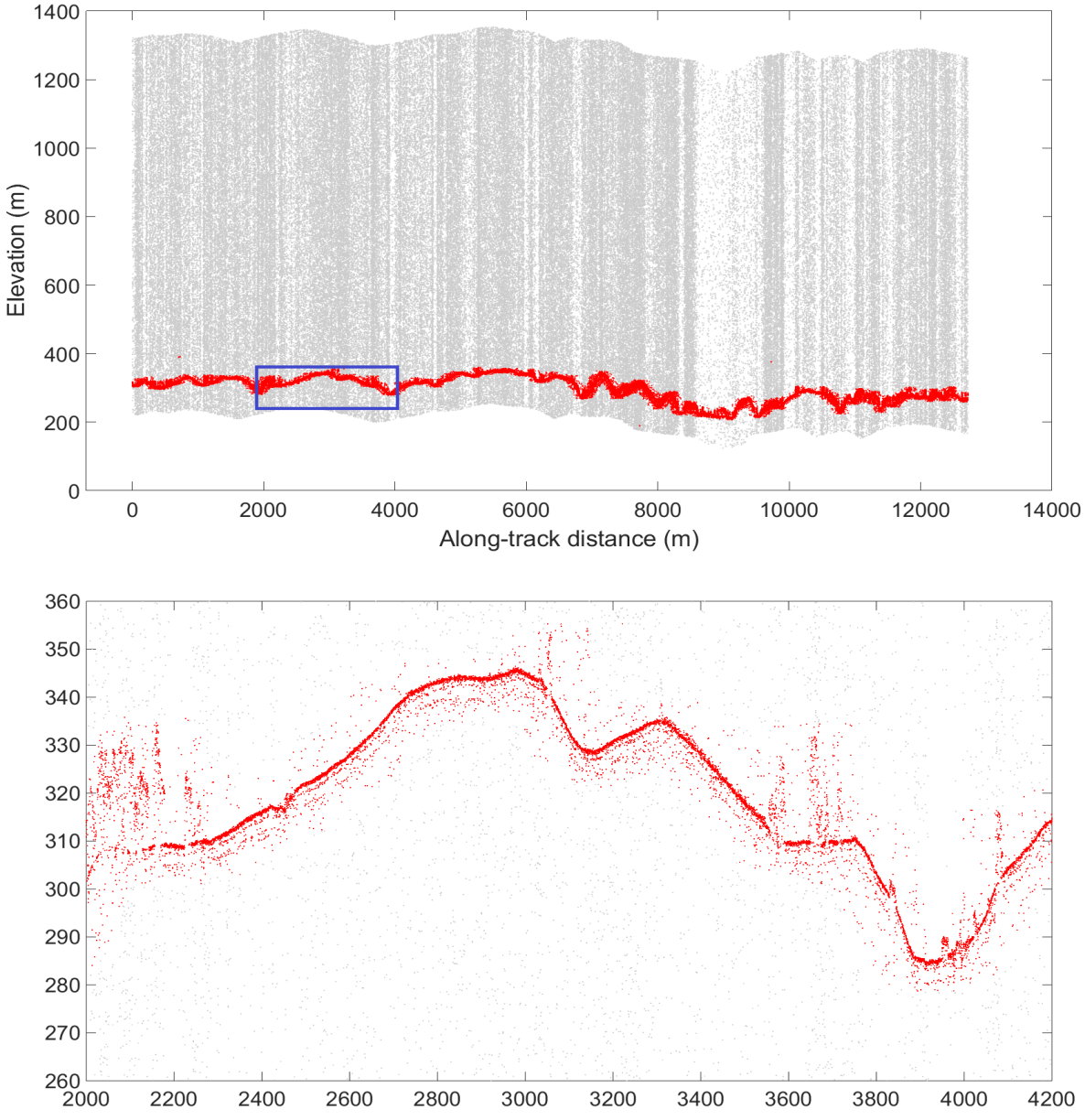


Figure 5.6: An example result of Wiener filter noise reduction applied on the image shown in Figure 4.2. A close up for the blue rectangle is shown at the bottom.

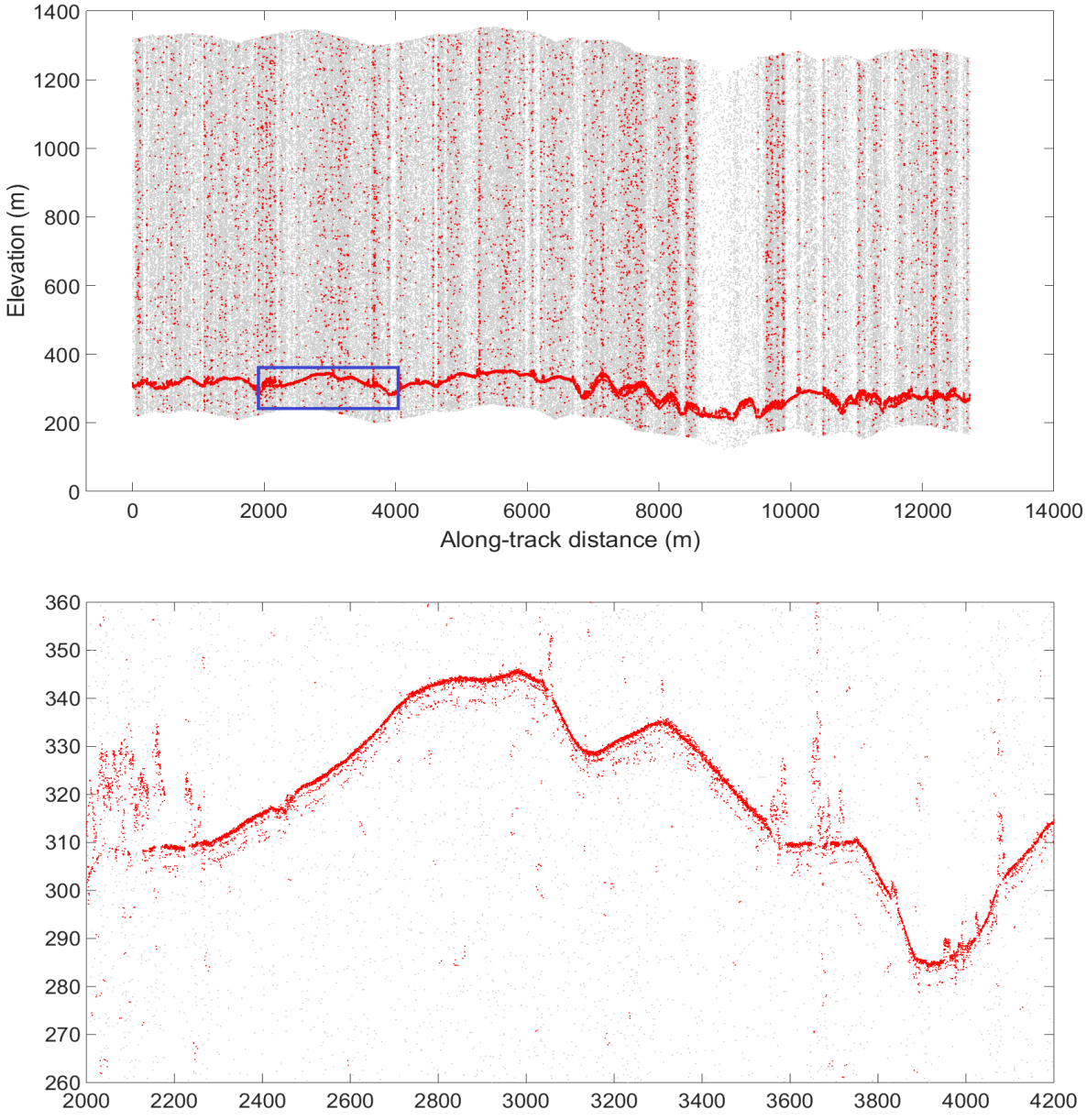


Figure 5.7: An example result of morphological noise reduction applied on the image shown in Figure 4.2. A close up for the blue rectangle is shown at the bottom.

show examples of the result of these two techniques applied to the example transect shown in Figure 4.2.

The morphological noise reduction method applies a morphological reconstruction operation which was first proposed by Vincent [104]. This composite operation performs an erosion followed by a reconstruction for the remaining connected components in the image.

5.4.5 The joint noise reduction method

The proposed joint noise reduction method is based on combining the results of all the previously mentioned noise reduction techniques. A majority voting for every pixel is used to determine whether it represents a noise or a signal photon. This technique combines the advantages of all the previously mentioned methods and avoids their drawbacks. The result of this technique is a nearly noise-reduced point cloud with very little loss of signal even under dense canopy. Figure 5.9 shows an example result of this technique applied to the example transect shown in Figure 4.2.

Although the noise reduction introduced in this section removes a significant amount of noise from point clouds, all noise reduction techniques suffers from losing some signal points. In many cases ground become occluded under dense canopies. In these cases, the few returns from the ground surface are very critical to any surface finding algorithm. Losing these few points might cause the ground finding algorithm to accidentally "jump" inside the canopy, losing the actual ground surface. Unfortunately, the density of ground surface points under dense canopies might have the same point density as background noise (sometimes even less). So, relying on a noise reduction technique alone might not lead to a good surface estimation. In the next section we will introduce our proposed framework to address this problem. A weighted voting procedure could be used instead of the simple voting mechanism. However, selecting appropriate weights is a challenging process given the fact that no single noise reduction algorithm shows superior results over all the others in all cases.

5.5 Initial surface estimation

After reducing noise, the next stage is to find the ground and TOC surfaces. In this chapter we propose using a modified version of ACWE (see section 2.3.2) which is a robust segmentation technique against noise, and is capable of handling concavities, sharp corners and background noise contamination that may present in the image. In this approach the associated curve is represented implicitly using a level-set function. It uses regional image statistics for segmentation, and as a result it is sometimes called a region-based level-set model.

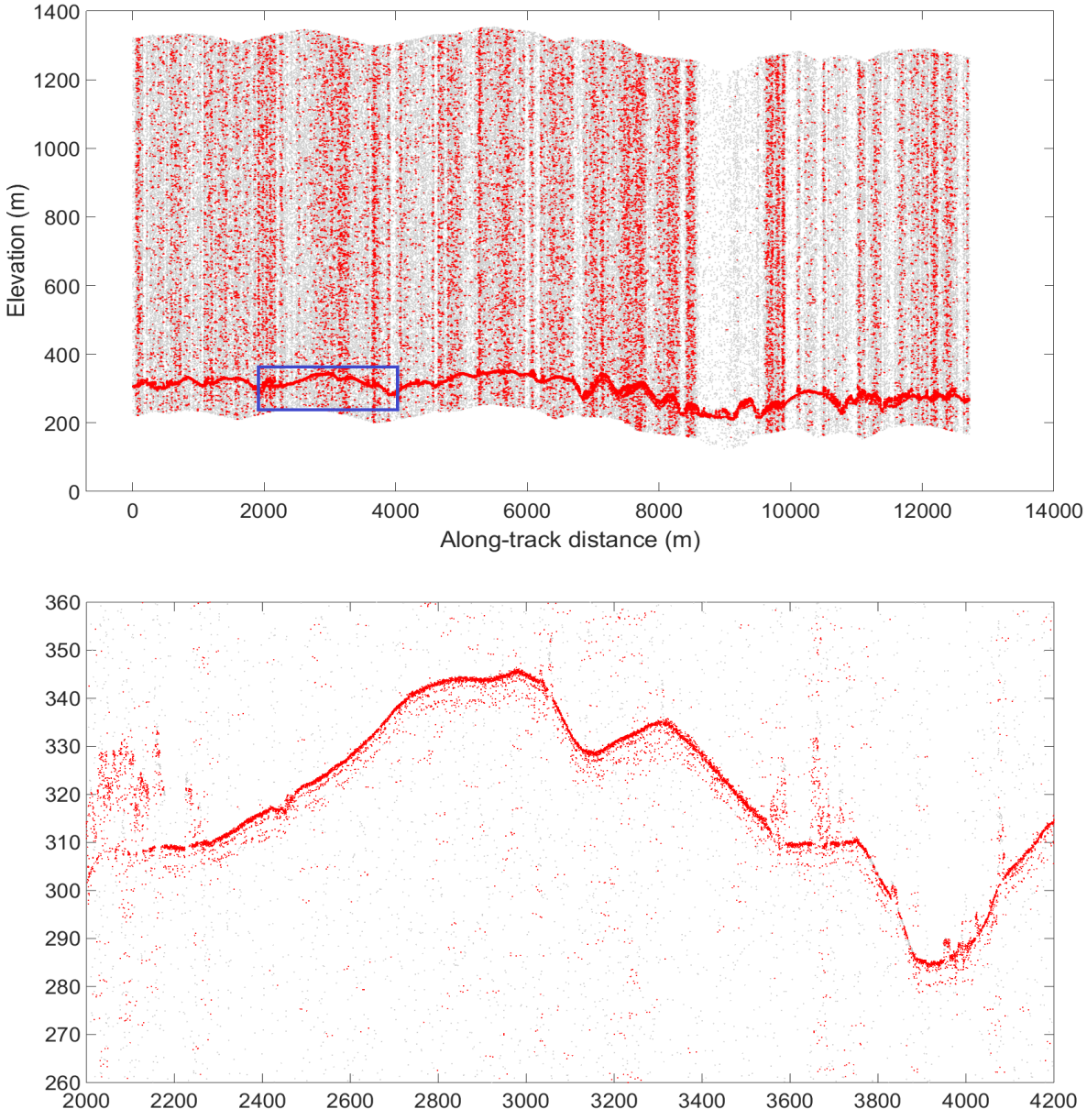


Figure 5.8: An example result of 2D median filter applied on the image shown in Figure 4.2. A close up for the blue rectangle is shown at the bottom.

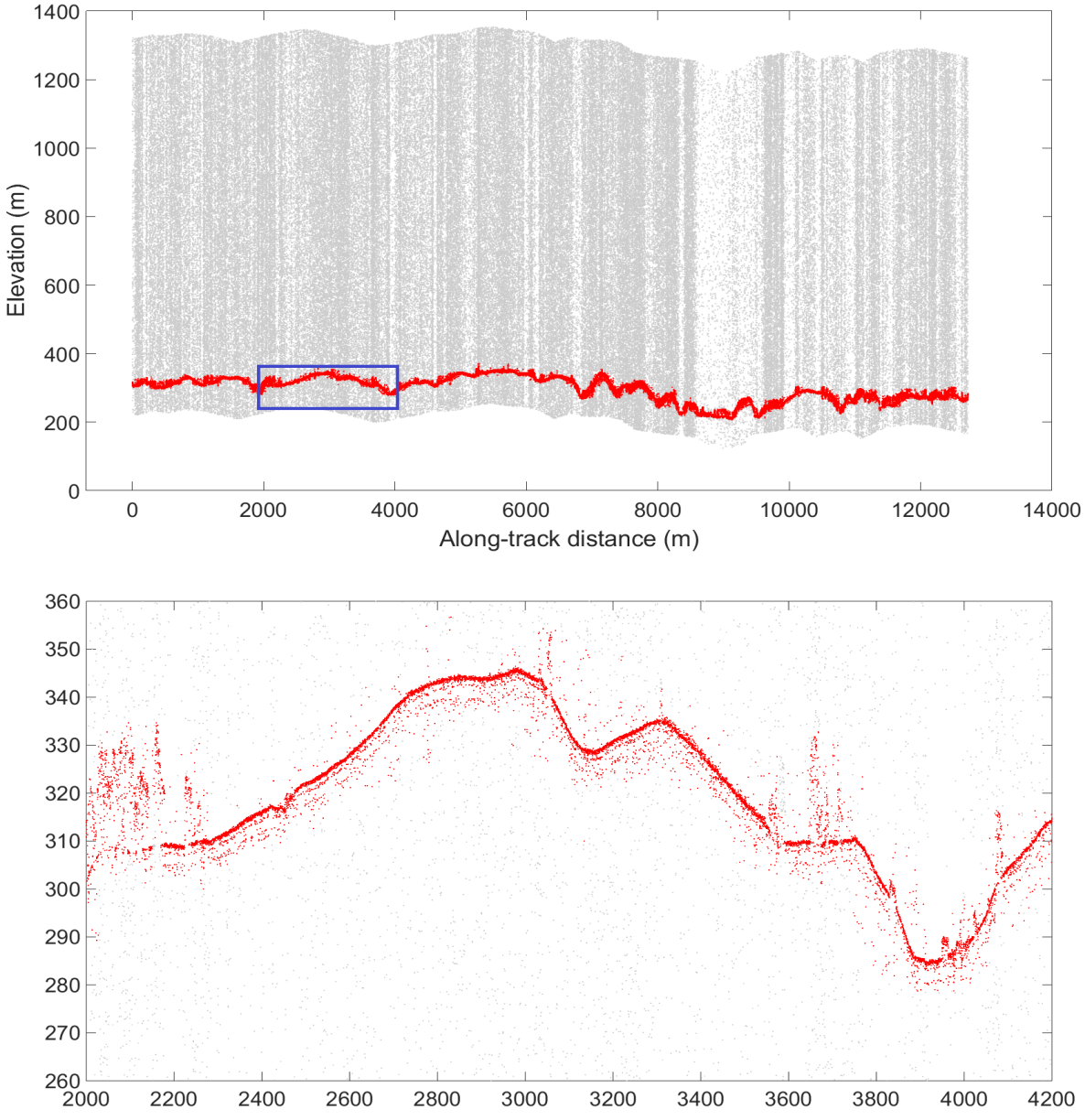


Figure 5.9: An example result of the joint noise reduction process applied on the image shown in Figure 4.2. A close up for the blue rectangle is shown at the bottom.

5.5.1 Modified ACWE

The LiDAR point cloud can be interpreted as a bimodal image I_0 that contains a single object of interest (vegetation plus ground) against a noisy background. In the original ACWE formulation, the object and background pixels average intensity levels in (2.5), the values c_1 and c_2, respectively, can be obtained from the image intensity values. However, our images are representations of point clouds, so we slightly change the original formulation to replace the average intensity by the average density. In our formulation, c_1 and c_2 represent the average *point density*. The density around a point P is calculated by counting the number of points in a window of size $N \times M$ centered on P. Then, from this perspective, the goal can be formulated as a search for a closed curve C within the image such that all signal (object) points are inside the curve, and all noise (background) points are outside the curve. The energy term was shown in (2.6) which is minimized when the contour C is coincident with the boundary of the foreground object in the image. The function $\phi(x,\, y)$ represents the signed Euclidean distance of an image point $(x,\, y)$ from the curve C:

$$\phi(x,y) = \begin{cases} 0 & \textit{if } (x,y) \textit{ is on the curve} \\ < 0 & \textit{if } (x,y) \textit{ is inside the curve} \\ > 0 & \textit{if } (x,y) \textit{ is outside the curve} \end{cases} \tag{5.15}$$

The function ϕ is therefore an implicit representation of the curve C, and is called the embedding function for C. The evolution equation is given by (2.7) which is composed of two terms. The first term $\mathrm{div}(\frac{\Delta\phi}{|\Delta\phi|})$ represents the curvature of C and is calculated as:

$$\mathrm{div}(\frac{\Delta\phi}{|\Delta\phi|}) = -\frac{\phi_{xx}\phi_y^2 - 2\phi_x\phi_y\phi_{xy} + \phi_{yy}\phi_x^2}{(\phi_x^2 + \phi_y^2)^{3/2}} \tag{5.16}$$

The second term is the model-fit error term which drives the evolution of the contour. This error term is minimized when the contour perfectly subdivides the image in such a way that the average densities inside and outside the contour perfectly match the constants c_1 and c_2, respectively.

An advantage of this approach is that strong intensity edges between foreground and background are not necessary for successful operation. However, this algorithm is very sensitive to the initial location of the contours. A proposed solution to this problem is to introduce a higher level technique to find a good initialization for the contours.

5.5.2 Sensitivity to initialization

When applying level sets to point clouds for segmentation, initialization is a key factor in determining the final result. If the initial curve is far from the ROI, it is more likely

to converge on a local minimum. Moreover, the presence of a large amount of noise may confuse the contours if it was not properly initialized close enough to the ROI. In addition, initializing level set far away from the ROI will take longer time to converge. Given that the noisy point clouds typically includes a large number of points, the convergence time needs to be considered.

Figure 5.10 and Figure 5.11 depict the importance of initialization. In Figure 5.11(a) our initialization technique was used to set the initial level set close to the top of canopy and ground surfaces. The result after 50 iterations is shown in Figure 5.11(b). Figure 5.10(a) and Figure 5.10(b) show another initialization of the contours away from the ROI along with the result of applying level sets for 600 iterations, respectively. In the case of initializing contours far away from the ROI the level sets got trapped in a local minimum far away from the required surfaces. The same effect of initialization appears in Figure 5.12 in synthetic data. The arbitrary initialization converged at the wrong contours after 2000 iterations with classification accuracy of 0.81, while the proposed initialization method converged after 500 iterations with classification accuracy of 0.93, which is significantly more accurate and more efficient.

The algorithm proposed in this chapter is fully automatic so it is necessary to have an automatic initialization technique, because a "good" initialization will result in a reduction of the convergence time and it also reduces the possibilities to converge at wrong contours due to presence of noise.

5.5.3 Proposed initialization methods

The framework presented here proposes an automatic procedure to discover a ROI to initialize the location of the curve near ground and TOC. A good selection of ROI helps prevent the attraction of the level sets to incorrect solutions, and can improve computation speed by reducing the number of iterations required to converge. We propose two methods to initialize the level set ($\phi_0(x, y)$) very near to the ROI. It is not a trivial problem in the presence of large amount of noise.

The first method is to use the slice-based horizontal projection explained in section 5.4.1 as an initialization technique for level set. Figure 5.13 shows detailed pseudocode for this initialization process. The second method is to use the result of the joint denoising method explained in section 5.4.5 as an initial ROI for level set. This can be obtained by dilating the denoised image to form a mask that can be used as $\phi_0(x, y)$ for ACWE. Both of these methods showed good initializations for level set in all cases of the dataset in hand.

5.5.4 Proposed stopping criterion

One of the most important parameters of level sets when applied to a point cloud is when to stop evolving contours (stopping criterion). Most of the stopping criteria in literature are based on the stationary curve, the stationary energy, maximum curvature, minimum curve length, or the maximum number of iterations. These stopping criteria work well for many applications. However, in the cloud of points the situation may be different. In many cases, the signal points are sparse enough to let the level set vanish without capturing the required signal leading to false estimation of ground and canopy surfaces. In this research we also propose a novel stopping criterion based on the sparsity estimation using Delaunay triangulation in a narrow band around the evolving contour. First, a narrow band around the evolving contour is defined (usually 5 pixels away from the original curve), then in each contour iteration, the average triangle area inside this narrow band is calculated. If the average triangle area in the narrow band begin to decrease "significantly" (exceeding a threshold), the optimization process has to stop at this point.

5.5.5 Adaptive parameter selection method

The proposed system has a set of parameters to choose. These parameters are: (1) size of image to which point clouds are mapped (see section 5.2), (2) maximum number of level set iterations, and (3) the size of the structuring element used for dilation operations in the denoising steps. In this research we propose an adaptive method to adjust the system parameters that consists of two major steps: feature extraction and parameter selection step.

In the feature extraction step, a set of features are extracted from the point cloud. These features are: (1) an estimation for the SNR, (2) noise density, (3) skewness S (see (5.17)) of the full image's horizontal projection and (4) the estimated density of signal points calculated by the average triangle area of the Delaunay triangulation of signal points. These parameters uniquely indicate the anticipated signal strength, the expected time of acquisition (daytime/night), the roughness of the terrain, and the expected point density of the point cloud under study. Using this set of features an adaptive method is used to adjust the system parameters for efficient convergence of the level sets.

$$S = \frac{m_3}{m_2^{3/2}} = \frac{\frac{1}{n}\sum_{i=1}^{n}(x_i - \bar{x})^3}{(\frac{1}{n}\sum_{i=1}^{n}(x_i - \bar{x})^2)^{3/2}} \tag{5.17}$$

The feature extraction step is then followed by a parameter selection step. In this step, we use Classification And Regression Tree (CART) analysis in order to predict the values of the system parameters. The tree was trained on the extracted four features using the datasets and ground truth provided by NASA (introduced in chapter 4). The training optimization criterion was to choose a split that minimizes the MSE of predictions compared to the

training data. The trained tree was used to predict an estimation for the values of the required parameters.

5.5.6 Ground and Top-of-Canopy Delineation

After detecting contours as described in the previous sections, the next stage is discriminating ground and canopy from the resultant contours. The ACWE output is a closed contour, or possibly several closed contours, so a further step is required to delineate ground and canopy surfaces. In our approach, the contour is digitized into image coordinates (x,y) for every column of the image. Then, at every column of the image, the uppermost points from the contour are assigned to the canopy surface, and the lowest points from the contour are assigned to the ground surface. We use linear interpolation to fill any gaps that result in either surfaces.

5.5.7 Edge detection using CORF

In some cases level sets do not introduce the best results for initial surfaces. These cases are when point cloud has a high SNR especially when the ground surfaces shows a strong response. In these cases, the ground shows a strong response and the noise points under the ground surface (including signal echo) becomes strongly evident. The level sets in these cases get trapped to the dense noise points under the actual ground surface and they fail to find a good initial ground estimation. Therefore, in the high SNR cases, we propose using an edge detector to find the initial ground and canopy surfaces after the noise reduction step. The CORF edge detector [102] introduces a very good directional edge detector. It performs an enhanced Canny-like [105] edge detection process. The enhancements proposed in CORF are based on novel biological findings of the vision neurons [102]. The CORF approach showed a very good success for the aforementioned cases because of the presence of a high signal response of the ground and canopy surfaces.

5.5.8 Surface refinement process

Due to the strong noise under the actual ground surfaces especially in the bare-ground areas, our initial estimation of ground surfaces fail to track the actual surface in these areas. The noise in these areas usually result from two sources: solar energy and sensor echo. Solar energy noise appears sparse and random similar to other parts of noise distributed in the transect. The sensor echo noise appears as a parallel strong surface(s) under the actual ground surface. Figure 5.14 shows an example of a part of a MABEL transect with the initial ground surface estimated using ACWE. As shown in this figure, the ACWE algorithm was able to estimate ground under dense canopies accurately but it failed to find the ground

surface in bare-ground areas. In order to produce a more accurate ground surface, some techniques are required to refine the estimated ground surface.

We employ a statistical techniques to refine the estimated surfaces. In our approach, we use a statistical local regression process in order to classify every part of the transect into either a bare ground or a vegetated area. This distinction allows the algorithm to adjust the ground and canopy surfaces differently according to the type of the area (bare ground or vegetation). This process greatly enhances the final surfaces because it is common for the initial contours obtained in the previous steps to get trapped under the actual ground surfaces in bare ground areas but it has a good initial estimation for ground and canopy surfaces in vegetated areas.

5.5.9 Identifying bare ground and vegetated areas

The initial contours produced in high altitude point clouds have a tendency to mis-classify noise points under bare ground surfaces as ground (as explained earlier). In order to find the ground surface accurately, a method of identifying bare ground areas should be used in order to treat them differently from the vegetated areas. The algorithm uses a non-parametric regression method called locally weighted scatterplot smoothing in order to fit a curve to the signal points of the whole transect. We call this curve the LOWESS curve. This curve tends to be closer to the point clusters that have higher density than their surroundings. In bare ground areas, the actual ground surface has a much higher point density than its surroundings. This allows the fitted curve to pass through the actual ground surface in these locations. So intuitively, the points in these locations will have smaller residuals from the curve. Therefore, the transect is then divided vertically into smaller sections. The size of each section is decided to be the area that includes 200 signal photons. The number of photons in each section is selected based on a informal study performed by members of ICESat-2 science definition team. The study revealed that the best results could be obtained by dividing the transect into variable length sections based on the number of signal photons, and they recommended the use of 200 signal photons in every section. The perpendicular residual between all the signal points in each section and the central curve is calculated.

Calculating the perpendicular residual means to measure the distance from every point in the point cloud to its closest point on the LOWESS curve. This distance is much more suitable than the vertical residual because in the very steep areas the vertical distance might be very large even if the point is very close to the curve perpendicularly ("actually"). Figure 5.15 illustrates this concept on a part of MABEL transect. In this particular instance, the vertical distance between the point and the curve is more than 14m, but the point is very close to the LOWESS curve. In this specific case the perpendicular distance is less than 2m which represents the "actual" residual between the point and the curve for the purpose of identifying ground sections in the transect.

This perpendicular distance can be calculated using simple mathematical concepts. Assum-

ing that the curve is piece-wise linear, we divide the curve into small line pieces and the distance is calculated for every line piece and the corresponding points in the point cloud. This reduces the problem to calculating the perpendicular distance between a line and a point. Given that the equation of a line as: $y = ax + by + c = 0$. This line's equation could be represented in the slope-intercept form by: $y = -\frac{a}{b} - \frac{c}{b}$. So this line has a slope of $-a/b$. Now consider the distance from a point (x_0, y_0) to the line. We first find the vector V perpendicular to the line which is:

$$V = \begin{bmatrix} a \\ b \end{bmatrix} \tag{5.18}$$

So a vector from the point to the line is given by:

$$r = \begin{bmatrix} x - x_0 \\ y - y_0 \end{bmatrix} \tag{5.19}$$

Projecting r onto V with some algebraic simplification we get the distance d as follows:

$$d = \frac{|ax_0 + by_0 + c|}{\sqrt{a^2 + b^2}} \tag{5.20}$$

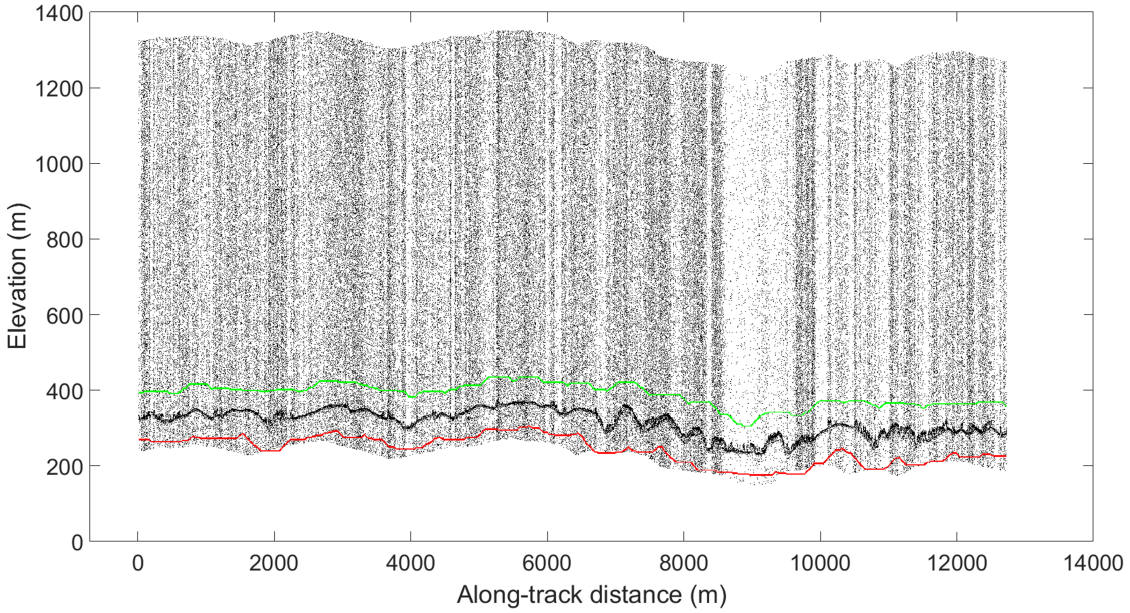
Then we calculate the mean absolute deviation (MAD) of the residuals calculated in every part. If the calculated MAD is very small, this area is marked as bare ground area, and otherwise this area is marked as vegetated area.

For every section of the transect, if this part is marked as bare ground, an elevation histogram is calculated and a Gaussian fitting method is used to decide the actual elevation of the ground surface in this part. If this part is marked as vegetation, the ground is adjusted for this part and the adjacent sections using spline curve fitting. Some parts include mixed vegetation and bare ground, in this case it is further divided and reclassified again. The final step is to fit a spline to all the points that is classified as ground in the whole transect and readjust the point classifications according to the fitted spline curve. The top of canopy surface is mainly obtained from the ACWE. The common approach to estimate the canopy surface from LiDAR data is to classify the highest 95th percentile of points above the estimated ground surface as canopy points. In our approach, we estimate initial canopy surface using ACWE and then we finally adjust the initial canopy surface afterwards using the highest 95th percentile of points above the estimated ground surface.

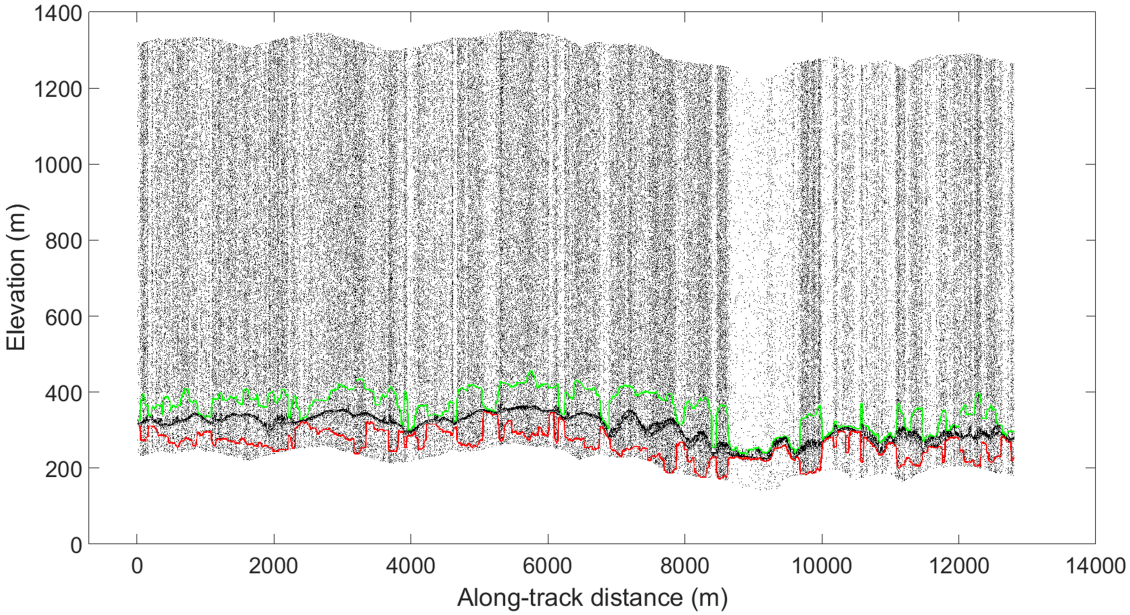
5.5.10 Drawbacks of ACWE approaches

It will be shown in the discussion of results (Section 7.1) that the performance of ACWE algorithm degraded clearly because of the presence of complex terrain, high noise density and occluded ground under dense canopy. Despite that ACWE approach showed good surface

estimation results in many cases but this drawback prevent us from depending only on this approach for surface estimation from ATLAS point clouds. In the next chapter we will tackle this drawback by introducing another surface estimation approach that is based on superpixel segmentation.

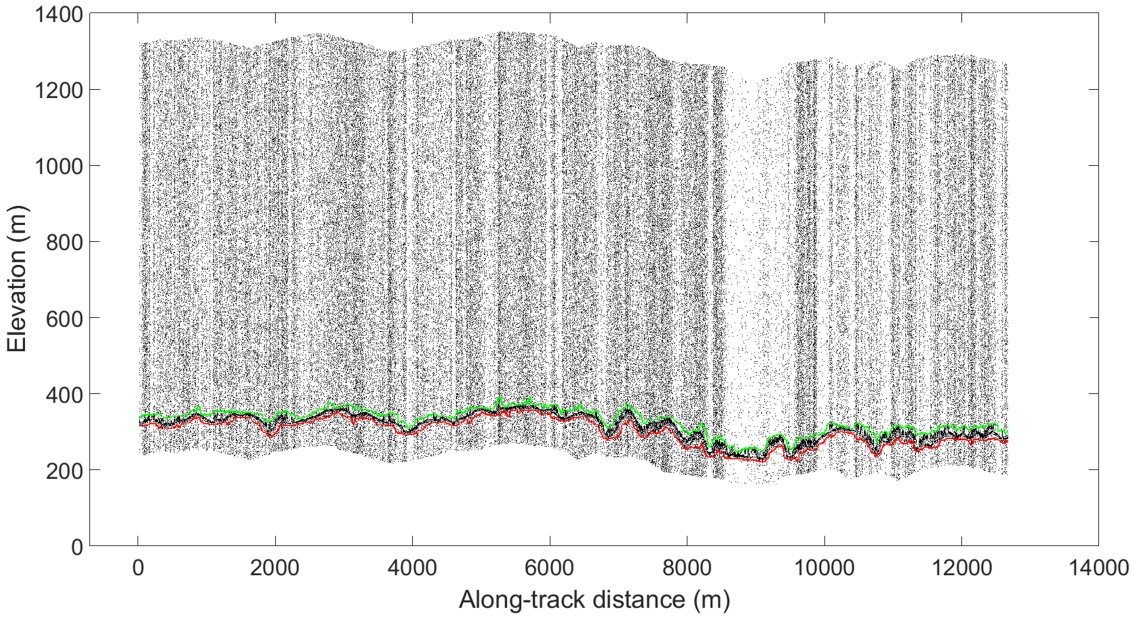


(a)

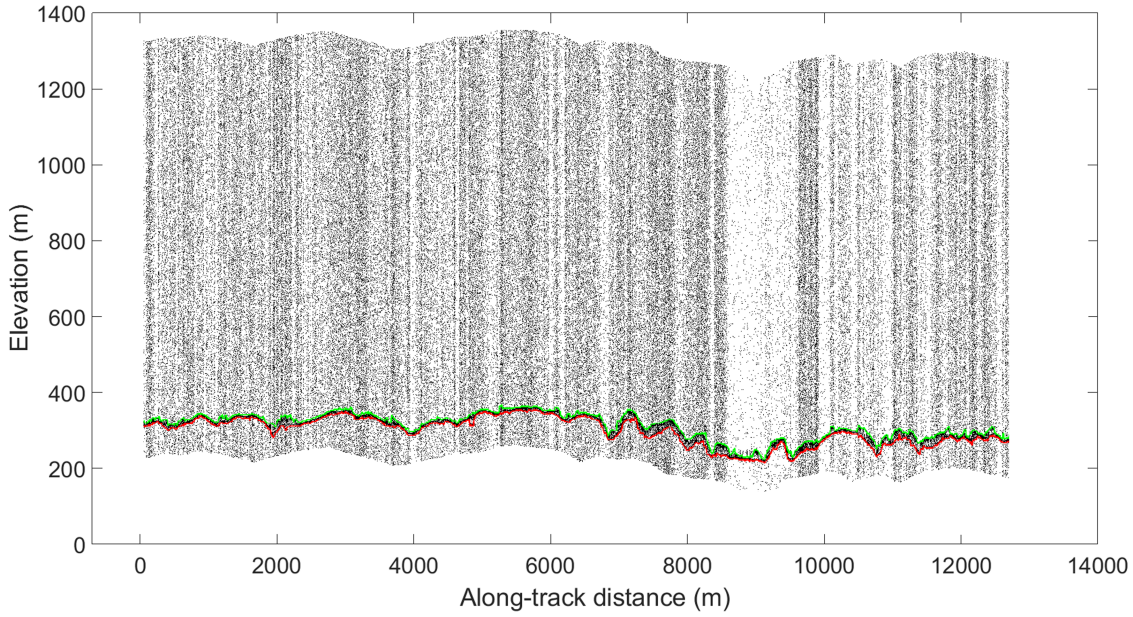


(b)

Figure 5.10: The effect of initial contours on the final result for one of the MABEL transects. Contours initialized far away from the ROI (a) and the result of ACWE after 600 iterations (b). The segmentation result of ACWE shows poor result that converged far away from the signal photons.



(a)



(b)

Figure 5.11: The effect of initial contours on the final result for one of the MABEL transects. Initial contours using the proposed method (a) and final results after 50 iterations (b). The segmentation result of ACWE shows much better result using the proposed initialization method compared to Figure 5.10. Zoomed sections of this transect are shown in Figure 7.6(c) and 7.31(c).

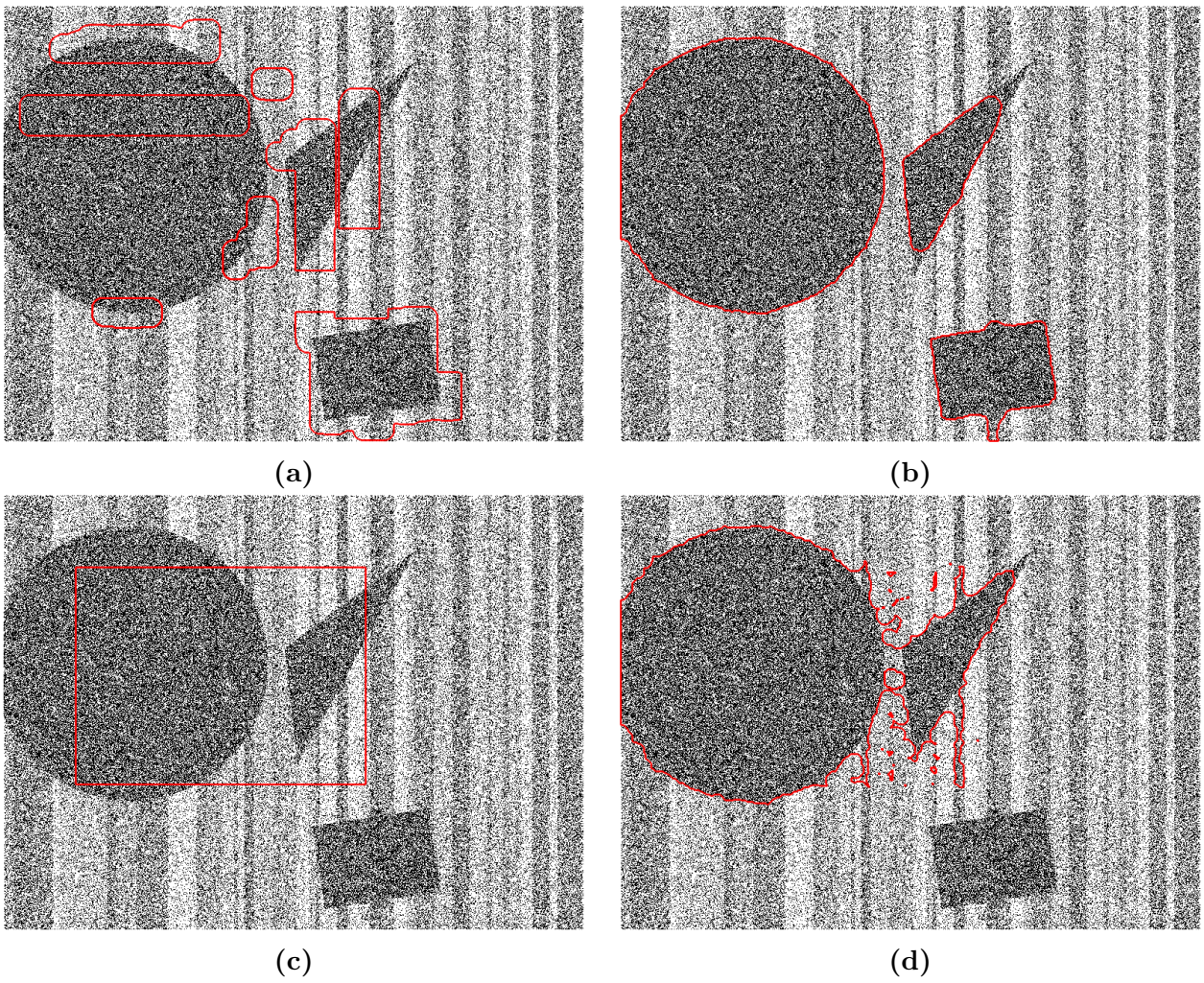


Figure 5.12: The effect of initial contours on the final result for one of the synthetic cases. Initial contours using the proposed method (a) and contours initialized far away from the ROI (c), final results after 500 iterations with accuracy 0.98 (b) and after 2000 iterations with accuracy 0.89 (d) for initial contours shown in (a) and (c) respectively. Good initialization derived the contours to better results.

Input: point cloud binary image I.
Output: initial ϕ_0.

Divide I into a set of vertical slices $V_I(n)$ *(n = 1,2,... N);*
for each slice $V_I(i)$ *do:*
    proj(i) = horizontal projection of $V_I(i)$*;*
    normalize proj(i);
    calculate edges $E_v(n)$ *and their directions* $D_v(n)$ *form proj(i);*
    for each e in $E_v(n)$ *do:*
        if $D_v(e)$ *positive:*
            find the matching -ve edge e_m*;*
            leave rows in $V_I(i)$ *between e and* e_m *unchanged;*
        otherwise
            remove points in the rows between e and e_m*;*
        end
    end
end:
ϕ_0 *= dilate I*

Figure 5.13: Horizontal projection initialization technique.

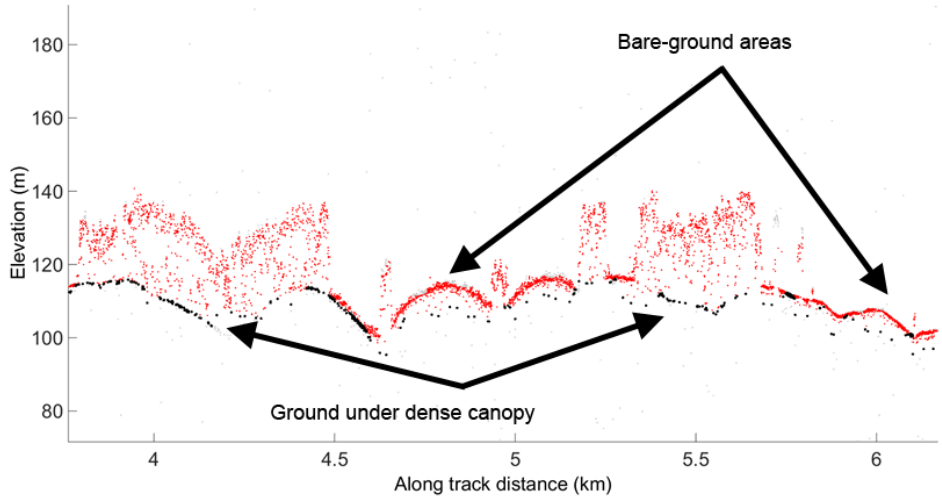


Figure 5.14: Example to show the initial estimated ground surface in one of the MABEL transects using ACWE. The initial estimation was accurate under dense canopy but it showed a tendency to mis-classify noise points under bare ground surfaces as ground.

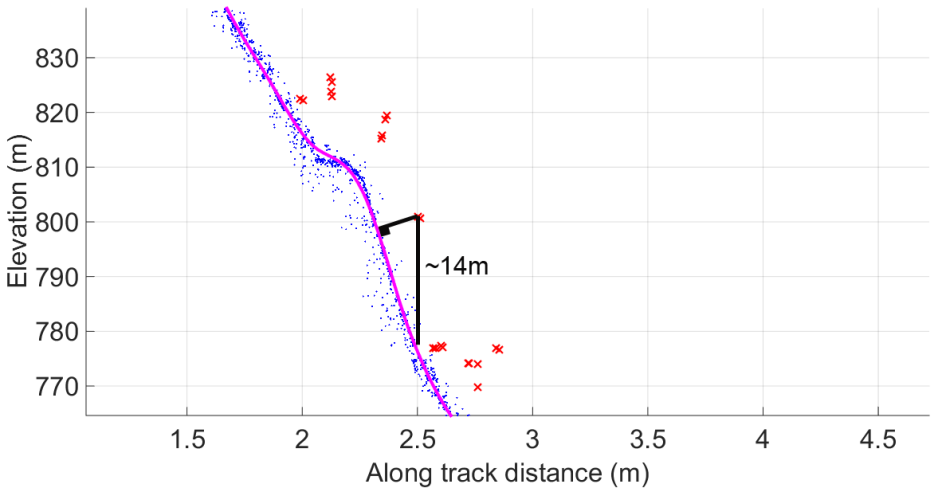


Figure 5.15: Example to illustrate the perpendicular residual vs. the vertical residual between a point and the regression curve. The transect is divided such that a single section cannot include both vegetation and bare-ground.

Chapter 6

Superpixel Segmentation of Noisy Point Sets in the Plane

6.1 Overview

In this chapter we address the problem from a general point cloud segmentation perspective. We address the problem of segmentation of planar sets of points for which point density is the only discriminating feature between foreground and background. The presence of noise within these planar point clouds, particularly impulsive noise with varying density, can make it difficult to obtain a good segmentation using traditional techniques (e.g., graph cuts) when the only discriminating feature is point density.

At first glance, it may appear that an image segmentation algorithm can be applied to this problem. However, those techniques tend to rely heavily on color, intensity edges, texture descriptors, and possibly shape-based prior information to obtain good results. We assume that color information and shape priors are not available. In our experiments, edge information and texture descriptors have proven to be unreliable because of the nature of the noise, and the results of image-based segmentation were poor.

Many algorithms have been developed to perform general clustering and/or segmentation of point clouds in $\mathbb{R}^n$, so initially it may seem surprising that the examples in Figure 1.3 are not easily solved. For traditional techniques, the major hurdle is the high level of noise along with the nonuniform density of points. Mean-based clustering methods such as k-means and k-medoids do not perform well for this type of point cloud because of the noise, and because of the potentially irregular shapes of the signal clusters. The DBSCAN algorithm [100] and revised versions (e.g., [101]) can find arbitrarily shaped clusters in noisy data for some applications, but they yield poor results here (see Figure 6.1). Nonuniform density causes the selection of parameters (minimum number of points, radius of search neighborhood) to be intractable for our data.

Spectral clustering and graph cut algorithms such as normalized cut (N-cut) [83], [106] also seem like good candidates for the segmentation problem discussed here. These techniques represent every point in the point cloud as a node in a graph, with the distances between points represented by edge weights connecting the nodes. However, as shown later in this dissertation, the variable density and absence of signal in some parts of the point cloud often lead these algorithms to incorrect segmentation results. These algorithms also have the drawback that the computational cost rises sharply with increasing numbers of points.

Other techniques include total variational and global minimization segmentation algorithms such as continuous maximal flow (CMF) [97], [98]. Total variation denoising (TVD) [99] was also considered. However, in our experiments these techniques also failed to achieve a good segmentation in many cases when confronted by the levels of noise that are of interest here (see Figure 6.1).

We introduced a segmentation approach that relies on deformable models, using active contours without edges [36] (level sets) in the previous chapter [82]. This is the only work comparable to the algorithm presented here. The main drawback of this algorithm is that it requires many parameters to be adjusted for a successful segmentation. It also proved to be sensitive to initialization, and it requires a robust stopping criterion.

For all of the reasons presented above, we believe that an approach based on *local grouping of points followed by a global clustering step* is needed in order to produce a successful segmentation of these point clouds, without the need to tune many parameters. We have therefore explored superpixel-based methods, adapting them fundamentally so that 1) they are not constrained to a regular grid, and 2) they rely almost entirely on point density.

The result is a novel segmentation approach for noisy point clouds in the plane that we call Point Cloud Density-based Segmentation (PCDS). PCDS is a density-based approach that performs segmentation by first performing a local grouping of points into over-segmented clusters of nearly uniform density that are analogous to superpixels. Final segmentation is performed through extraction and refinement of superpixel boundaries. To our knowledge, this is the first superpixel-based method for segmenting noisy point clouds in the plane, and we present results that demonstrate significantly better performance than previous techniques.

The approach is based on a Markov random field (MRF) optimization that we have developed for density based point cloud segmentation. The resulting superpixels conform to object boundaries and have better agreement to the data than state-of-the-art superpixel segmentation algorithms. The emphasis in this work has been the development of an appropriate density-based energy function, rather than the creation of iterative update strategies. As a result, the new energy term can be plugged into existing systems for comparison. We have performed these comparisons by inserting our density-based energy computation in place of the color-based energies currently implemented in existing algorithms [57], [88], [89]. Moreover, a regularization term was introduced to prevent degenerate or "trivial" superpixels that contain no signal or noise points. The new technique has yielded very good results

quantitatively and qualitatively in our experiments (more details are given in Chapter 7). After finding superpixels, the algorithm also finds region boundaries.

6.2 Point Cloud Density-based Segmentation (PCDS)

The proposed PCDS algorithm consists of two major stages: density-based superpixel segmentation, followed by global identification of foreground/background superpixels.

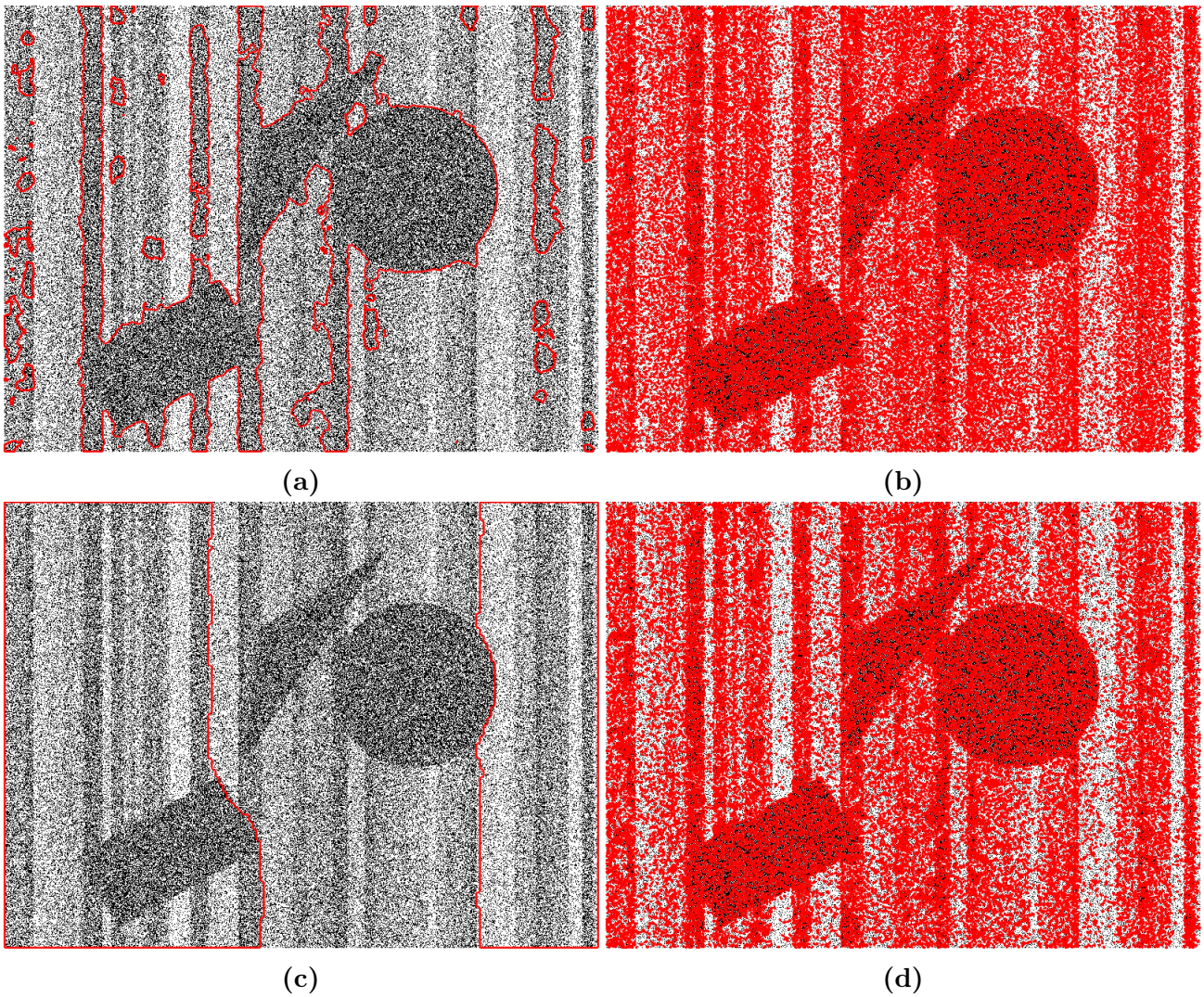


Figure 6.1: Examples of the result of traditional segmentation algorithms. The detected edges are shown in red, (a) continuous maximal flow (b) total variational denoising (c) N-cut (d) revised DBSCAN.

6.2.1 Mapping point clouds onto a planar grid

The first stage of PCDS is to perform a 2D quantization of the point cloud into a grid that encodes the spatial locations of points. This process is explained in detail in the previous chapter. This quantized point cloud can be represented by an image that facilitates applying image processing algorithms on the point cloud directly.

6.2.2 Proposed MRF energy

Given a point cloud D in the plane with N points, our first goal is to partition D into a set of K superpixels. Assume that the point cloud has been mapped onto the quantization plane I with M gridding pixels. Partitioning this plane I into K superpixels can be represented with the mapping:

$$s: \{1, \ldots, N\} \rightarrow \{1, \ldots, K\} \tag{6.1}$$

Let $s_p \in \{s_1, \ldots, s_K\}$ be the superpixel to which pixel p was assigned. Here we follow [89] in formulating our segmentation problem as a k-means-like objective function. The superpixels are required to be coherent in the density of the points contained in this superpixel and also to have regular shape. We propose the use of a density term E_{den}, and we add a regularization term to prevent empty superpixels (which do not contain any points from the cloud). Let μ_i represent the center of mass of the i-th superpixel and let d_i represent the mean point density (explained later) of the pixels included in that superpixel. The proposed MRF energy is

$$\begin{aligned} E(\boldsymbol{s}, \boldsymbol{\mu}, \boldsymbol{d}) = \lambda_d \sum_p E_{den}(s_p, d_{s_p}) + \lambda_{pos} \sum_p E_{pos}(s_p, \mu_{s_p}) + \\ \lambda_b \sum_p \sum_{q \in N_8} E_b(s_p, s_q) + E_{topo}(\boldsymbol{s}) + E_{size}(\boldsymbol{s}) \end{aligned} \tag{6.2}$$

where $\boldsymbol{s} = (s_1, ..., s_M)$ is all the random variables that represent the segmentation, $\boldsymbol{\mu} = (\mu_1, \ldots, \mu_K)$ is the set of centers, and $\boldsymbol{d} = (d_1, \ldots, d_K)$ is the set of mean density values for all superpixels, and N_8 the 8-neighborhood q of pixel p. The values λ_d, λ_{pos}, and λ_b are weights to control the contribution of every term. The energy terms can be defined now in more detail.

Density coherence term: This is the core term for our energy function. It ensures that the points included in every superpixel are homogeneous in terms of a proposed density measure:

$$E_{den}(s_p, d_{s_p}) = (I(p) - d_{s_p})^2 \tag{6.3}$$

where $I(p)$ is the density around point at image pixel p and d_{s_p} is the mean density of all points contained in the superpixel s_p. In order to measure the density around a point we introduce a Radial Basis Function (RBF) whose value depends on the distance between a center point $c \in D$ and all points p_x in a predefined area of interest.

$$\Phi(p_x, c) = \Phi(\|p_x - c\|) \tag{6.4}$$

In this algorithm we define the density using the Gaussian RBF:

$$\Phi(p_x - c) = e^{-\frac{(p_x - c)^2}{2\sigma^2}} \tag{6.5}$$

We now define the density measure $I(c)$ around point c by summing up RBF values for all neighbors within a radius r,

$$I(c) = \sum_{x \in D_c} \Phi(\|p_x - c\|_2^2) \tag{6.6}$$

with $D_c = \{p_x \in D : \|p_x - c\| \leq r\}$ equal to the set of all points within distance r from the center point c.

The density calculated in (6.6) represents just one component of the proposed density descriptor. The other two components are n_i, which is the number of points projected in this pixel's grid cell in the projection plane, and $f_i = \|\max(d_i) - \min(d_i)\|$, the difference between maximum and minimum density values for the points inside the superpixel.

One advantage of this approach is that a relative orientation prior can be incorporated. This can be achieved by selecting the area of interest D_c to be an oriented ellipse according to the prior knowledge about the orientation of objects represented in the point cloud. The superpixel density descriptor d_{s_p} is calculated by averaging the values of $I(c)$ over all points enclosed in the superpixel (not the number of cells or pixels in the superpixel). We emphasize that all density calculations are performed using actual point coordinates, not the quantized grid locations. The quantization plane is used to facilitate nearest-neighbor search, as a replacement for a k-d tree structure, but quantization-plane coordinates are not used for computing point density.

Shape regularization: This term is similar to the one shown in [89]. It encourages the superpixels to have a regular shape:

$$E_{pos}\left(s_p,\ \mu_{s_p}\right) = \left\|p - \mu_{s_p}\right\|_2^2 \tag{6.7}$$

where μ_{s_p} is the centroid of the superpixel s_p.

Boundary length: This term encourages the superpixels to have short boundaries, therefore emphasizing low curvature:

$$E_b\left(s_p, s_q\right) = \begin{cases} 1, & if\ s_p \neq s_q \\ 0, & otherwise \end{cases} \tag{6.8}$$

Superpixel topology: This term is achieved by checking the superpixel connectivity before any block assignment update. It guarantees that the superpixel forms a connected component. This term was originally proposed in [88] and was also used in [89].

Superpixel minimum size: In this term we propose not only to force the superpixel to have a minimum size in terms of the number of pixels, but also to force the superpixel to have a minimum number of signal or noise points. This is in order to guarantee that no superpixel is composed of only the spaces between points in the point cloud. In this term we set a minimum threshold for the superpixel's size $T_s = \lambda_{size} \times L_s$, and a minimum threshold for the number of points included in the superpixel $T_n = \lambda_n \times K_s$, where L_s and K_s are the initial size and the initial number of points, respectively, of the superpixel s, for $1 > \lambda_n,\ \lambda_{size} > 0$.

In order to show the utility of the proposed energy function to perform successful superpixel segmentation for this kind of point cloud, we incorporated it into three state-of-the-art algorithms: Yao et al. [89], SLIC [57], and SEEDS [88]. In the next three subsections we will describe these modifications.

6.2.3 Modified Yao algorithm

Yao's algorithm [89] initializes the superpixels within a regular grid and uses a coarse-to-fine approach to gradually change the boundaries of theses superpixels to optimize a given color-based objective function. Our proposed algorithm uses the coarse-to-fine optimization technique. However, we optimize a density-based energy function that makes superpixel segmentation suitable for noisy point clouds in the plane. An example of the result of this modified algorithm is shown in Figure 6.2(right column). As compared to the results of the original algorithm shown in Figure 6.2(left column), the modified algorithm shows superpixel boundaries that are more regular and much better conformity to object boundaries.

6.2.4 Modified SEEDS algorithm

The SEEDS algorithm operates in a manner very similar to that of Yao. However, SEEDS optimizes a different energy function:

$$E(s) \;=\; H(s) \;+\; \gamma G(s) \tag{6.9}$$

where $H(s)$ is the term that controls the coherence of the color of the superpixels, and the other term $G(s)$ is the shape regularization term to control the superpixel boundaries. The algorithm proceeds by calculating the histogram of colors and the boundary term for every superpixel, and incrementally optimizes the energy term by changing the boundary blocks, which increases the color similarity within the superpixels. The disadvantage of this algorithm is that it handles the boundary regularization inefficiently [89], which may cause the resulting superpixels to have very irregular shapes (see Figure 6.2(left column)).

We replaced the color coherence term in this algorithm with the density descriptor explained earlier. Figure 6.2(right column) shows an example result of the modified SEEDS algorithm. The boundaries of superpixels that enclose noise points seem much more irregular; however, the superpixels that enclose the object points show much more regularity and conformity to object boundaries than the original algorithm.

6.2.5 Modified SLIC algorithm

The SLIC algorithm originally starts by assigning an initial grid of cluster centers. Then the algorithm changes pixel assignments to the nearest cluster followed by new cluster center calculation (similar to k-means) and repeats this process until no further change of cluster centers occurs. The only parameter of the algorithm is K, the desired number of superpixels. It shares the regular grid initialization with the other two algorithms. The algorithm converts images into the CIELAB color space in order to proceed. The algorithm performs clustering using a feature vector for every cluster center $C_i = [l_i\ a_i\ b_i\ x_i\ y_i]$, where $l_i, a_i,$ and b_i represent the three CIELAB color components and x_i and y_i represent the centroid of the superpixel.

We replace the feature vector by $C_i = [d_i\ n_i\ f_i\ x_i\ y_i]$, where d_i is the calculated density as shown in (6.6), n_i is the number of points projected in this cell of the projection plane, and f_i is the difference between $\max(d_i)$ and $\min(d_i)$ inside the superpixel. The results of this modified algorithm show significant improvement in terms of the conformity to object boundaries and the length of superpixel boundary and its shape regularity (see Figure 6.2(left column) and Figure 6.2(right column)). Figure 6.3 also shows a set of example results for these algorithm on a LiDAR-like synthetic point cloud.

6.2.6 Proposed full segmentation technique

The next step is to obtain a full segmentation of the point cloud by analyzing the superpixels globally in the point cloud. The fundamental assumption is that point density is lower in noise superpixels than in object superpixels. We propose two methods to perform the final segmentation, using Otsu's [107] thresholding method and using Gaussian Mixture Model (GMM).

Otsu's thresholding method

To perform the final segmentation using this method, we first calculate the average density descriptor for every superpixel. Then, a binning histogram is calculated for all superpixels' densities. This histogram of superpixel densities is used to adaptively calculate a density threshold based on Otsu's method [107]. The threshold represents an optimum separation of the superpixels into signal/noise categories. In a few cases, due to the nonuniform density of the noise, some noise superpixels may have point density very close (or even higher) than some of the signal superpixels. To resolve this problem we used another stage of classification based on the k-NN (k-Nearest-Neighbor) algorithm. The system calculates the density-weighted center of gravity of each superpixel, and then it finds the k nearest neighbors for every superpixel. (In all our experiments k was chosen empirically to be 5). Finally, the system calculates a distance-weighted voting descriptor for every superpixel, which is used to reduce false negative and false positive superpixel classifications.

Gaussian Mixture Model (GMM) method

This method is based on the assumption that the densities of the superpixels form two Gaussians. Using the Expectation Maximization algorithm (given that the high density superpixels have high probability to be foreground and the lower density superpixels are most probably background) we can estimate the parameters of the two Gaussian distributions for both densities (μ and σ). Once the parameters for every distribution have been estimated, the decision boundary that minimizes the probability of error can be calculated in a straightforward way. Figure 6.4 shows one of the examples of the resulting probability density function (pdf) for the superpixel densities of one of the synthetic point clouds. Examples of the final segmentation result obtained using this method is shown in Figure 6.5 and Figure 6.6. Finally, Figure 6.7 shows a zoomed view for one of the full segmentation results of both Yao and SEEDS algorithms. The figure shows the poor conformity to object boundaries and the irregularity of the resulting superpixels.

The decision boundary that minimizes the probability of error can be determined by the discriminant functions g_i: $g_i(x) = \ln[p(x/\omega_i)] + \ln[p(\omega_i)]$ given that $p(x/\omega_i) \sim N(\mu_i, \Sigma_i)$, so the general decision boundary can be found by: $g_i(x) = -\frac{1}{2}[(x-\mu_i)^T\Sigma_i^{-1}(x-\mu_i)] -$

$\frac{1}{2}\ln|\Sigma_i| + \ln[p(\omega_i)]$

6.2.7 Calculating the best number of superpixels

The number of superpixels used for point cloud segmentation is an important parameter because it affects the final segmentation results. It is related to the relative size of the foreground objects in the point cloud. Small objects require small superpixels (which means higher number of superpixels) in order to be segmented correctly. In LiDAR point clouds we propose a simple method to calculate the number of superpixels that produces a good segmentation. This method is based on the prior knowledge of the average tree height (which is roughly 80m to 100m). The number of superpixels np can be calculated by $np = (\frac{\max(y)-\min(y)}{h}) \times (\frac{\max(x)-\min(x)}{n})$ where x and y are the coordinates of points, h is the average tree height, and n is the required width for the superpixel.

6.2.8 Surface finding using superpixels

After performing a superpixel segmentation for the LiDAR point cloud, the goal is to estimate the ground and canopy surfaces accurately. We extend this work by proposing a method to estimate ground and canopy surfaces using superpixel segmentation.

After the full segmentation step, the algorithm uses the filtered superpixels' external boundaries as initial surfaces. The external boundaries are those boundaries that are not shared by more than one superpixel. Any other boundary that is shared by more than one superpixel is removed. These external boundaries are connected together to form an initial surface estimation. The algorithm then proceeds to enhance this initial estimation using the LOWESS technique as explained in the previous chapter. We assessed the accuracy of this technique as compared to the algorithm explained in the previous chapter in the experiments section.

6.2.9 ACWE approach vs. superpixels approach

In this dissertation we proposed two approaches for surface finding from photon counting LiDAR point clouds (ACWE and PCDS). Figure 6.8 shows a diagram that explains the full framework of the algorithms introduced in this. According to our experiments (see Figure 7.6(c)), ACWE showed good surface estimation results in many cases but it suffers a drawback in some cases. ACWE approach's performance degrade clearly in cases in which there is complex terrain with lots of bare-ground spaces, high noise density and very occluded ground under dense canopy. However, in these complex and challenging cases the PCDS approach showed better results as shown in the next chapter.

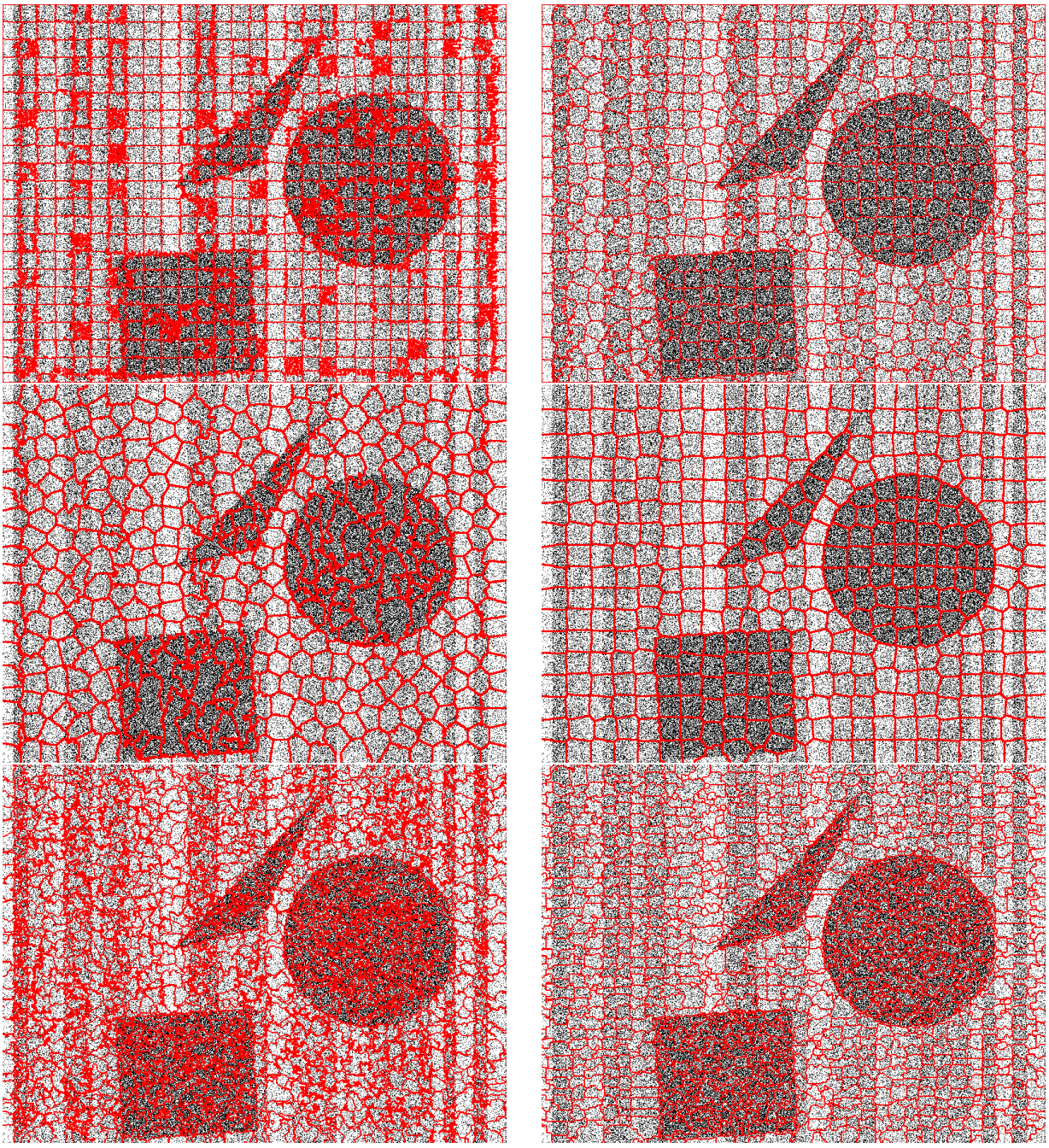


Figure 6.2: Example results of original Yao, SLIC, and SEEDS algorithms from top to bottom respectively (left column) and their PCDS counterparts; PCDS-Yao, PCDS-SLIC and PCDS-SEEDS for superpixel segmentation of a synthetic point cloud. Note that the superpixel boundaries in Yao, SLIC and SEEDS tend to be long and irregular, and exhibit poor conformity to object boundaries. However, the superpixel boundaries are much more regular for PCDS-Yao and PCDS-SLIC than in the traditional algorithms, and they show good conformity to object boundary.

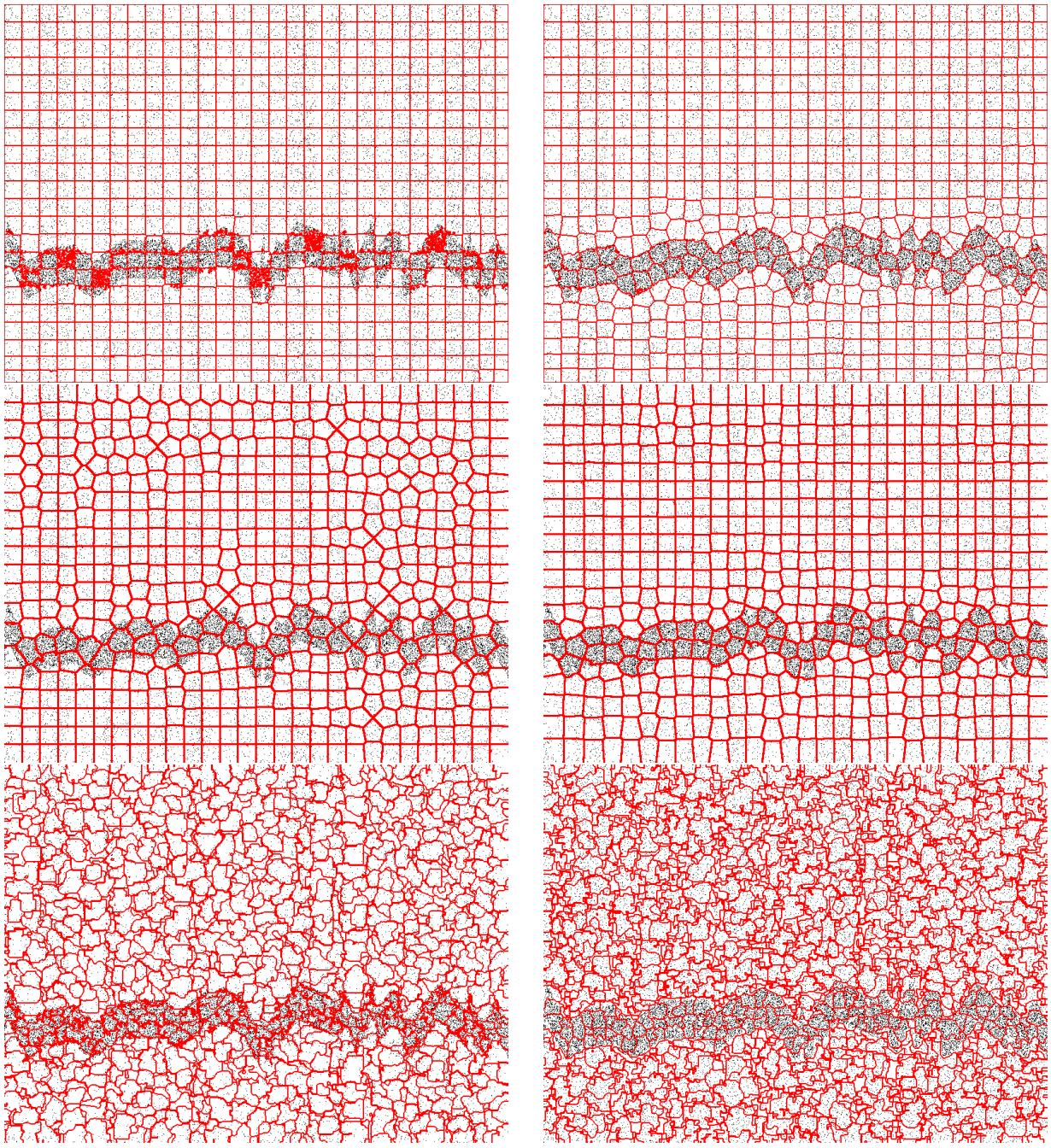


Figure 6.3: Example results of original Yao, SLIC, and SEEDS algorithms from top to bottom respectively (left column) and their PCDS counterparts; PCDS-Yao, PCDS-SLIC and PCDS-SEEDS for superpixel segmentation of a synthetic LiDAR-like point cloud. Note that the superpixel boundaries in Yao, SLIC and SEEDS tend to be long and irregular, and exhibit poor conformity to object boundaries. However, the superpixel boundaries are much more regular for PCDS-Yao and PCDS-SLIC than in the traditional algorithms, and they show good conformity to object boundary.

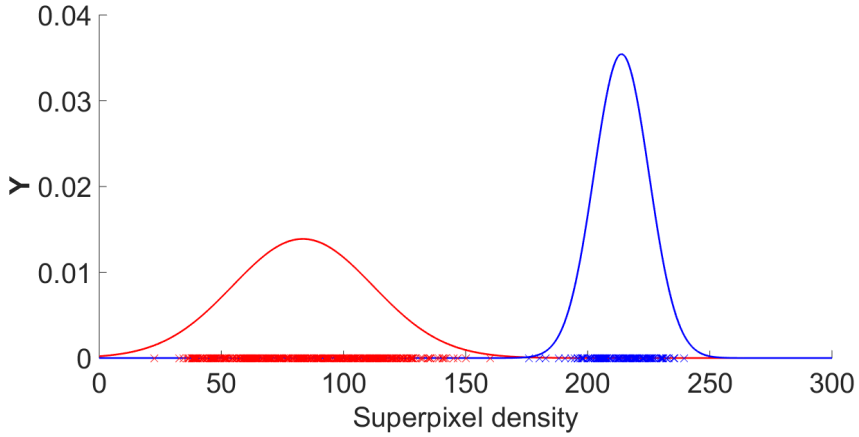


Figure 6.4: Example of the GMM produced for one of the synthetic point clouds.

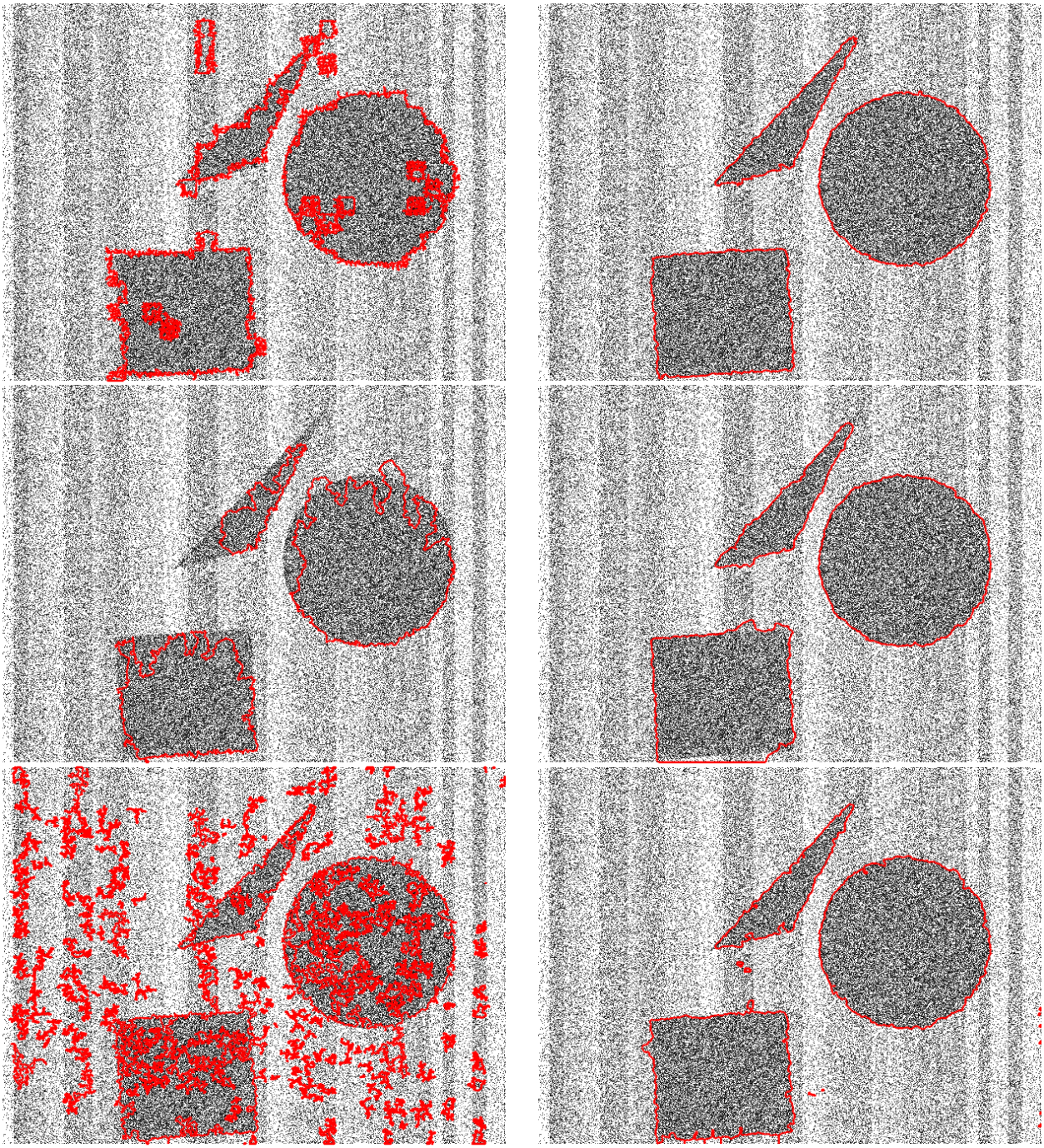


Figure 6.5: Example results of full segmentation for a primitive shapes point cloud using the original Yao, SLIC, and SEEDS algorithms from top to bottom respectively (left column) and their PCDS counterparts; PCDS-Yao, PCDS-SLIC and PCDS-SEEDS (right column). Note that full segmentation of the PCDS algorithms produces boundaries that have better conformity to object boundary.

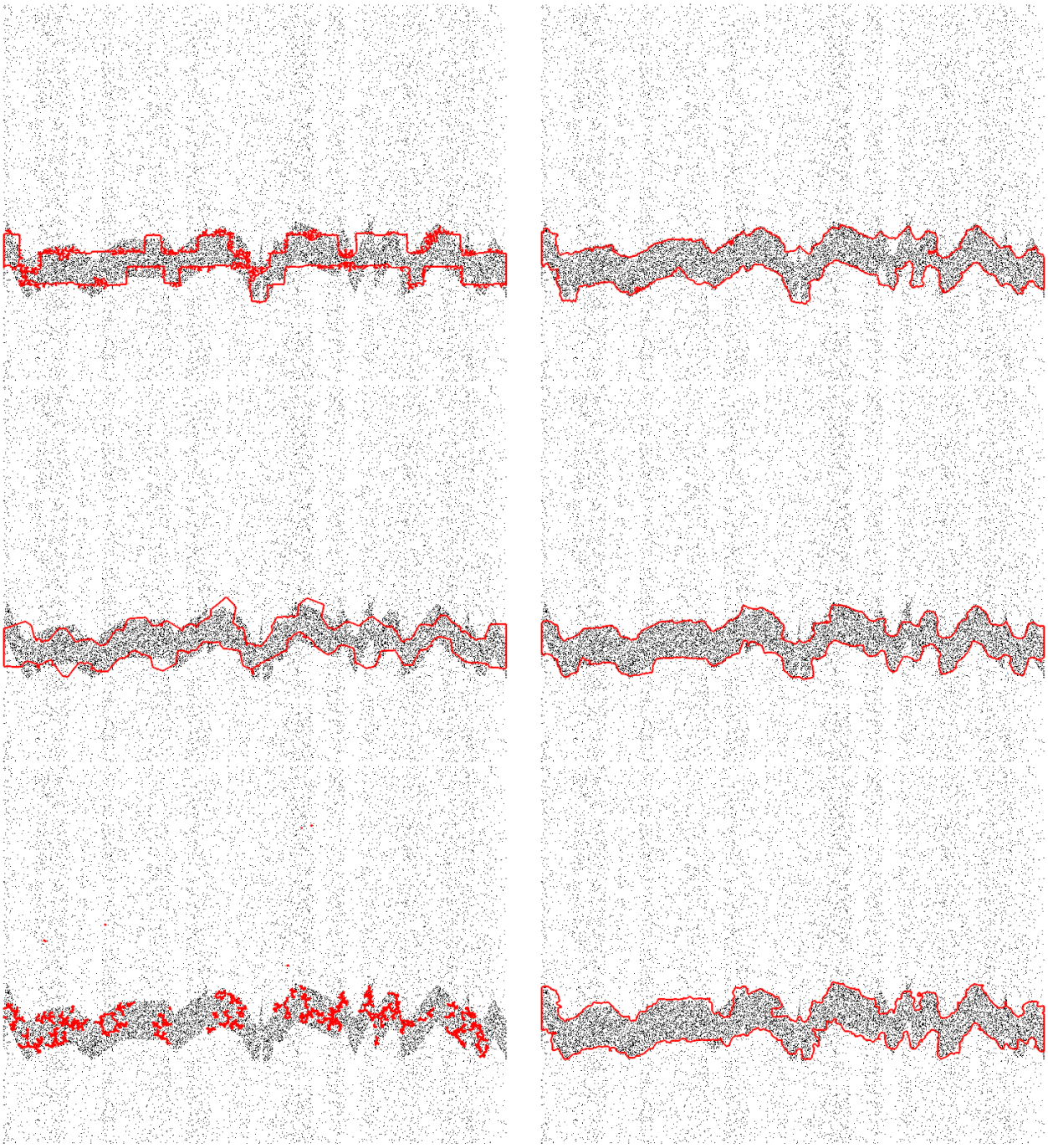


Figure 6.6: Example results of full segmentation for a LiDAR-like point cloud using the original Yao, SLIC, and SEEDS algorithms from top to bottom respectively (left column) and their PCDS counterparts; PCDS-Yao, PCDS-SLIC and PCDS-SEEDS (right column). Note that full segmentation of the PCDS algorithms produces boundaries that have better conformity to object boundary.

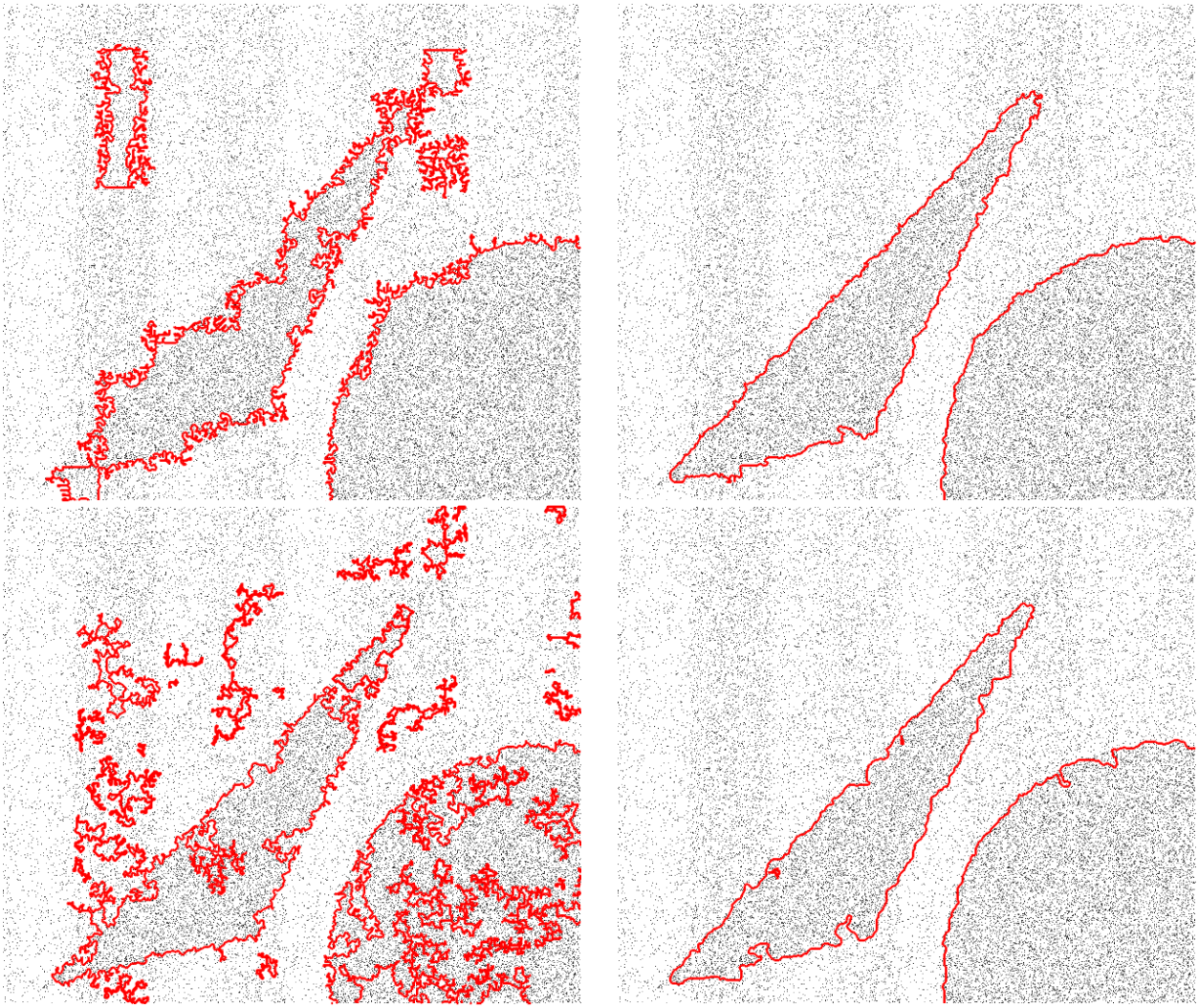


Figure 6.7: A zoomed view for the full segmentation for a primitive shapes point cloud using the original Yao (top) and SEEDS (bottom) algorithms in the left column and their PCDS counterparts in the right column. Note that the traditional state-of-art superpixel segmentation algorithms produce long and irregular boundaries that have poor conformity to object boundaries and the proposed PCDS algorithms show much regular boundaries and much more conformity to object boundaries.

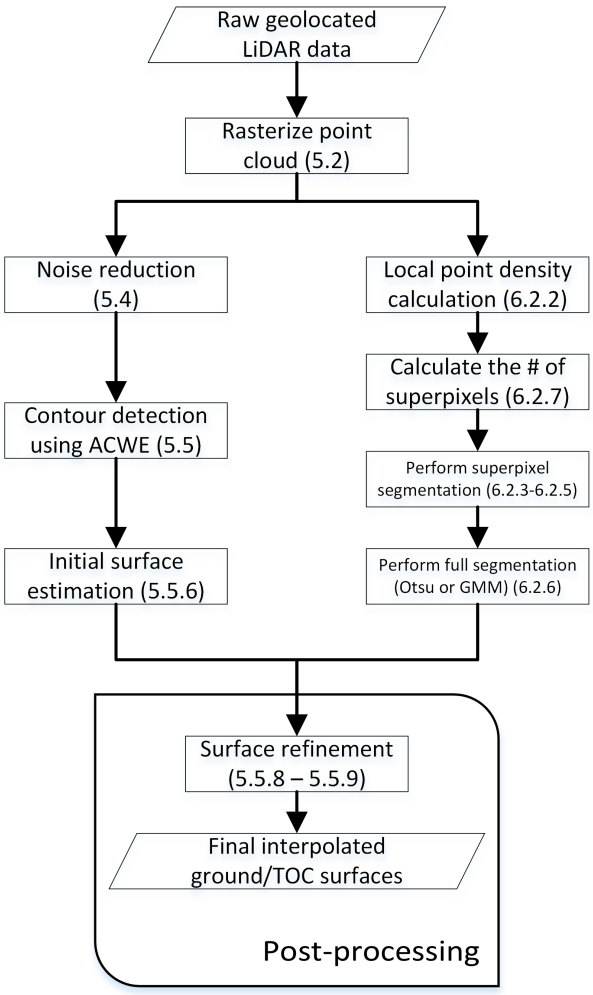


Figure 6.8: A diagram representing the full framework of the surface estimation algorithms introduced in this dissertation. The numbers between brackets represent the section numbers.

Chapter 7

Experimental Results

All the algorithms explained in this dissertation were implemented using MATLAB and C++. The experiments were performed on all real and synthetic types of noisy point clouds explained earlier in the dissertation. However, we have two separate experiments. One experiment is to evaluate the accuracy of surface estimation algorithms using the proposed modified ACWE algorithm and the proposed superpixel based algorithm. The other experiment is to evaluate the performance of the point cloud segmentation algorithm using ACWE and superpixel based algorithms (PCDS-Yao, PCDS-SLIC and PCDS-SEEDS).

7.1 Evaluating surface finding algorithm using ACWE

In order to show the difficulty for standard algorithms to find surfaces in photon-counting LiDAR profiles, we performed an experiment to use one of the well known filtering algorithms that has been widely used for scanning LiDAR data [108].

In this part, we first simulate a three-dimensional point cloud around the original transect. Here, we try to reproduce the point cloud that would have resulted if a scanning LiDAR instrument was used, rather than the profiling one used. This is done by taking advantage of the relative planar symmetry of forest vegetation. We replicate each point in the original transect with slightly jittered horizontal and vertical coordinates (that is x, y and z values). The jittering is done following a normal distribution using mean zero and standard deviation two meters. Each point is replicated 50 times, to give a new transect of width 4 meters (see figure 7.1). After this, we ran a standard implementation of an adaptive triangulated irregular network (TIN) based ground filtering algorithm (Axelsson, 2000). This was attractive because of its overall robustness and its efficacy over forested regions. We used an initial seeding step size of 5 meters.

The results using the simulated 3D point cloud can be seen in Figure 7.2. We notice that the

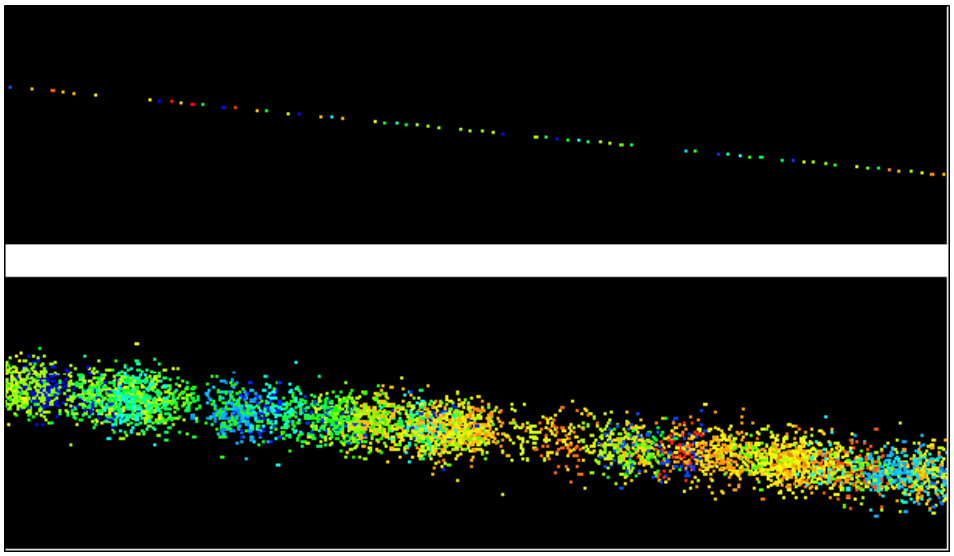


Figure 7.1: The upper portion shows the top view of a typical MABEL transect, with profiling lidar data. The lower part shows the top view of a part of the same transect, but with similated scanning lidar data. The bottom transect is 4 meters wide, and can be processed by ground extraction algorithms developed for three-dimensional data.

ground points that have been detected are below the actual ground surface. It is clear from this figure that the ground points in the simulated cloud follow the lower envelope of the point cloud which is echo noise under the actual ground. This indicates that the traditional surface finding algorithms are not suitable for this type of data, and other noise-robust algorithms are needed.

In order assess the applicability and accuracy of the proposed ACWE algorithm a MATLAB program was implemented and tested. Experiments were performed using three types of photon-counting LiDAR point clouds (see section 4.1). The dataset used in these experiments includes transects from Sigma Space MPL, MABEL and MATLAS point clouds. This dataset includes transects from diverse locations in USA including profiles for different forests, different acquisition times, and multiple simulations to match the expected ATLAS data stream. Figure 4.1 shows the study area of the transects included in this dissertation.

Two types of ground truth were provided for this dataset. The MPL dataset was provided with a flag that indicates the signal/noise classification for every photon in the transect. In this dataset we analyze the accuracy of our algorithm according to this signal/noise classification. For this purpose we used two classification accuracy measurements.

The first accuracy measure is the harmonic mean of recall (R) and precision (P), which is commonly referred to as the F-measure. The recall and precision indicators have been used for evaluation of image-retrieval and image-segmentation techniques [109], and are formally

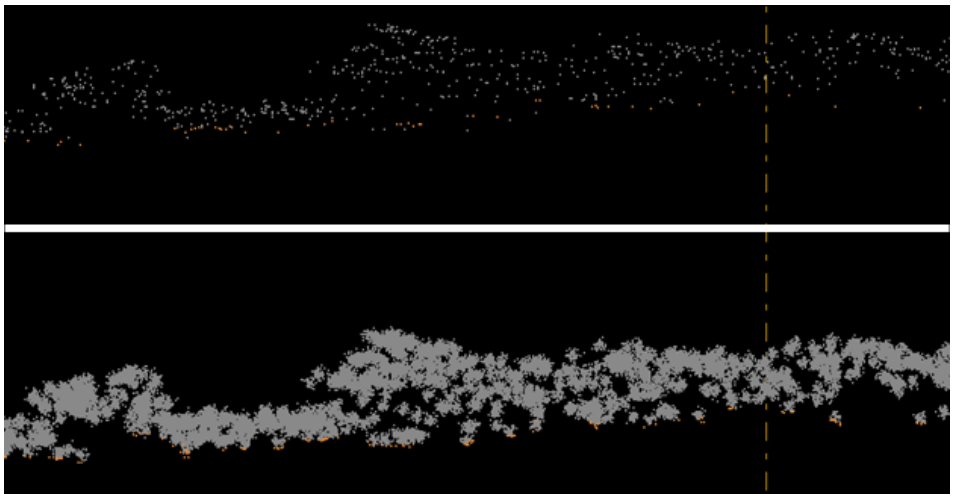


Figure 7.2: A side view of the ground classification result using simulated 3D transect data. The top part shows ground classification (ground points in orange, non-ground in grey) when using the level set algorithm. The bottom part shows the ground classification when using an adaptive TIN-based ground filtering algorithm.

defined as

$$R = \frac{TP}{TP + FN}, \qquad P = \frac{TP}{TP + FP} \tag{7.1}$$

where TP, FP, and FN stand for the numbers of true positives, false positives, and false negatives, respectively. Precision and recall values are between 0 and 1. The F-measure can be calculated as follows:

$$F = \frac{2PR}{P + R} \tag{7.2}$$

where $0 \leq F \leq 1$.

The second accuracy measure is the Matthews Correlation Coefficient (MCC) [110]. The MCC is just a particular application of the Pearson correlation coefficient to a confusion table. MCC can be calculated as follows:

$$MCC = \frac{TP \times TN - FP \times FN}{\sqrt{(TP + FP)(TP + FN)(TN + FP)(TN + FN)}} \tag{7.3}$$

Figure 7.4 and Figure 7.5 show the results of both F-measure and MCC for MPL data. As shown in these figures, our proposed ACWE algorithm produces good signal/noise classification with F-measure values ranged from 0.65 to 0.98 and MCC values ranged from 0.55 to 0.86 for the 15 MPL lines considered here. The figure also shows that our classification accuracy is affected by the amount of noise. One reason for the degraded performance in some few cases is that many of the noise points present in these transects exist within the canopy, which makes them impossible to be distinguished from the actual signal. Figure 7.3

shows an example for one of the MPL transects showing its ground truth classification. As shown in the figure some noise points cannot be distinguished from the signal points. In this figure the no-data section indicates the presence of a cloud. Our algorithms in this case bridge the gaps that result from such anomalies.

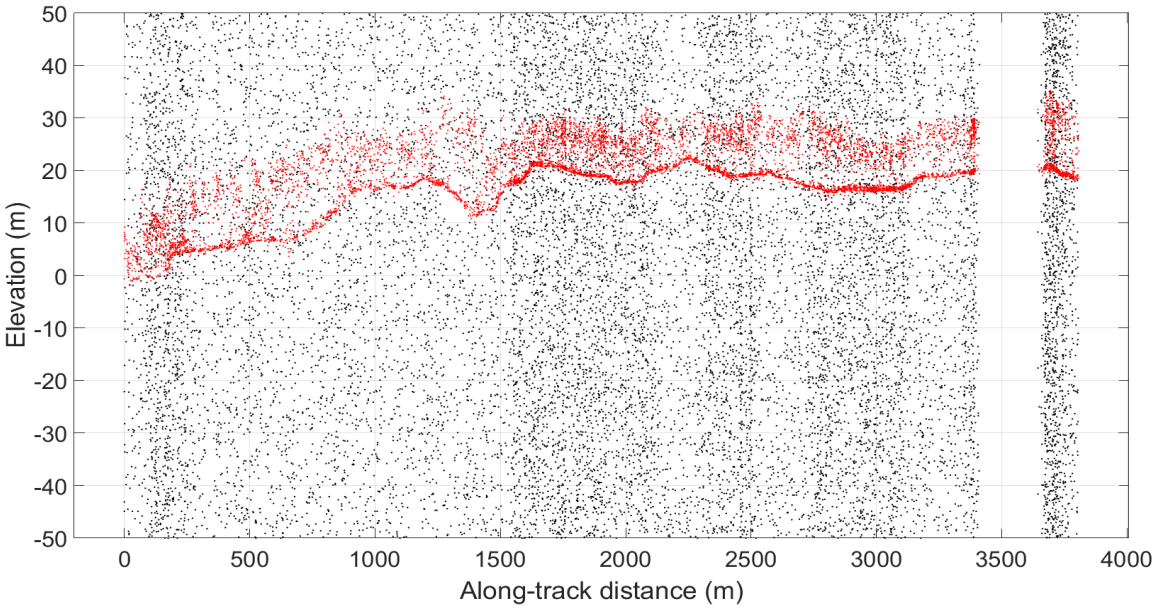


Figure 7.3: Example MPL ground truth for Cedar-2 transect collected over the Silas Little and Cedar Bridge towers, in the Pine Barrens of southern New Jersey. Red represents signal and black represents noise. Note that many noise points are mixed withing the vegetation region. Our algorithms do not attempt to distinguish those points.

To assess the accuracy of the proposed ACWE surface finding algorithm quantitatively, we utilized several G-LiHT datasets that intersect MABEL and MATLAS flight lines. G-LiHT is a small-footprint, discrete-return, scanning airborne LiDAR. Thus, the ability to detect forest canopy height and to calculate forest biomass along the G-LiHT flight lines is well understood, and maps of canopy height along the flight transects have already been generated by the G-LiHT science team [111]. We have used the G-LiHT data to validate our surface estimation algorithm and to evaluate our ability to derive forest biomass from simulated ICESat-2 across a forest ecosystem gradient, as described below.

The G-LiHT files were co-registered manually to the MABEL and MATLAS data by researchers at Colorado State University, so that the G-LiHT data could serve as a ground-truth reference. In these experiments, our algorithm detected ground and top-of-canopy curves in the dataset automatically, and then a comparison was made between the corresponding profiles to estimate the algorithm accuracy. Two different measures of accuracy were used. The first measure is the Root Mean Squared Error (RMSE) for both surfaces. The RMSE was calculated by comparing the estimated surface point elevation to its corre-

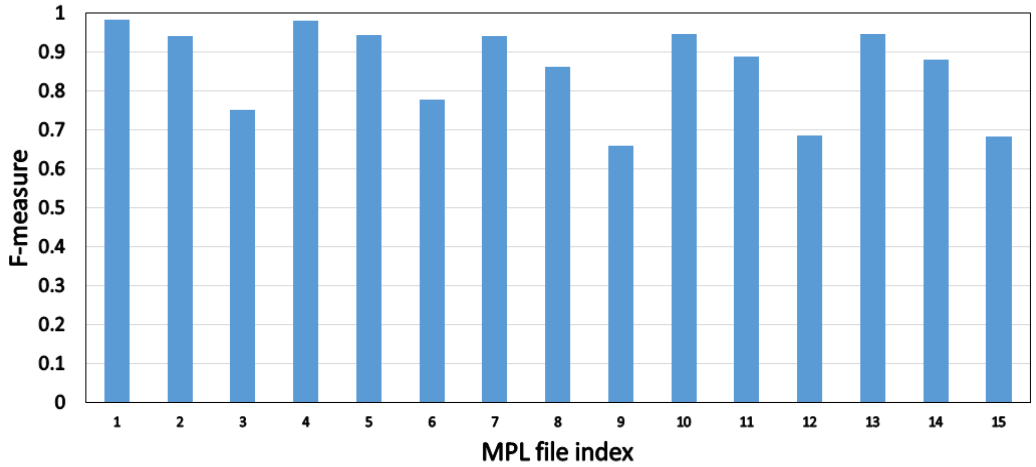


Figure 7.4: F-measure classification accuracy results of ACWE algorithm for MPL dataset. File index locations are explained in Chapter 4.1.

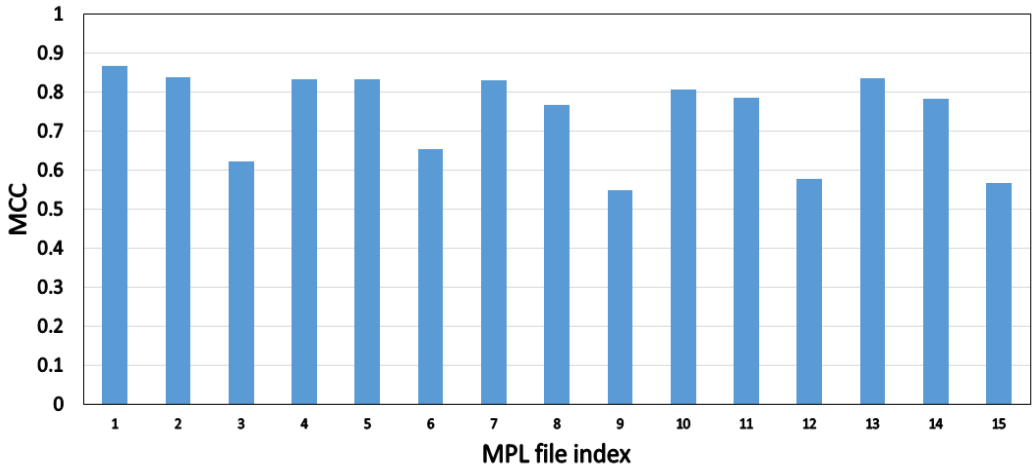


Figure 7.5: MCC classification accuracy results of ACWE algorithm for MPL dataset. File index locations are explained in Chapter 4.1.

sponding G-LiHT reference. The second measure is the correlation coefficient between the estimated surface and the corresponding G-LiHT reference.

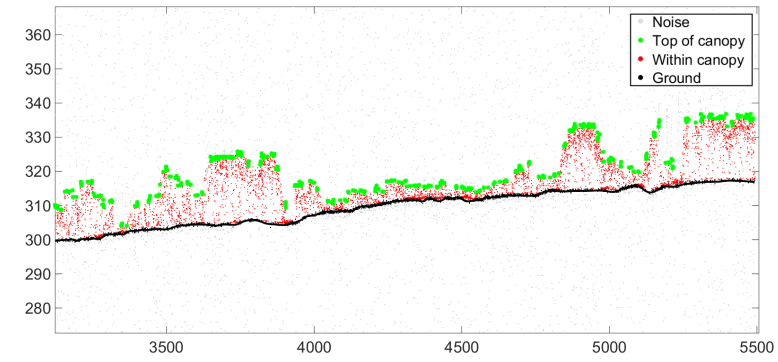
Qualitatively, Figure 7.6 show examples for our ACWE algorithm results for three MABEL transects. This example demonstrates that the proposed ACWE algorithm can estimate ground and top of canopy reasonably well despite the large amount of noise and the complexity of terrain. However, the algorithm performs less accurately in the situations in which complex terrain is combined with partially occluded ground surface. In the next section we will show that the superpixel-based algorithm performs much better in these cases.

Figure 7.7 shows the correlation plot for the DTM and the DSM for the transect shown in Figure 7.6a. As shown in the figure there is a strong correlation (0.99 for both DTM and DSM) between the estimated surfaces and the reference data.

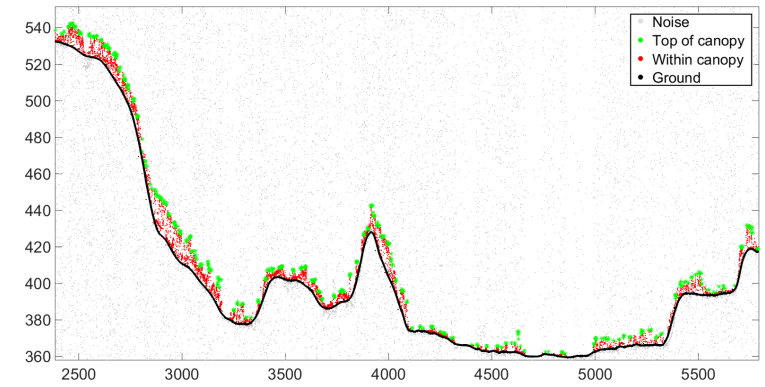
Quantitatively, Figure 7.8 shows the RMSE for DTM (ground) and DSM (top of canopy) for 25 MABEL transects taken from southeastern Alaska in July, 2014. The average and the median DTM RMSE are 1.1 meters and 0.78 meters, respectively, while the average and median DSM RMSE are 4 meters and 3.4 meters, respectively. As shown in the figure, most of DTM RMSE values are below 1.5 meters except for very few cases which will be discussed later in detail. The DSM RMSE values are much higher because of a bias in ground truth that will be discussed later in the next subsection. Similarly, Figure 7.9 shows the correlation coefficient for DTM (ground) and DSM (top of canopy) for the same transects. The correlation coefficient shows a very strong correlation between the estimated surfaces and the reference data with a correlation coefficient over 0.98 for almost all the cases with one or two exceptions. The average and median correlation coefficient for the DTM is 0.99 while the average and median correlation coefficient for the DSM is 0.99 and 0.97, respectively.

Figure 7.10 shows the RMSE and correlation coefficient for DTM estimated from 50 MABEL transects taken in September, 2012 over diverse locations and different echo systems across the United States. The average and median DTM RMSE is 3.2 meters and 3.3 meter, respectively, which is much higher than MABEL 2014 transects for two reasons. The first reason is that the ground signal in MABEL 2012 transects is weaker than the ground signal in MABEL 2014 transects. The second reason is that the G-LiHT ground truth sometimes appears horizontally and/or vertically shifted from the actual MABEL locations. This problem will be discussed in detail in the next subsection. However, the correlation coefficient shows a strong correlation between the estimated surfaces and the reference data with a correlation coefficient over 0.9 for the majority of the cases. The average and median correlation coefficient for the DTM are 0.91 and 0.95, respectively.

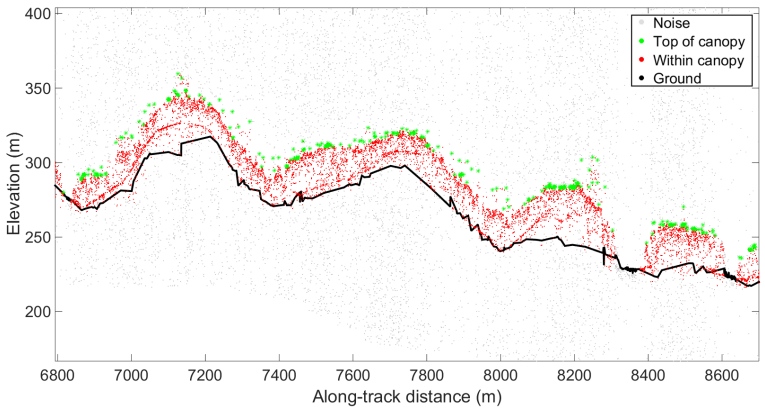
Figure 7.11 shows the DTM RMSE for 20 MATLAS transects, which were simulated by Dave Harding from a few MABEL 2012 transects. The files were simulated to represent both the strong and weak beams expected from ATLAS. Each file is simulated with two noise densities which represent different sun angles. The average and median DTM RMSE are 3.1 meters and 2.7 meters, which is much higher than MABEL 2014 transects because of the two reasons explained earlier. The correlation coefficient shows a strong correlation



(a)

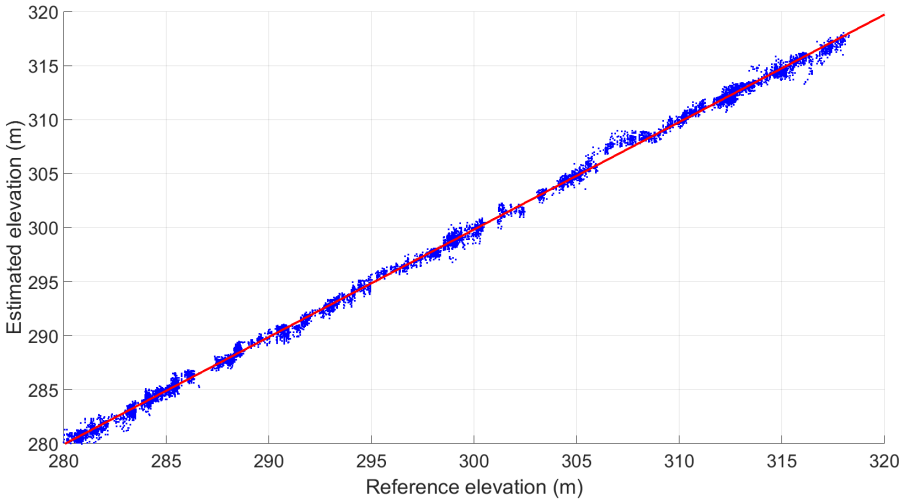


(b)

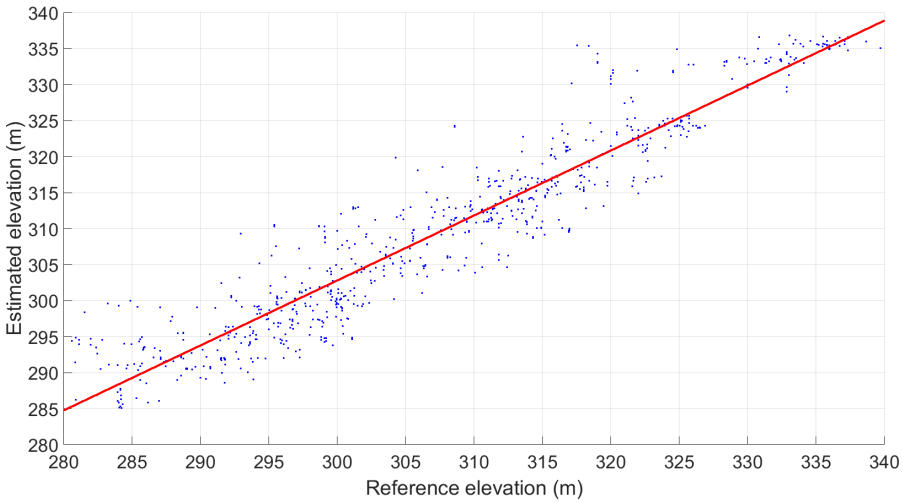


(c)

Figure 7.6: Example results of ACWE-based surface estimation algorithm on MABEL 2014 (a),(b) and MABEL 2012 (c) transects.

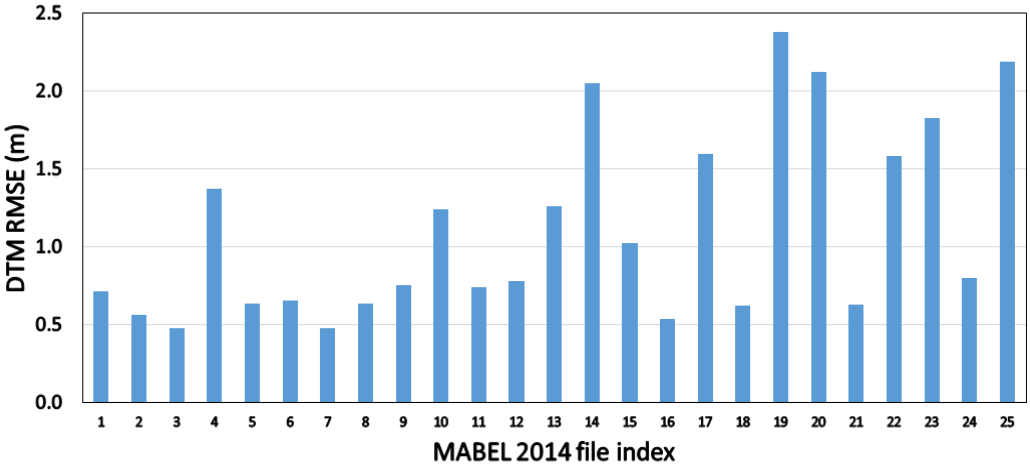


(a)

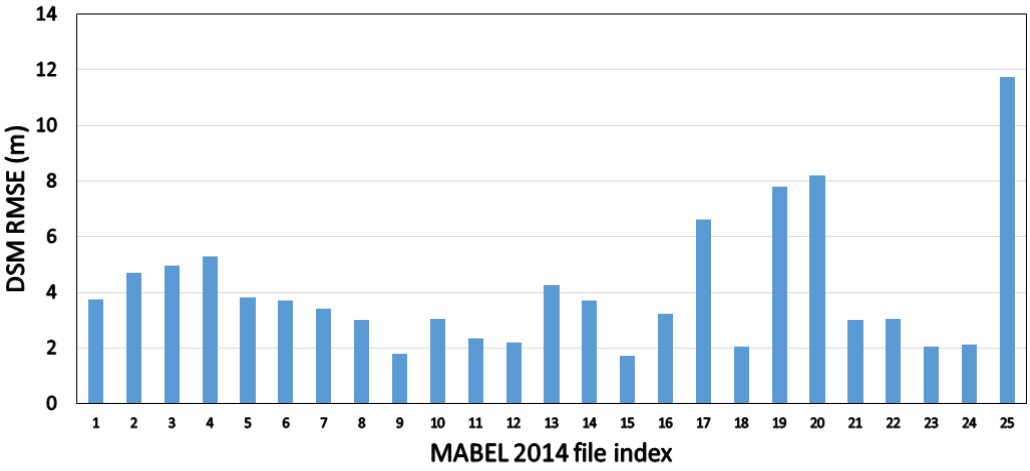


(b)

Figure 7.7: Example correlation plot for the DTM (a) and DSM (b) of ACWE-based surface estimation algorithm for the transect shown in Figure 7.6a.

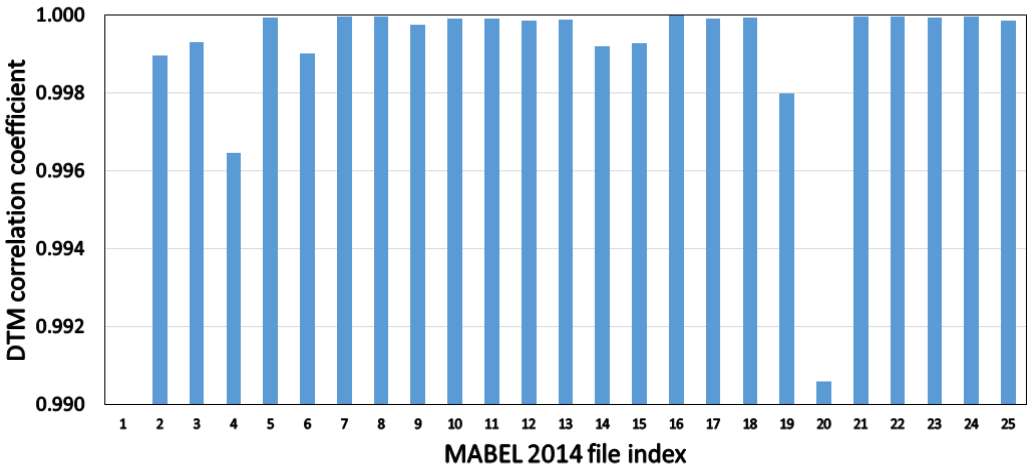


(a)

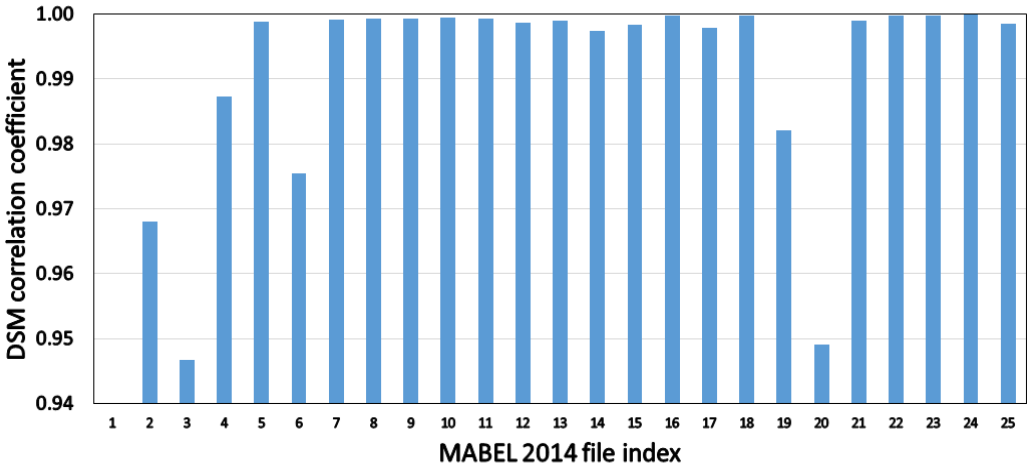


(b)

Figure 7.8: RMSE for both DTM (a) and DSM (b) surfaces extracted from MABEL 2014 Alaska transects.

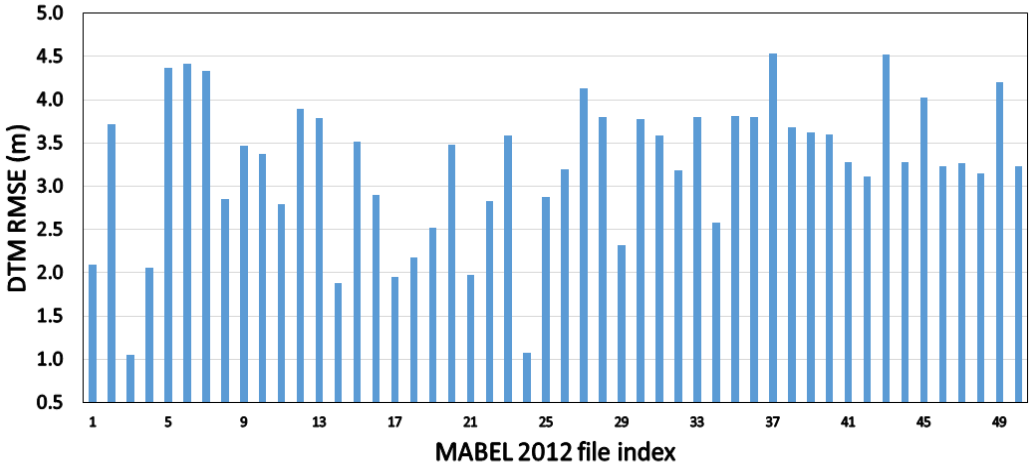


(a)

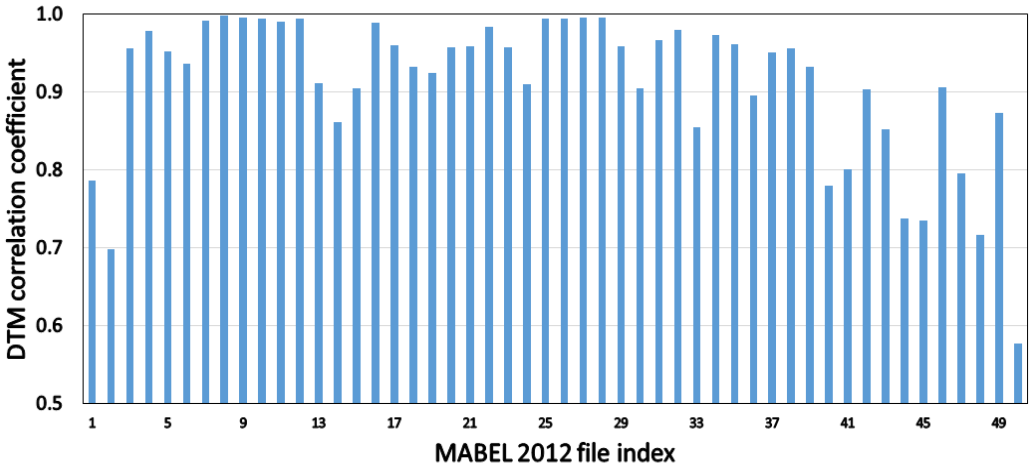


(b)

Figure 7.9: Correlation coefficient for both DTM (a) and DSM (b) surfaces extracted from MABEL 2014 Alaska transects.



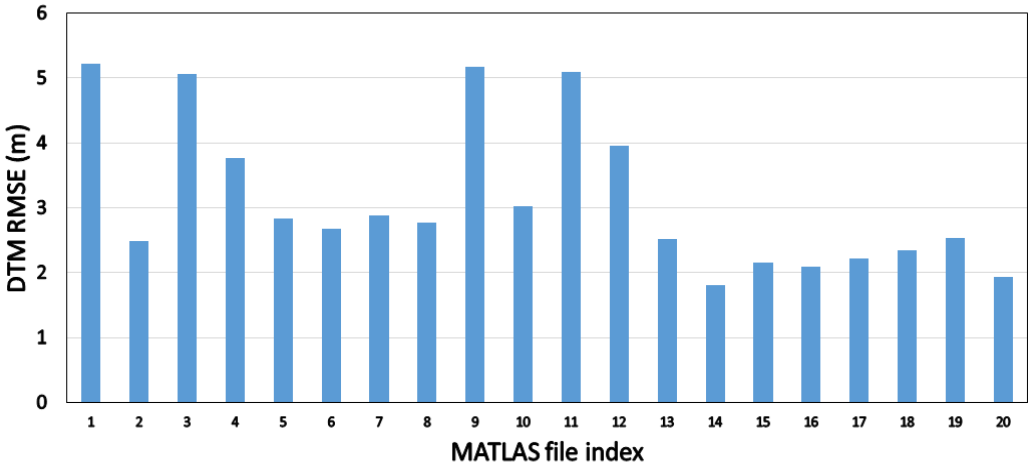
(a)



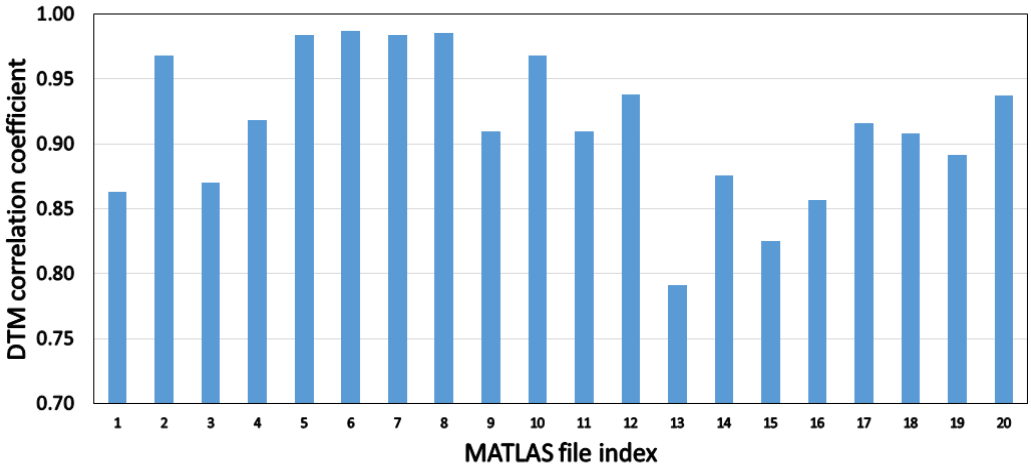
(b)

Figure 7.10: RMSE (a) and correlation coefficient (b) for DTM surface extracted from MABEL 2012 transects.

between the estimated surfaces and the reference data with a correlation coefficient over 0.9 for the majority of the cases. The average and median correlation coefficient for the DTM is 0.91.



(a)



(b)

Figure 7.11: RMSE (a) and correlation coefficient (b) for DTM surface extracted from MATLAS transects.

We did not perform RMSE and correlation coefficient calculation for DSM for the MABEL 2012 transects and the MATLAS transects because of the bias problem in the ground truth, which is much more severe in these cases and does not reflect the actual accuracy of the proposed ACWE algorithm. This issue will be discussed also in the next subsection.

7.2 Issues with G-LiHT DTM and DSM as ground truth

In this dissertation we use the DTM and DSM extracted from manually co-registered G-LiHT flight lines as ground truth for the co-located MABEL and MATLAS transects. Although this has provided us with an acceptable means to evaluate the algorithm quantitatively, sometimes there are obvious problems with the G-LiHT reference data which will contribute to the calculated errors, but do not arise from our surface finding algorithms. It was noticed that some G-LiHT reference data have an elevation bias up to ∼4 meters, and some others do not have this bias. This unsystematic bias is hard to resolve. Moreover, in some cases there is a vertical bias that might contribute to the error. Figure 7.12 shows some of these problems.

In some cases the difference of the time of acquisition between MABEL and G-LiHT and/or some disturbances in MABEL geolocation might cause the reference data to represent a slightly different locations in which you can find clipped hills and trees that does (or does not) exist in the reference data. Finally, there is a fundamental difference in the resolution and point density between G-LiHT and MABEL. This difference causes the reference data to show canopy gaps and DSM details that are impossible to be obtained from the MABEL or MATLAS data. In all the experiments in this dissertation we have chosen the transects that shows the best fit to the reference data.

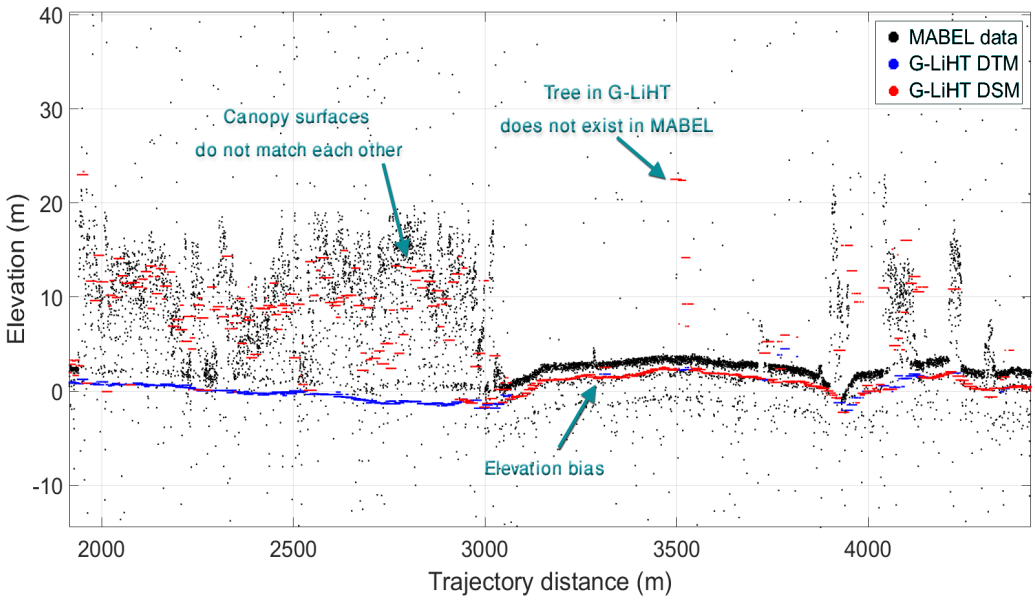


Figure 7.12: Example showing the problems in G-LiHT, which is used as a reference for algorithm evaluation. Black points represent MABEL point cloud, blue points represent G-LiHT DTM, and red points represent G-LiHT DSM.

7.3 Superpixel experiments

The experiments are performed to assess the performance of the superpixel segmentation algorithm. All the results are evaluated on a test set of 400 synthetic point clouds containing primitive shapes, 400 LiDAR-like synthetic point clouds and 15 MPL point clouds. We use standard metrics to evaluate the segmentation performance, as used in most recent superpixel research [57], [88], [89]. These are Undersegmentation Error (UE), Corrected Undersegmentation Error (CUE) and Achievable Segmentation Accuracy (ASA). Moreover, we also used a standard metric for measuring superpixel compactness based on the isoperimetric quotient [112]. Let $S = \{s_1, s_2, \ldots, s_k\}$ be the resulting superpixel segmentation and let $G = \{g_1, \ldots, g_n\}$ be the ground truth segmentation of n segments where g_i is the segment number i. Then the accuracy metrics can be defined as follows.

7.3.1 Undersegmentation Error (UE)

The undersegmentation error measures the conformity of a superpixel boundary such that it do not overlap more than one object.

$$UE(S, G) = \frac{\sum_i \sum_{k:s_k \cap g_i \neq \phi} |s_k - g_i|}{\sum_i |g_i|} \tag{7.4}$$

where $|.|$ indicates the size of a segment.

This metric is not accurate because using this metric, a segmentation based on a rectangular grid outperforms SLIC superpixels [57] as well as other algorithms [88]. Therefore the corrected undersegmentation error metric was proposed in [88], which overcomes this problem.

7.3.2 Corrected Undersegmentation Error (CUE)

In this metric each superpixel is matched to the ground truth segment that has the largest overlap with it. Then, the sum of the number of pixels that lie outside of each ground truth element is calculated:

$$CUE(S, G) = \frac{\sum_j |s_j - \text{argmax}_{g_i \in G} |s_j \cap g_i||}{\sum_i |g_i|} \tag{7.5}$$

7.3.3 Achievable Segmentation Accuracy (ASA)

In this metric also the superpixels are labeled with the label of the ground truth segment which has the largest overlap. It gives the maximum performance when taking superpixels

as units for object segmentation.

$$ASA(S,G) = \frac{\sum_j \max_i |s_j \cap g_i|}{\sum_i |g_i|} \tag{7.6}$$

7.3.4 Superpixel compactness

This metric measures the compactness of the shape of the superpixels [112]. It relates the shape of the superpixel to the shape of the circle which is the most compact shape. The isoperimetric quotient relates the area of a given shape to the area of a circle that has the same perimeter as this shape. It can be calculated as:

$$Q_s = \frac{2\pi A_s}{L_s^2} \tag{7.7}$$

where A_s and L_s are the area and perimeter of a shape s, respectively. Then the compactness metric can be calculated as:

$$CO = \sum_{s_i \in S} Q_s \cdot \frac{|S|}{|I|} \tag{7.8}$$

where S is the superpixel segmentation of image I.

Because our PCDS algorithm can be based on different superpixel algorithms, we use the names PCDS-Yao, PCDS-SLIC, and PCDS-SEEDS to indicate the use of our modified versions of Yao, SLIC, or SEEDS, respectively.

Figure 7.13 and Figure 7.14 show the F-measure and MCC comparison between our variations of PCDS algorithm and several traditional segmentation algorithms including CMF [97], [98], TVD [99], N-cut [83], DBSCAN [100] and revised DBSCAN [101] on the primitive-shape and LiDAR-like synthetic datasets respectively. This figure also includes the results obtained from the ACWE algorithm proposed in chapter 5 [82]. These figures show the comparison between algorithms as a function of SNR.

As shown in the figures, most of the algorithms show better performance as the SNR increases. Also, our proposed PCDS-Yao and PCDS-SLIC algorithms outperforms all of the other algorithms significantly in terms of segmentation accuracy (PCDS-SEEDS algorithm shows relatively poor results in some cases due to the inefficient boundary handling inherited from the original SEEDS algorithm). However, the proposed ACWE algorithm outperforms the superpixel segmentation algorithms in the very high SNR cases and in the very low density signal cases. The PCDS-Yao and PCDS-SLIC show the best performance among other algorithms. It is noted also that, most of the algorithms show relatively low MCC values at very high SNR because most of these point clouds are relatively sparse which causes the error calculated by MCC to increase. Moreover, on the MPL dataset, the PCDS-Yao algorithm outperforms the ACWE algorithm in many cases especially in the higher noise cases as shown in Figure 7.15. The average F-measure score for ACWE is 0.86 and the average

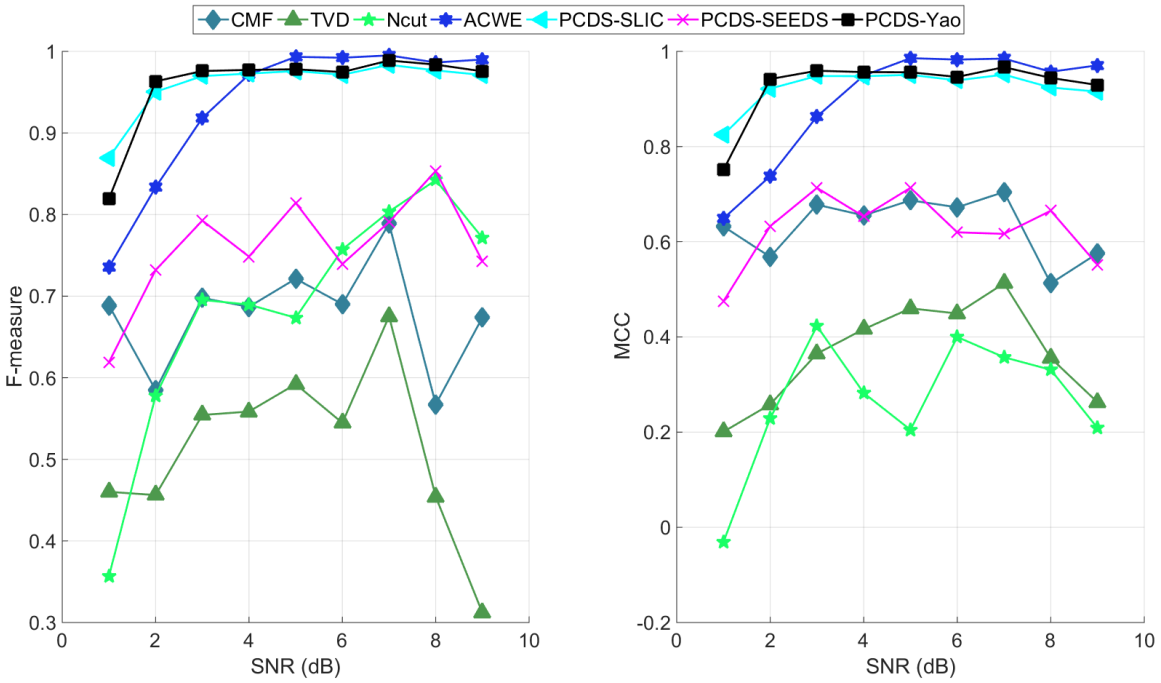


Figure 7.13: F-measure and MCC comparison between the proposed ACWE and PCDS segmentation algorithm, and several other clustering algorithms on the primitive-shape synthetic point clouds.

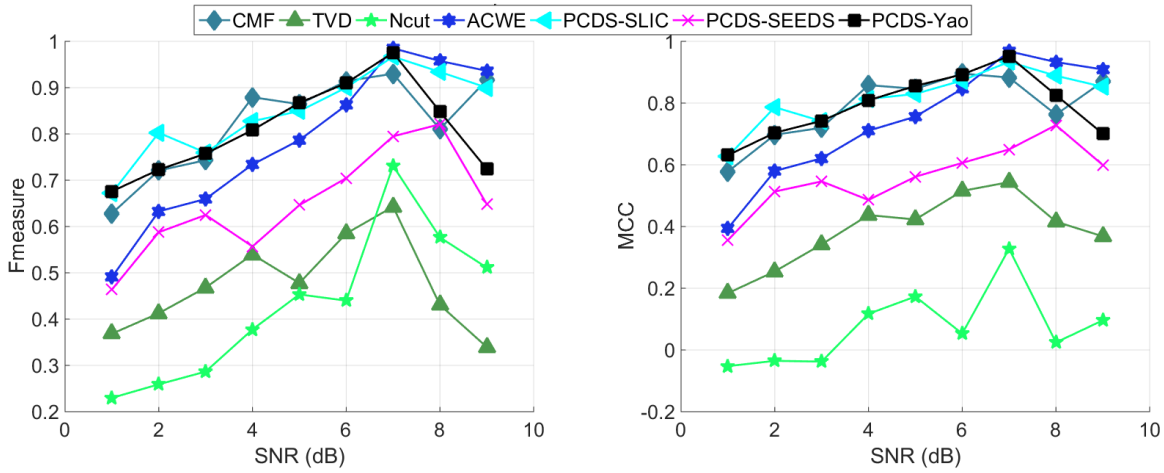


Figure 7.14: F-measure and MCC comparison between the proposed ACWE and PCDS segmentation algorithms, and several other clustering algorithms on LiDAR-like synthetic point clouds.

F-measure score for PCDS-Yao is 0.88. The PCDS-Yao algorithm outperforms ACWE in the cases with higher noise rates (files with indices 3, 6, 9, 12, 15) because this high noise rates surprisingly increases the density of points in the signal points causing the superpixel

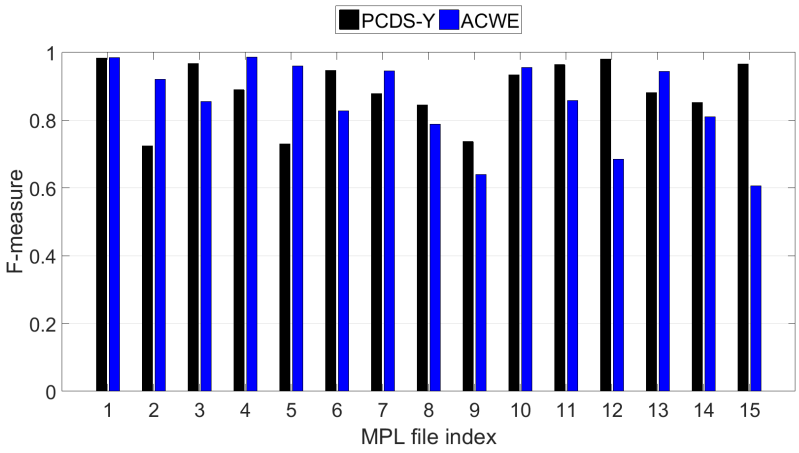


Figure 7.15: F-measure comparison between our PCDS-Yao algorithm and our ACWE algorithm. Our PCDS-Yao technique achieved an average F-measure score that was 21% higher than ACWE. In one case PCDS-Yao achieved an increase of 59% of F-measure score.

boundaries to become more accurate in separating low density points from high density points.

We further investigated the N-cut, DBSCAN and revised DBSCAN algorithms' classification performance as compared to our PCDS-Yao algorithm on the MPL LiDAR dataset. The parameters of these algorithms are adjusted empirically to produce the best performance. The gridding parameter was fixed on 800×600 for this experiment. The result of this experiment is shown in Figure 7.16. As shown in the figure, PCDS-Yao segmentation algorithm outperforms other algorithms in almost all cases, especially in the cases with higher noise rates (files with indices 3, 6, 9, 12, 15).

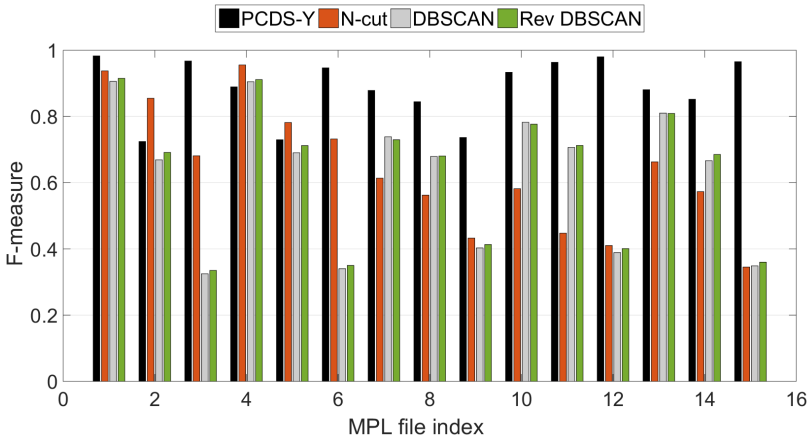


Figure 7.16: F-measure comparison between the proposed PCDS-Yao segmentation algorithm and several other clustering algorithms on MPL lidar point clouds.

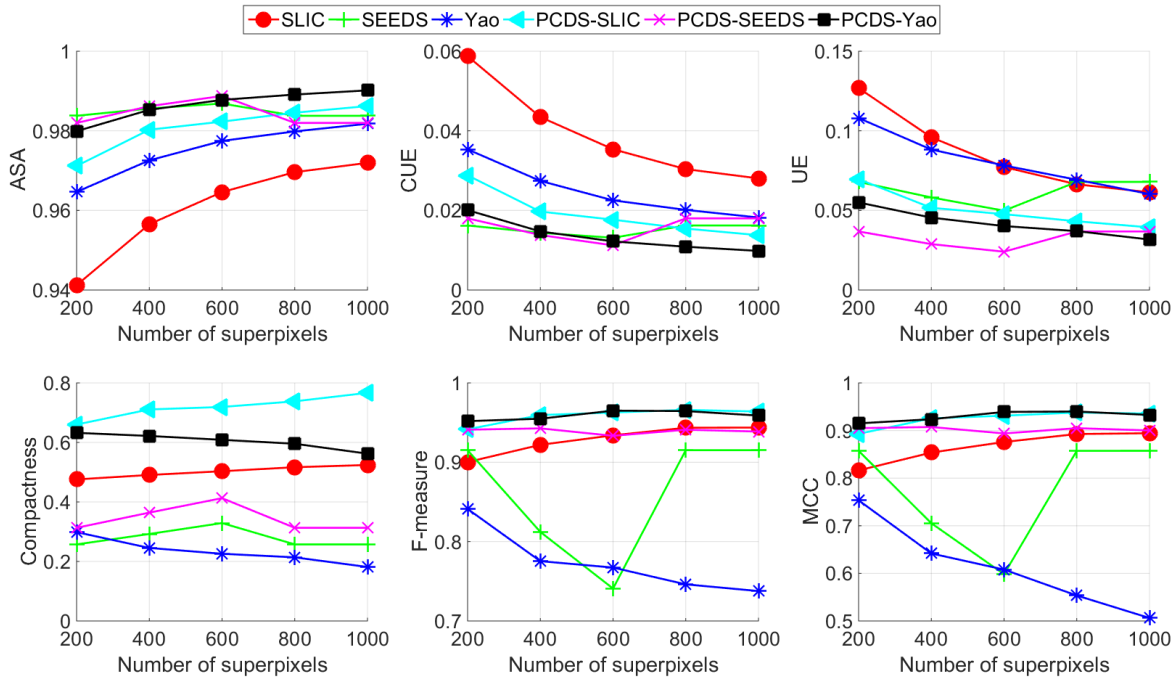


Figure 7.17: Evaluation of the PCDS algorithms and the state-of-the-art on the primitive-shape synthetic dataset in terms of number of superpixels.

Another experiment was performed to assess the enhancement of accuracy gained from using the proposed segmentation algorithms as compared to previous superpixel segmentation algorithms. The experiment compares the standard metrics explained earlier along with F-measure and MCC accuracy of each original algorithm (Yao, SEEDS, and SLIC) to the modified version to assess the accuracy enhancement. The experiments compare the algorithms in terms of the number of superpixels (Figure 7.17 and Figure 7.18), and the SNR (Figure 7.19 and Figure 7.20) for all synthetic cases. For these experiments the higher the values of ASA, compactness, F-measure and MCC the better the algorithm's performance and the lower the values of CUE and UE the better the algorithm's performance.

As shown from these experiments, using the proposed energy function for point-cloud superpixel segmentation leads to a significant improvement in accuracy over the original algorithms. It is also shown in the experiment that changing the number of superpixels has a little effect on the resulting accuracy. Moreover the proposed algorithms are more stable and robust for noise than the previous state-of-the-art techniques.

To analyze the algorithms in greater detail, we performed another experiment to study the algorithms' performance and sensitivity to different algorithm parameters. We also used this analysis to select the parameters that produce better results in further experiments. For the Yao and the PCDS-Yao algorithms, we analyzed the performance on changing three weight parameters: appearance, boundary length and the regularization term. The results of these

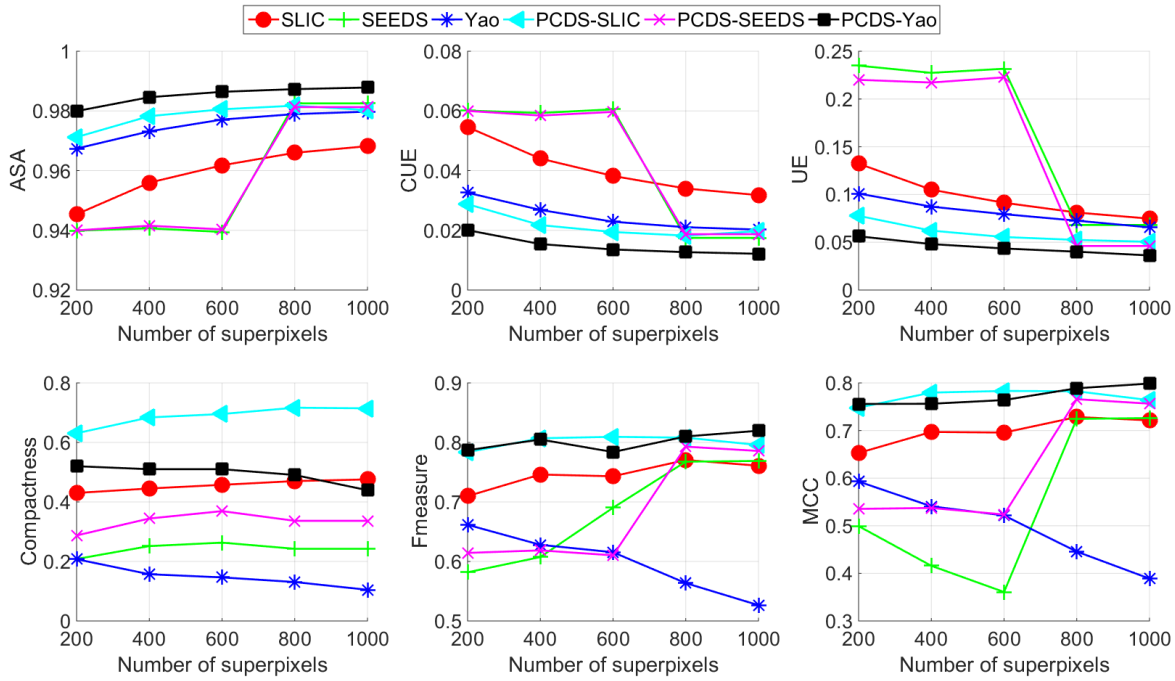


Figure 7.18: Evaluation of the PCDS algorithms and the state-of-the-art on LiDAR-like synthetic dataset in terms of number of superpixels.

experiments are shown in Figure 7.21, Figure 7.22 Figure 7.23, Figure 7.24, Figure 7.25 and Figure 7.26.

It is clear from these figures that the algorithms are stable in regards to the algorithm parameters because changing parameters has a small effect on the algorithm's performance .

For the SLIC and PCDS-SLIC algorithms there was only one parameter to study which is the compactness weight. This weight controls the compactness of the superpixels result from the algorithm.

These figures show that, SLIC and PCDS-SLIC algorithms are not sensitive to the compactness weight parameter. The effect of this parameter on the algorithm performance is minimal.

The results are also very good qualitatively, as shown by the example results in Figure 7.30. This figure shows a full segmentation obtained by PCDS-Yao for point clouds shown in Figure 1.3. These results are substantially better than we have seen for other segmentation and clustering approaches (see Figure 6.5 and Figure 6.6 for some examples). In the LiDAR-like example some peaks and valleys were missed by the segmentation algorithm. These results can be enhanced in the LiDAR dataset by applying the surface refinement approach presented in chapter 5.

The qualitative assessment of the superpixel-based surface estimation shows accurate results

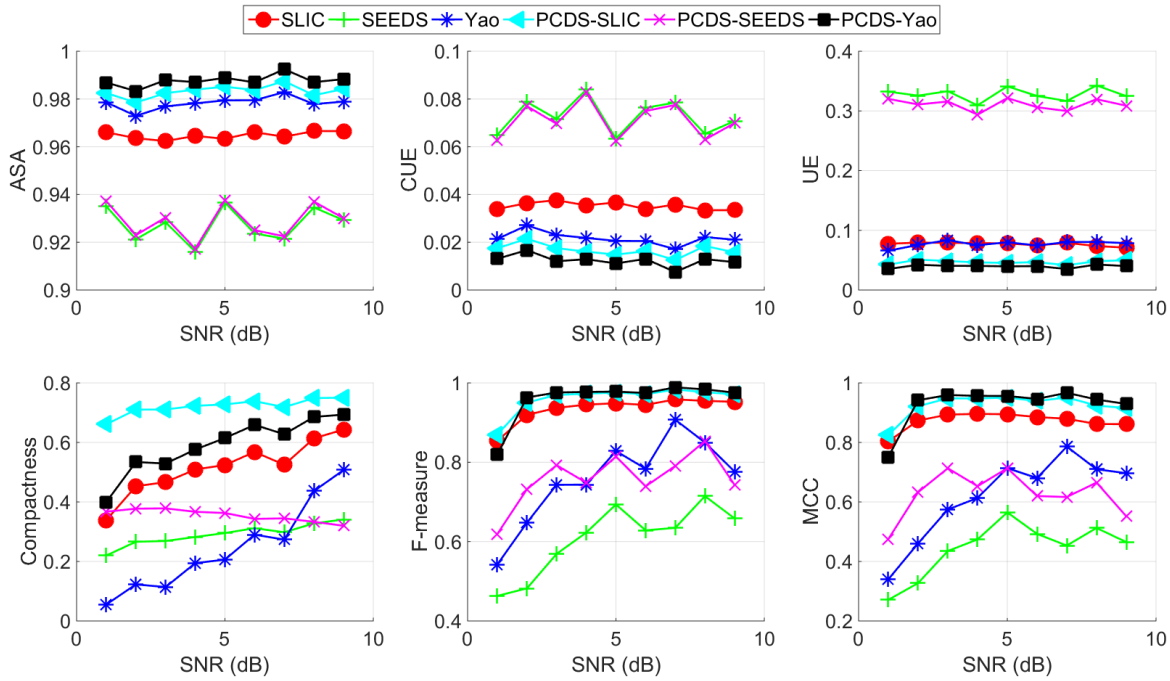


Figure 7.19: Evaluation of the PCDS algorithms and the state-of-the-art on the primitive-shape synthetic dataset in terms of SNR.

as shown in the example results in Figure 7.31. This figure shows the surface estimation results for the transects shown in Figure 7.6. As shown in this figure, the superpixel-based surface estimation shows much better results than ACWE in the cases that combines very high noise density with partially occluded ground surface.

Another experiment was conducted in order to study the effect of changing the size of the cells on the projection plane (explained in section 4.1), which we call the gridding parameter. In this experiment we quantized one of the MPL point clouds onto a gridded plane with array sizes ranging from 800×600 to 300×100, decreasing 100 pixels on each dimension every time. Figure 7.32 shows the results of this experiment. As shown in this figure, the proposed PCDS algorithm is not overly sensitive to this parameter. The performance degrades very slightly with the smaller images because the noise effectively becomes denser and affects the object boundaries.

The final experiment is performed to compare the accuracy and time efficiency of surfaces estimated using ACWE to the accuracy and time efficiency of surfaces estimated using superpixel segmentation. Figure 7.33 shows the RMSE (top) and execution time (bottom) for ground surface estimated by both algorithms. The experiment was conducted on an Intel Core i5 processor (2.4 GHz). The algorithms of surface finding were implemented using Matlab. However, the superpixel segmentation step is implemented using C++.

As shown in the figure, the RMSE difference between the two algorithms is relatively small.

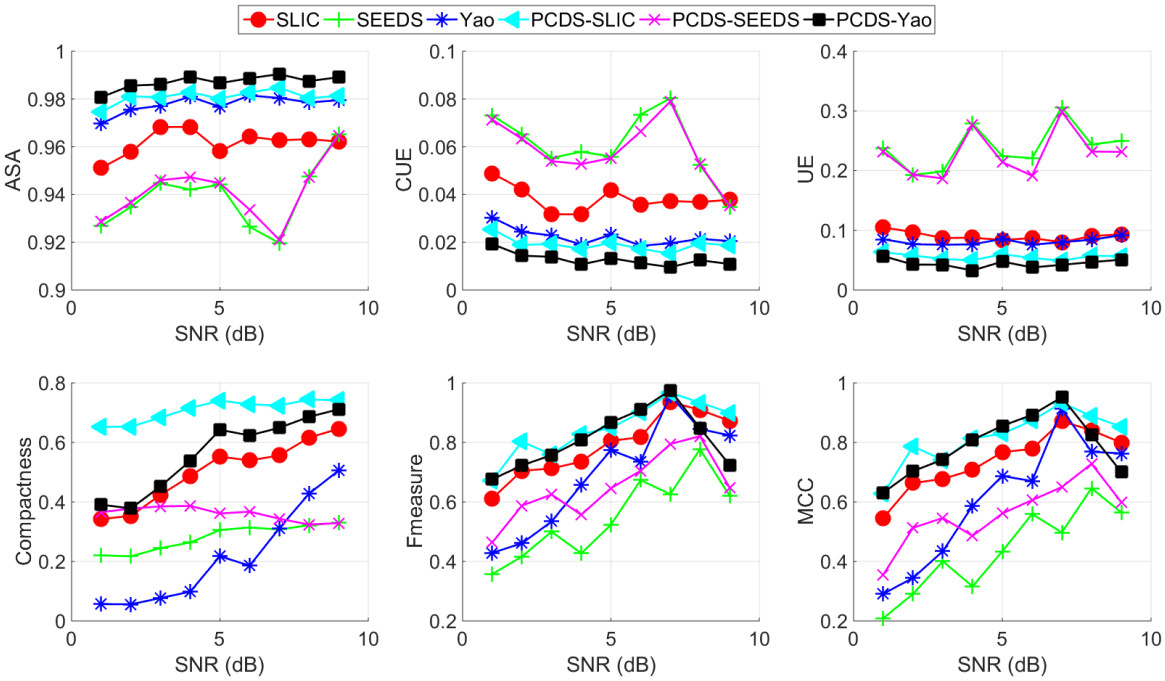


Figure 7.20: Evaluation of the PCDS algorithms and the state-of-the-art on LiDAR-like synthetic dataset in terms of SNR.

The superpixel-based surface extraction outperforms the ACWE in the very noisy cases and produces better ground surfaces under dense canopies. The ACWE algorithm shows slightly better performance in less challenging cases (less noise, less complex terrain and more strong ground surface that results from a less dense canopy). This figure also shows that the PCDS algorithm is more efficient than ACWE in terms of execution time in all the test cases. PCDS-Y shows a significant improvement over ACWE in terms of execution time.

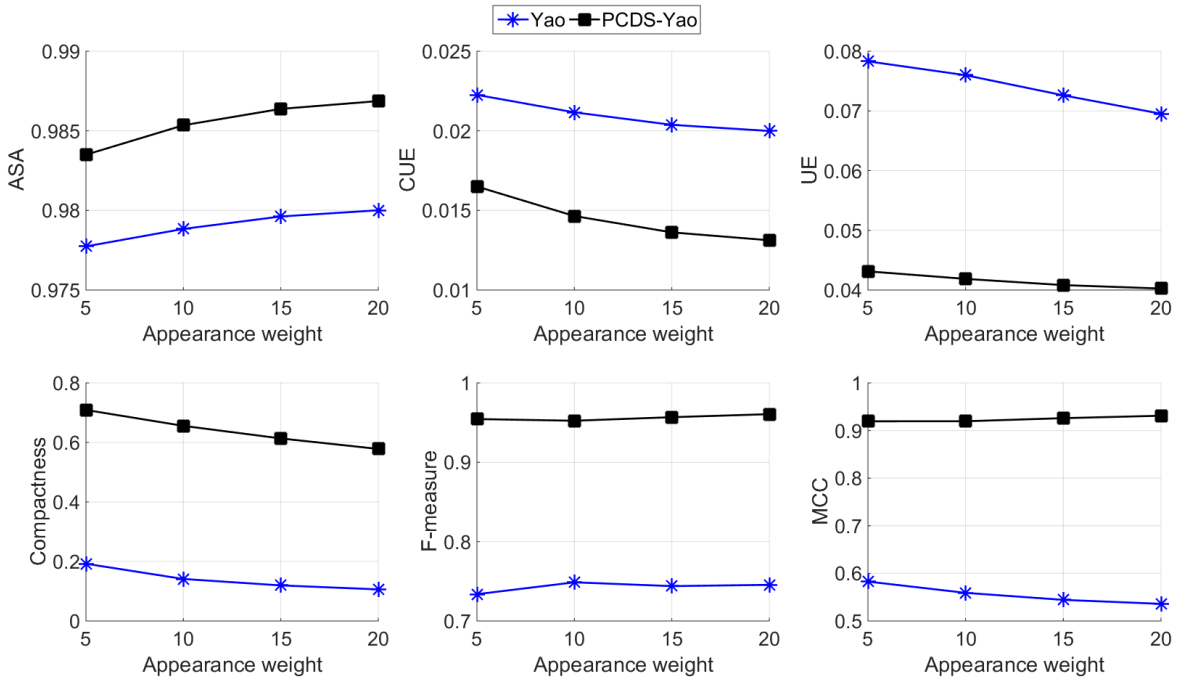


Figure 7.21: Evaluation of the Yao and PCDS-Yao algorithms on the primitive-shape synthetic dataset in terms of the appearance parameter.

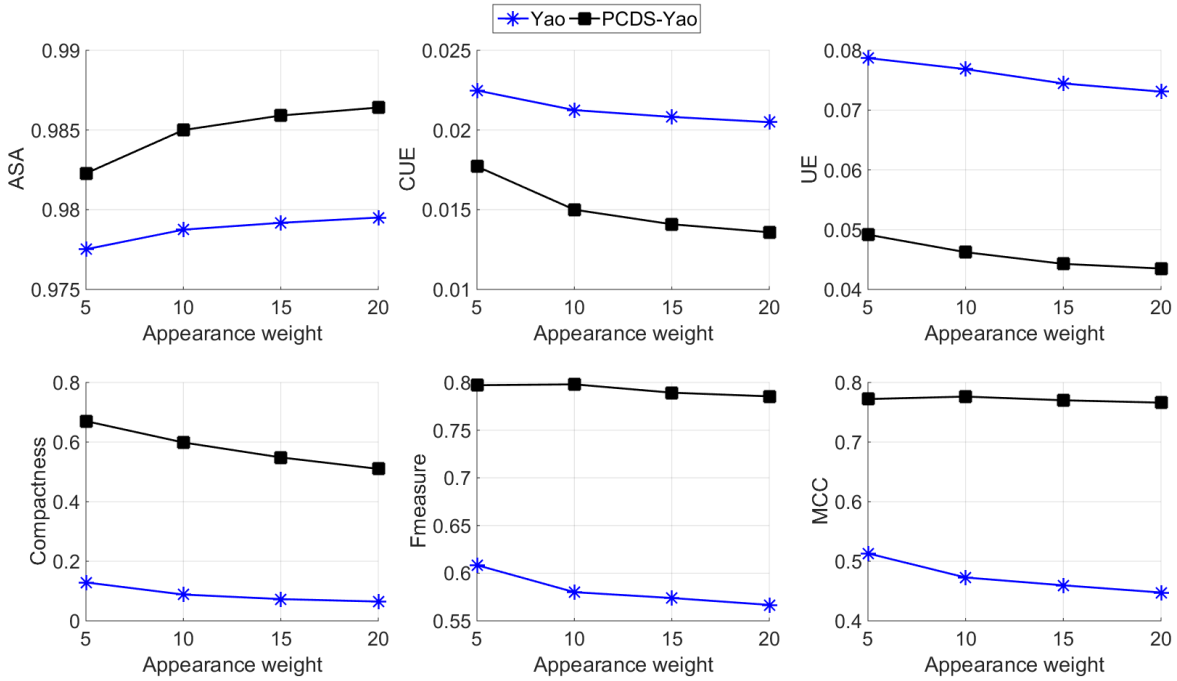


Figure 7.22: Evaluation of the Yao and PCDS-Yao algorithms on LiDAR-like synthetic dataset in terms of the appearance parameter.

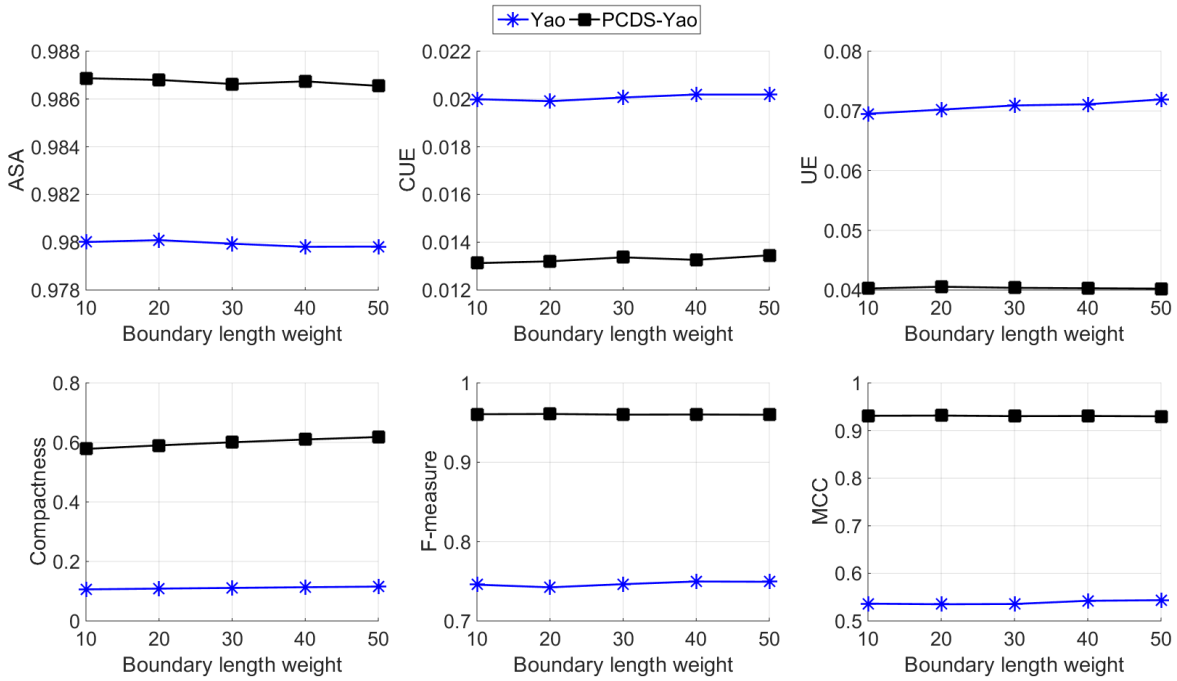


Figure 7.23: Evaluation of the Yao and PCDS-Yao algorithms on the primitive-shape synthetic dataset in terms of the boundary length parameter.

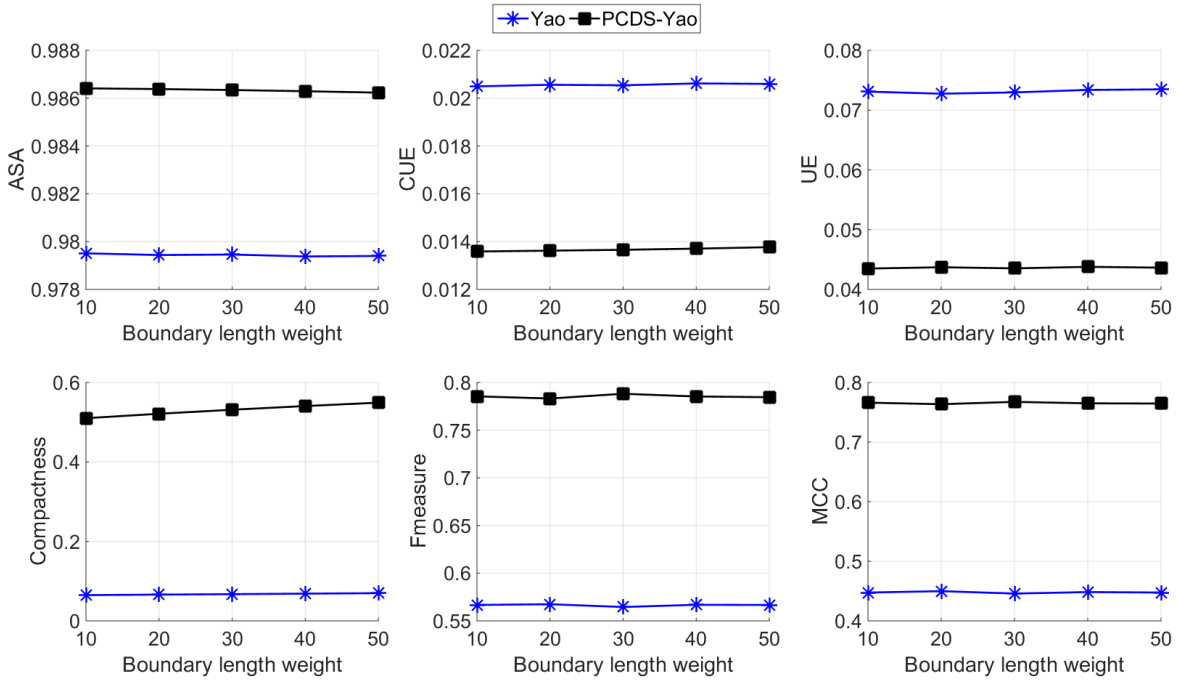


Figure 7.24: Evaluation of the Yao and PCDS-Yao algorithms on LiDAR-like synthetic dataset in terms of the boundary length parameter.

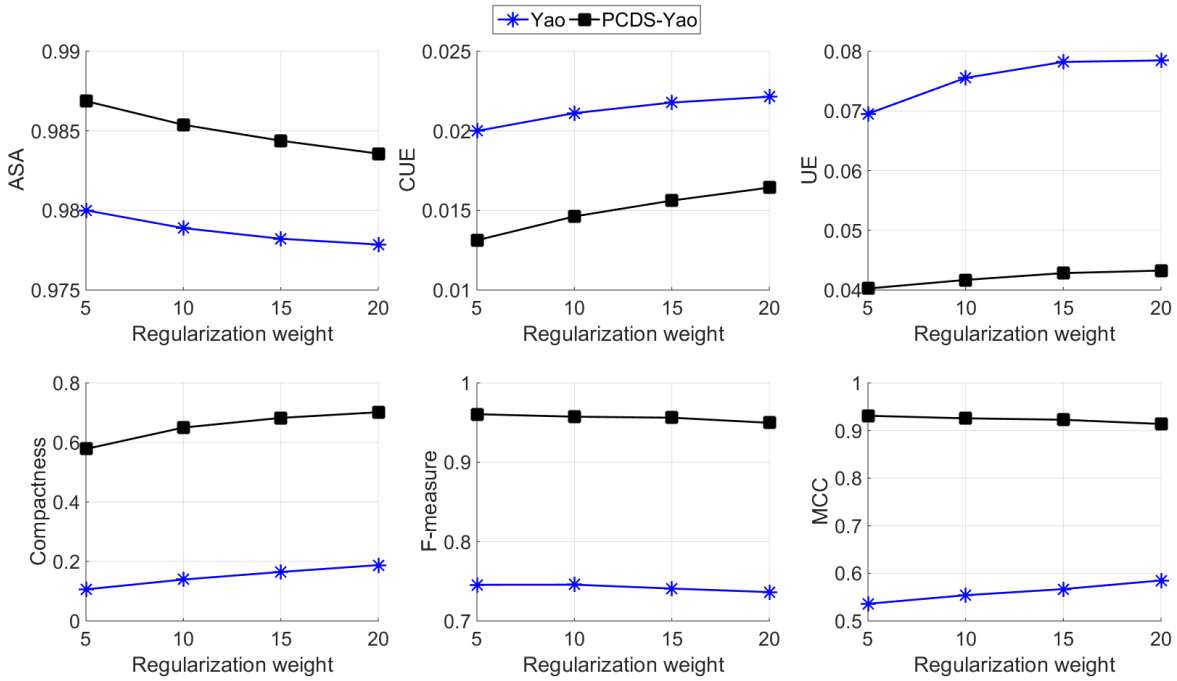


Figure 7.25: Evaluation of the Yao and PCDS-Yao algorithms on the primitive-shape synthetic dataset in terms of the regularization parameter.

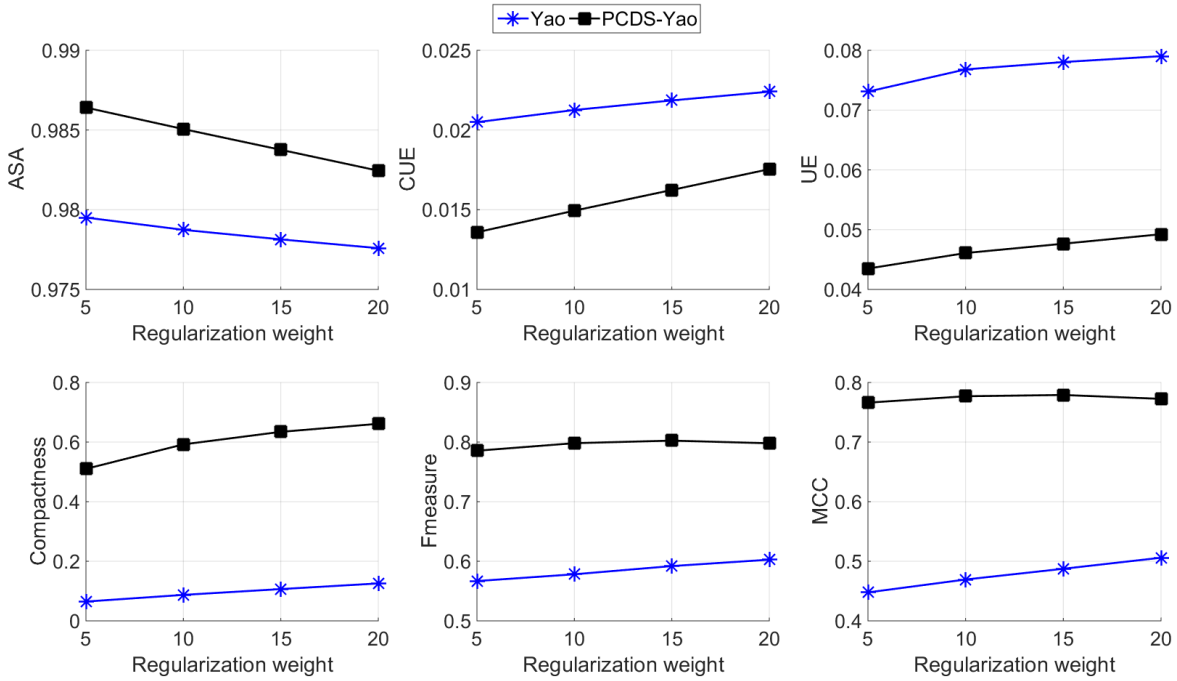


Figure 7.26: Evaluation of the Yao and PCDS-Yao algorithms on LiDAR-like synthetic dataset in terms of the regularization parameter.

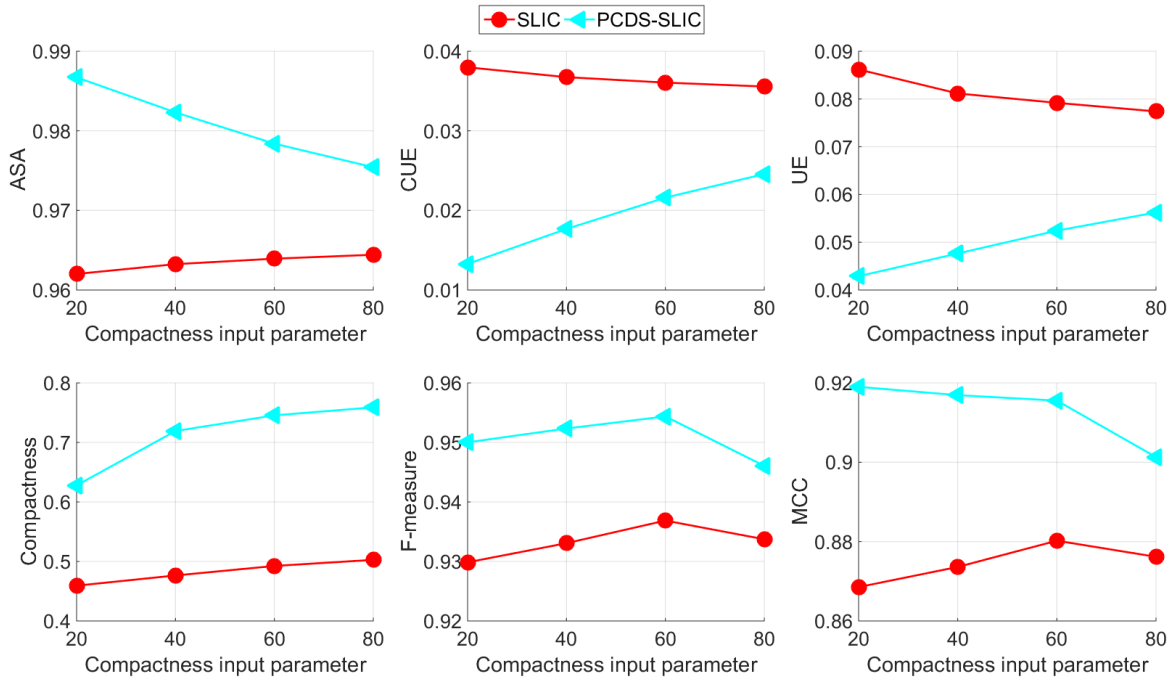


Figure 7.27: Evaluation of the SLIC and PCDS-SLIC algorithms on the primitive-shape synthetic dataset in terms of the compactness parameter.

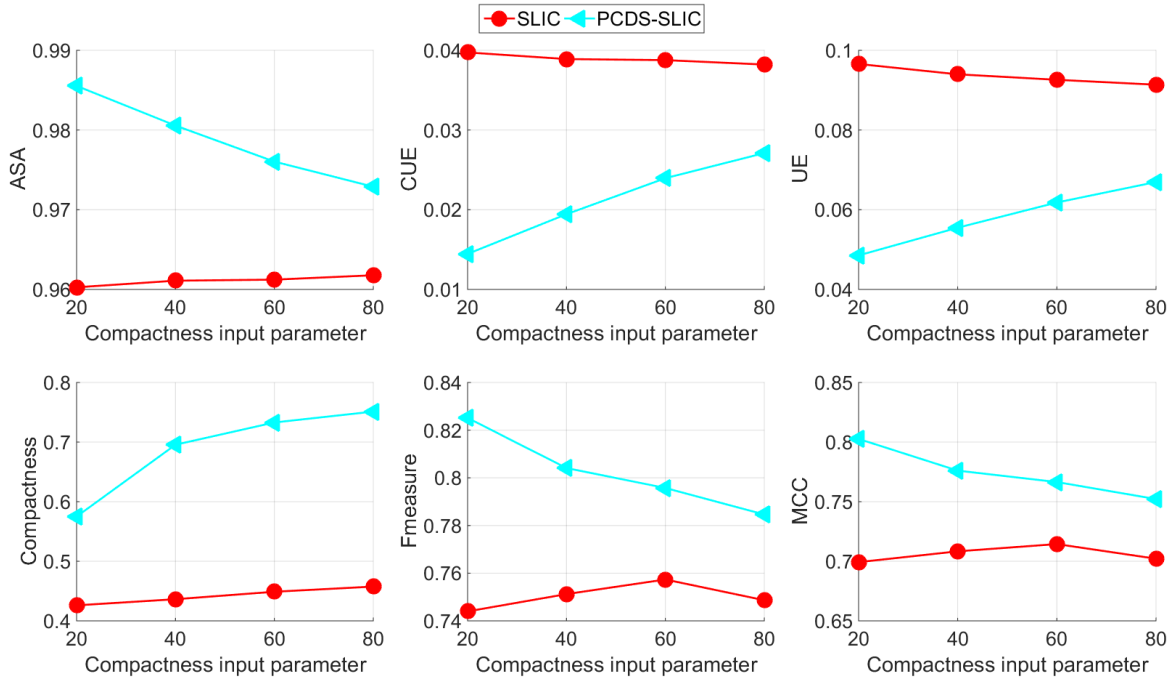


Figure 7.28: Evaluation of the SLIC and PCDS-SLIC algorithms on LiDAR-like synthetic dataset in terms of the compactness parameter.

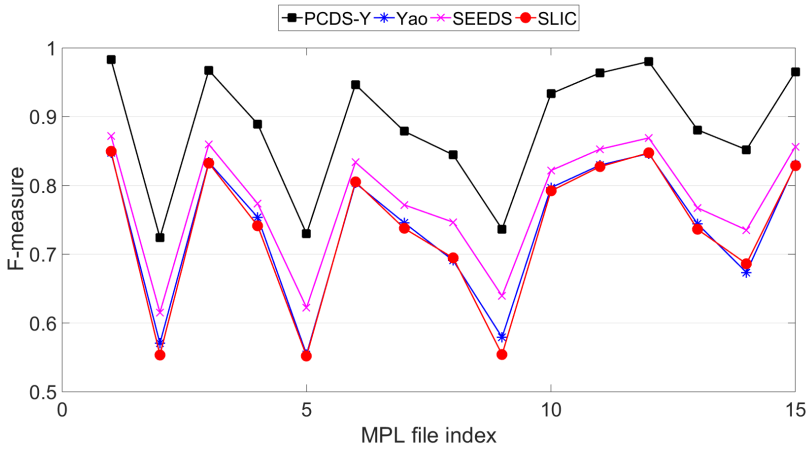


Figure 7.29: F-measure comparison between the proposed segmentation algorithm and other superpixel segmentation algorithms on MPL lidar point clouds.

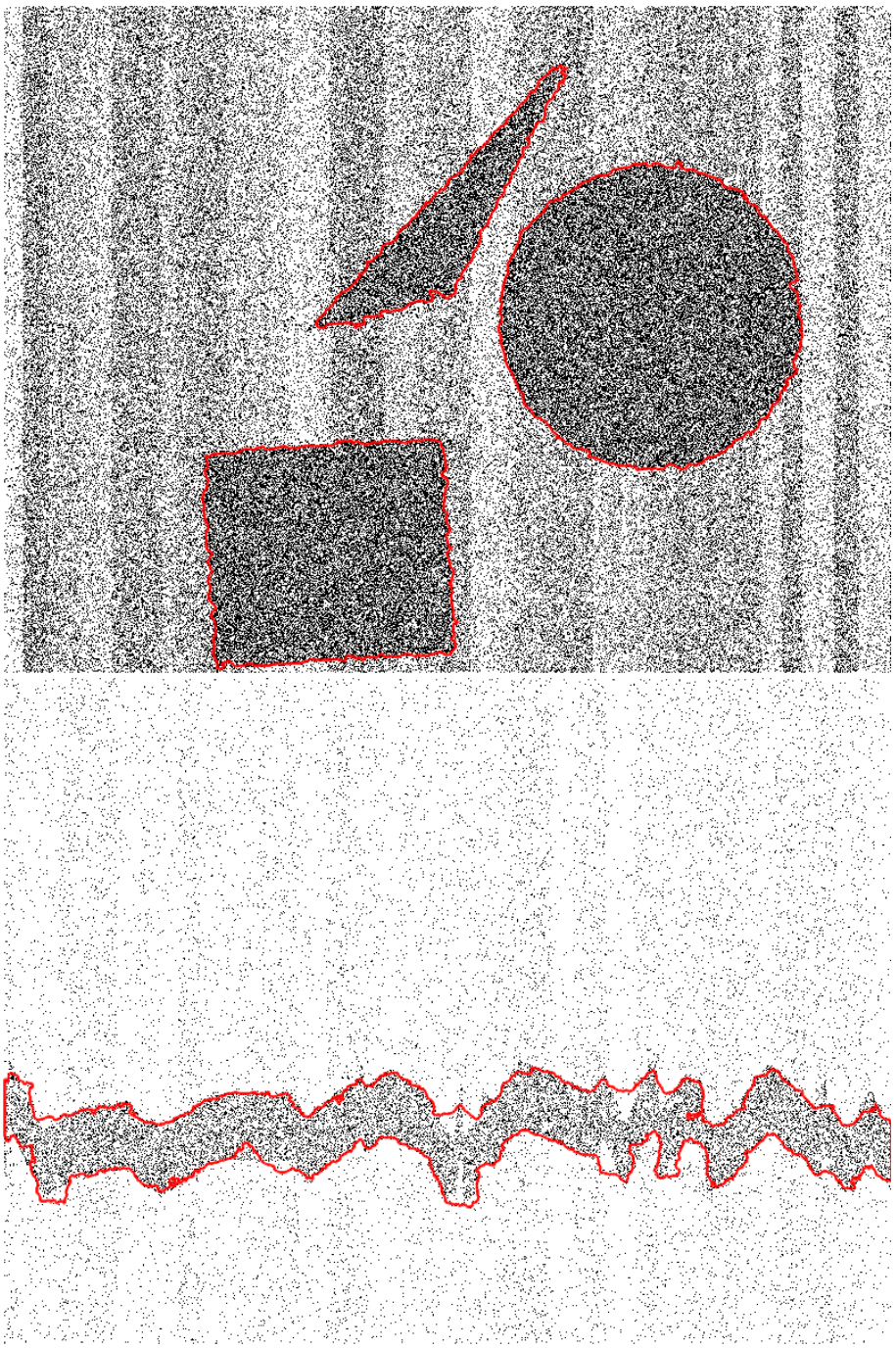
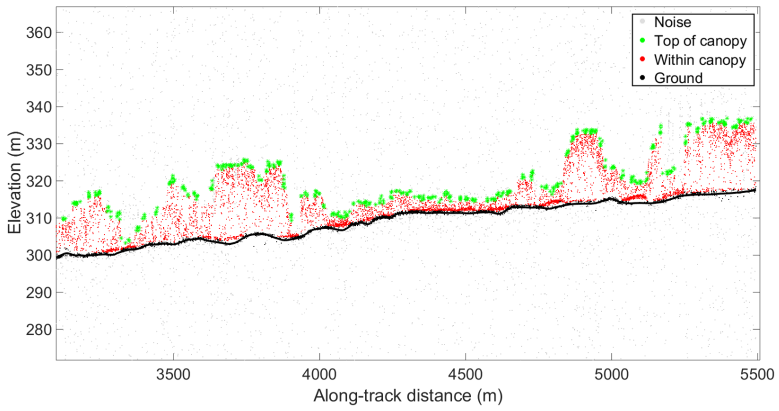
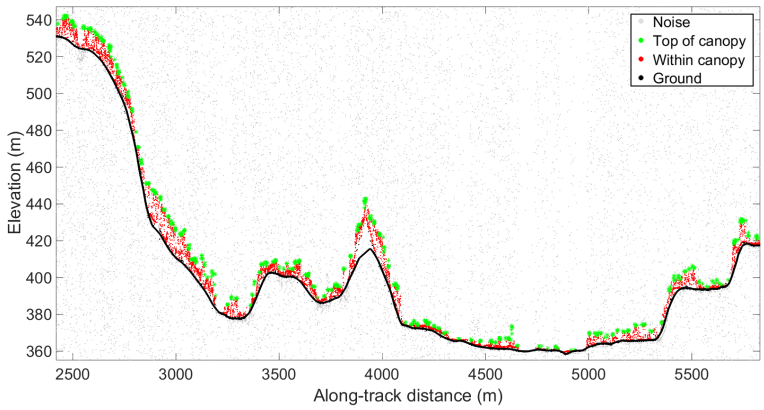


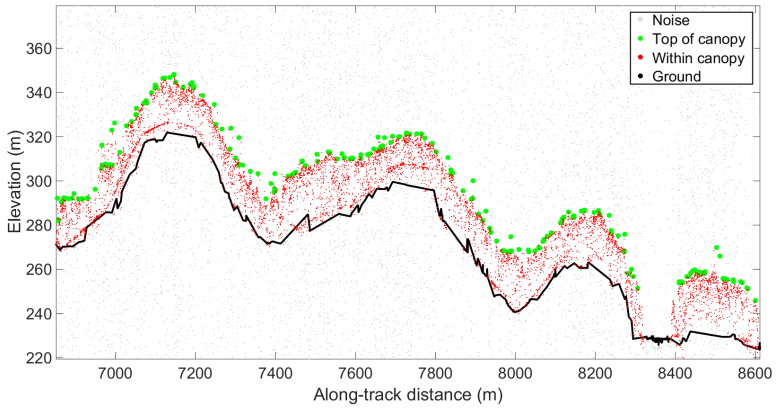
Figure 7.30: Example of the result of a full segmentation obtained by PCDS-Yao for the synthetic primitive shapes point cloud (top) and the synthetic LiDAR-like point (bottom). These results are much better than all other segmentation and clustering algorithms that we have attempted.



(a)



(b)



(c)

Figure 7.31: Example results of the PCDS-Yao superpixel-based surface estimation algorithm on MABEL 2014 (a),(b) and MABEL 2012 (c) transects show in Figure 7.6. PCDS-Yao algorithm showed better results as compared to ACWE in the bottom case (high noise density with partially occluded ground surface)

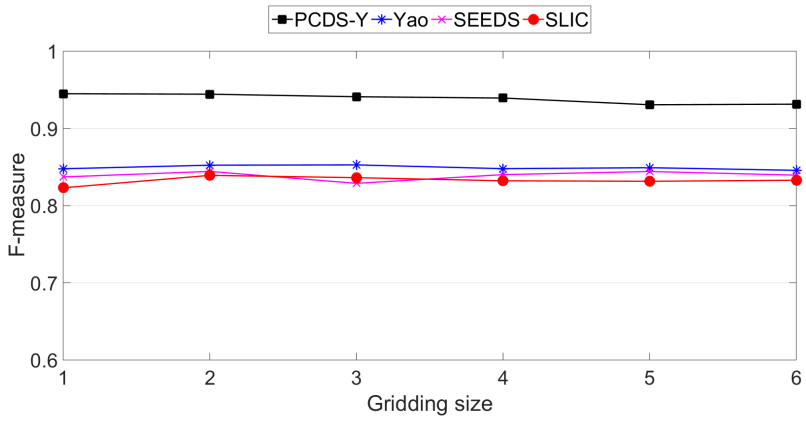


Figure 7.32: Effect of grid size on the proposed segmentation algorithm, PCDS-Yao, as compared to other superpixel segmentation algorithms. The x axis represents griding sizes ranging from 800×600 to 300×100, decreasing 100 pixels on each dimension.

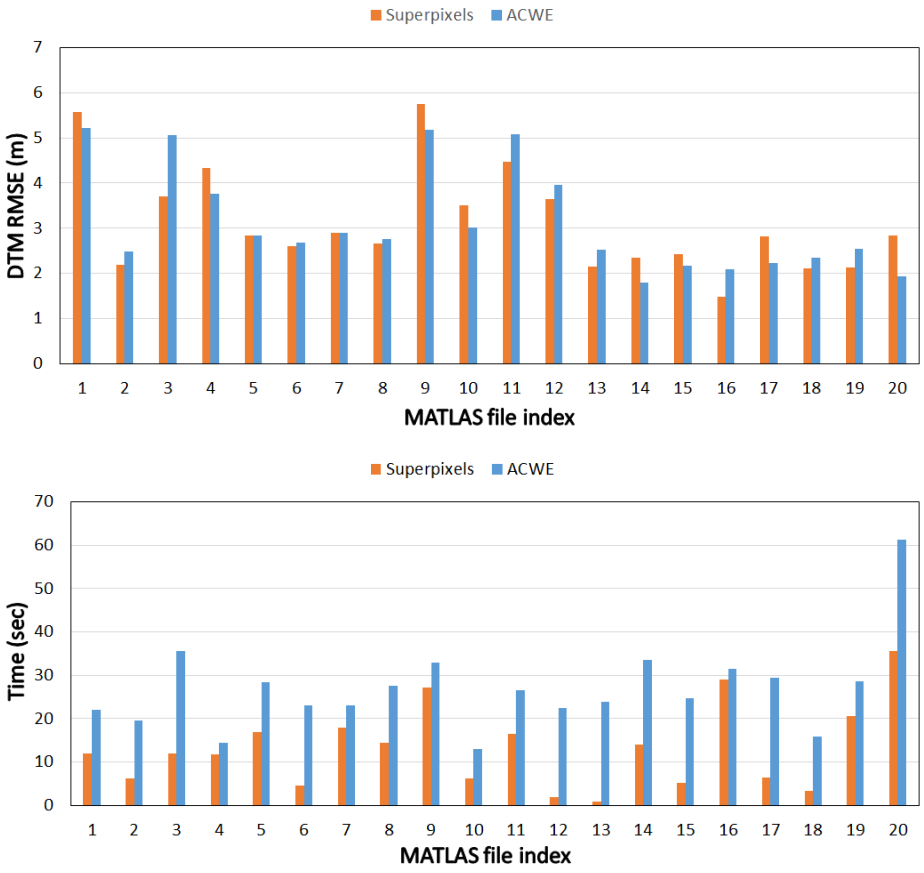


Figure 7.33: A comparison between the RMSE (top) and execution time (bottom) produced by the ACWE as compared to PCDS-Yao surface estimation algorithms.

Chapter 8

Conclusions and Future Work

8.1 Conclusions

The estimation of vegetation biomass on a large scale is of critical importance in characterizing and understanding planetary-scale changes that are taking place in the Earth system. ICESat-2 will be the continuation of this global biomass estimation effort after the use of ICESat. Because of the large amount of data that will be generated by the sensor, automated techniques are essential for extracting measurements in a way that is both fast and cost-effective. The launching of the ICESat-2's next-generation photon-counting ATLAS sensor brings new challenges to the current state-of-art surface estimation from LiDAR point clouds including higher noise levels, lower point densities and less canopy penetration.

To our knowledge, no existing system is capable of automatically analyzing data that will be obtained from the ICESat-2 photon-counting sensor. This research has outlined a set of novel approaches and algorithms for addressing the general problem of segmentation of noisy point cloud in plane. The segmentation was achieved using two methodologies. The first methodology is a noise-robust modified active contours algorithm which produced better segmentation results in extreme noise cases and the second methodology is the density-based superpixel segmentation that outperformed the state-of-art. The results of the segmentation is obtained using both methods to solve canopy-ground detection problem. It is expected that this work will facilitate follow-on studies that can bridge from vegetation height over multiple transects to large-scale estimates of biomass.

To our knowledge this dissertation introduces the first fully automatic ground and canopy surface estimation algorithm with a good accuracy on all available ICESat-2 datasets without assuming the availability of any prior elevation information. For this purpose a novel statistical noise reduction technique was introduced, a novel formulation for ACWE was exploited, a new initialization technique and stopping criterion were invented to provide an efficient and noise-robust automatic algorithm. The proposed algorithms are able to adaptively handle

different challenges including the presence of clouds. The algorithms in this case bridge the gaps that result from these anomalies. Moreover, in this dissertation we introduced the first superpixel-based segmentation algorithm for noisy point clouds in a plane in which the noise is assumed to be impulsive and of unknown, nonuniform distribution. The technique relies on point densities alone, which is important because additional information (e.g., color or texture) is not available for the applications of interest (LiDAR). We also introduced the first publicly available synthetic dataset for noisy point clouds in the plane. The dataset contains either primitive shapes or a LiDAR-like polygon of random points that have higher density than the noise points. This dataset will facilitate the algorithm analysis and development for future research in this subject.

We performed our experiments from two perspectives. From the surface finding perspective, more than 150 real photon-counting LiDAR point clouds have been tested and the estimated ground and canopy surfaces were compared to the ground truth. The RMSE of the estimated surface was good compared to the results obtained from previous research on ICESat data. The average RMSE was in the range of 0.2m – 3.2m which is much better than the results obtained from ICESat which were in the range of 2m – 6m. The superpixel-based surface extraction outperforms the ACWE in the very noisy cases and produces better ground surfaces under dense canopies. The ACWE algorithm shows slightly better performance in less challenging cases.

From the point cloud segmentation perspective, a test set of 400 point clouds with primitive shapes and 400 LiDAR-like synthetic point clouds in addition to the 150 real LiDAR point clouds were used for our experiments. Qualitative and quantitative analysis both demonstrated that our density-based approach yields superpixel segmentation that is much more accurate in terms of superpixel shape and conformity to object boundaries. Our experimental results have shown significant improvement over all other segmentation and clustering techniques that we have tried.

8.2 Future Work

This work can be extended in variety of ways, this includes:

1. Fusion of LiDAR data with image data from Landsat to enhance results.
2. Employing machine learning techniques (SVM, logistic regression) by extracting feature vector from each point (or patch of points). This feature vector may include:
   (a) (Along Track Distance, elevation) tuple.
   (b) Distance to the closest point.
   (c) Angle of closest point.

   (d) Area of triangle in which this point is forming a vertex.
   (e) Direction of closest point (making histogram bins for angle directions).
   (f) Number of points inside a circle of radius r centered at this point.
   (g) Distances between the point and the closest point to it in n direction neighborhood (maybe n = 4 or n = 8).

3. Canny-like density-based edge detector in point clouds.

4. Enhancing current algorithms using optimization techniques to obtain unbiased and bounded estimates of canopy heights.

5. Improving the evolution equation of ACWE by incorporating the narrow band stopping criterion into the evolution equation.

6. Enhancing the simulated data set in order to model the different signal densities of ground as compared to the top of canopy.

7. Enhancing the denoising process by incorporating a size filter to remove small connected components.

8. Investigate employing the scale-space theory to enhance the algorithm. This can be achieved by mapping the point cloud onto 2D grids with increasing gridding parameters (coarse to fine mapping) and use the results of coarse grids to guide the contours in fine grids.

Bibliography

[1] R. Nelson, M. A. Valenti, A. Short, and C. Keller, "A multiple resource inventory of Delaware using airborne laser data," *BioScience*, vol. 53, no. 10, pp. 981–992, 2003.

[2] K. Zhang, S.-C. Chen, D. Whitman, M.-L. Shyu, J. Yan, and C. Zhang, "A progressive morphological filter for removing nonground measurements from airborne LiDAR data," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 41, no. 4, pp. 872–882, 2003.

[3] U. C. Herzfeld, B. W. McDonald, B. F. Wallin, T. Neumann, T. Markus, A. Brenner, and C. Field, "Algorithm for detection of ground and canopy cover in micropulse photon-counting LiDAR altimeter data in preparation for the ICESat-2 mission," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 52, no. 4, pp. 2109–2125, 2014.

[4] K. C. Slatton, M. M. Crawford, and B. L. Evans, "Fusing interferometric radar and laser altimeter data to estimate surface topography and vegetation heights," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 39, no. 11, pp. 2470–2482, 2001.

[5] G. Sithole and G. Vosselman, "Experimental comparison of filter algorithms for bare-earth extraction from airborne laser scanning point clouds," *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 59, no. 1, pp. 85–101, 2004.

[6] K. Kampa and K. Slatton, "An adaptive multiscale filter for segmenting vegetation in ALSM data," in *Proceedings of IEEE International Geoscience and Remote Sensing Symposium IGARSS*, vol. 6, 2004, pp. 3837–3840.

[7] B. Koch, U. Heyder, and H. Weinacker, "Detection of individual tree crowns in airborne LiDAR data," *Photogrammetric Engineering and Remote Sensing*, vol. 72, no. 4, p. 357, 2006.

[8] H. Lee, K. C. Slatton, B. Roth, and W. Cropper Jr., "Adaptive clustering of airborne LiDAR data to segment individual tree crowns in managed pine forests," *International Journal of Remote Sensing*, vol. 31, no. 1, pp. 117–139, 2010.

[9] R. Nelson, W. Krabill, and J. Tonelli, "Estimating forest biomass and volume using airborne laser data," *Remote Sensing of Environment*, vol. 24, no. 2, pp. 247–267, 1988.

[10] E. N.æsset, "Predicting forest stand characteristics with airborne scanning laser using a practical two-stage procedure and field data," *Remote Sensing of Environment*, vol. 80, no. 1, pp. 88–99, 2002.

[11] J. B. Drake, R. G. Knox, R. O. Dubayah, D. B. Clark, R. Condit, J. B. Blair, and M. Hofton, "Above-ground biomass estimation in closed canopy neotropical forests using LiDAR remote sensing: Factors affecting the generality of relationships," *Global Ecology and Biogeography*, vol. 12, no. 2, pp. 147–159, 2003.

[12] S. C. Popescu, R. H. Wynne, and R. Nelson, "Measuring individual tree crown diameter with LiDAR and assessing its influence on estimating forest volume and biomass," *Canadian Journal of Remote Sensing*, vol. 29, no. 5, pp. 564–577, 2003.

[13] K. S. Lim and P. M. Treitz, "Estimation of above ground forest biomass from airborne discrete return laser scanner data using canopy-based quantile estimators," *Scandinavian Journal of Forest Research*, vol. 19, no. 6, pp. 558–570, 2004.

[14] R. Nelson, A. Short, and M. Valenti, "Measuring biomass and carbon in Delaware using an airborne profiling LiDAR," *Scandinavian Journal of Forest Research*, vol. 19, no. 6, pp. 500–511, 2004.

[15] Z. J. Bortolot and R. H. Wynne, "Estimating forest biomass using small footprint LiDAR data: An individual tree-based approach that incorporates training data," *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 59, no. 6, pp. 342–360, 2005.

[16] G. Sun, K. Ranson, J. Masek, A. Fu, and D. Wang, "Predicting tree height and biomass from GLAS data," in *10th International Symposium on Physical Measurements and Signatures in Remote Sensing*, 2007.

[17] V. Thomas, P. Treitz, J. McCaughey, and I. Morrison, "Mapping stand-level forest biophysical variables for a mixedwood boreal forest using LiDAR: An examination of scanning density," *Canadian Journal of Forest Research*, vol. 36, no. 1, pp. 34–47, 2006.

[18] J. A. Van Aardt, R. H. Wynne, and J. A. Scrivani, "LiDAR-based mapping of forest volume and biomass by taxonomic group using structurally homogenous segments," *Photogrammetric Engineering and Remote Sensing*, vol. 74, no. 8, pp. 1033–1044, 2008.

[19] F. G. Hall, K. Bergen, J. B. Blair, R. Dubayah, R. Houghton, G. Hurtt, J. Kellndorfer, M. Lefsky, J. Ranson, and S. Saatchi, "Characterizing 3D vegetation structure from space: Mission requirements," *Remote Sensing of Environment*, vol. 115, no. 11, pp. 2753–2775, 2011.

[20] S. Zolkos, S. Goetz, and R. Dubayah, "A meta-analysis of terrestrial aboveground biomass estimation using LiDAR remote sensing," *Remote Sensing of Environment*, vol. 128, pp. 289 –298, 2013.

[21] G. Sun, K. Ranson, D. Kimes, J. Blair, and K. Kovacs, "Forest vertical structure from GLAS: An evaluation using LVIS and SRTM data," *Remote Sensing of Environment*, vol. 112, no. 1, pp. 107 –117, 2008.

[22] M. A. Lefsky, D. J. Harding, M. Keller, W. B. Cohen, C. C. Carabajal, F. Del Bom Espirito-Santo, M. O. Hunter, and R. de Oliveira, "Estimates of forest canopy height and aboveground biomass using ICESat," *Geophysical Research Letters*, vol. 32, no. 22, n/a–n/a, 2005, L22S02, ISSN: 1944-8007.

[23] Z. Guo, H. Chi, and G. Sun, "Estimating forest aboveground biomass using HJ-1 Satellite CCD and ICESat GLAS waveform data," *Science China Earth Sciences*, vol. 53, no. 1, pp. 16–25, 2010.

[24] H. Zwally, B. Schutz, W. Abdalati, J. Abshire, C. Bentley, A. Brenner, J. Bufton, J. Dezio, D. Hancock, D. Harding, T. Herring, B. Minster, K. Quinn, S. Palm, J. Spinhirne, and R. Thomas, "ICESat's laser measurements of polar ice, atmosphere, ocean, and land," *Journal of Geodynamics*, vol. 34, no. 34, pp. 405 –445, 2002, ISSN: 0264-3707.

[25] NASA, ICESat-2, [Online]. Available: `http://icesat.gsfc.nasa.gov/icesat2/index.php` (visited on 07/05/2016).

[26] B. Schutz, H. Zwally, C. Shuman, D. Hancock, and J. DiMarzio, "Overview of the ICESat mission," *Geophysical Research Letters*, vol. 32, no. 21, 2005.

[27] W. Abdalati, H. Zwally, R. Bindschadler, B. Csatho, S. Farrell, H. Fricker, D. Harding, R. Kwok, M. Lefsky, T. Markus, A. Marshak, T. Neumann, S. Palm, B. Schutz, B. Smith, J. Spinhirne, and C. Webb, "The ICESat-2 laser altimetry mission," *Proceedings of the IEEE*, vol. 98, no. 5, pp. 735–751, 2010.

[28] J. D. Spinhirne, "Micro pulse LiDAR," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 31, no. 1, pp. 48–55, 1993, ISSN: 0196-2892.

[29] M. Nilsson, "Estimation of tree heights and stand volume using an airborne LiDAR system," *Remote Sensing of Environment*, vol. 56, no. 1, pp. 1–7, 1996.

[30] E. Naesset, "Estimating timber volume of forest stands using airborne laser scanner data," *Remote Sensing of Environment*, vol. 61, no. 2, pp. 246–253, 1997.

[31] J. E. Means, S. A. Acker, B. J. Fitt, M. Renslow, L. Emerson, and C. J. Hendrix, "Predicting forest stand characteristics with airborne scanning LiDAR," *Photogrammetric Engineering and Remote Sensing*, vol. 66, no. 11, pp. 1367–1372, 2000.

[32] S. C. Popescu, R. H. Wynne, and J. A. Scrivani, "Fusion of small-footprint LiDAR and multispectral data to estimate plot-level volume and biomass in deciduous and pine forests in Virginia, USA," *Forest Science*, vol. 50, no. 4, pp. 551–565, 2004.

[33] J. A. van Aardt, R. H. Wynne, and R. G. Oderwald, "Forest volume and biomass estimation using small-footprint LiDAR-distributional parameters on a per-segment basis," *Forest Science*, vol. 52, no. 6, pp. 636–649, 2006.

[34] K. Barbieri, A. Brenner, T. Markus, T. Neumann, J. Saba, D. Yi, and K. Brunt, "Description of ICESat-II simulated data created from Sigma Space MPL laser data," *NASA Goddard Space Flight Center, Greenbelt, MD, USA, Tech. Rep*, 2010.

[35] M. McGill, T. Markus, V. S. Scott, and T. Neumann, "The multiple altimeter beam experimental LiDAR (MABEL): An airborne simulator for the ICESat-2 mission," *Journal of Atmospheric and Oceanic Technology*, vol. 30, no. 2, pp. 345–352, 2013.

[36] T. F. Chan and L. A. Vese, "Active contours without edges," *IEEE Transactions on Image Processing*, vol. 10, no. 2, pp. 266–277, 2001.

[37] J. B. Campbell and R. H. Wynne, *Introduction to Remote Sensing*, fifth. New York, USA: The Guilford Press, 2011, ISBN: 978-1609181765.

[38] T. Fujii and T. Fukuchi, *Laser Remote Sensing*. CRC Press, 2005.

[39] M. Flood, "Laser allimetry: From science to commercial LiDAR mapping," *Photogrammetric Engineering and Remote Sensing*, vol. 67, no. 11, 2001.

[40] X. Liu, "Airborne LiDAR for DEM generation: Some critical issues," *Progress in Physical Geography*, vol. 32, no. 1, pp. 31–49, 2008.

[41] J. Carter, K. Schmid, K. Waters, L. Betzhold, B. Hadley, R. Mataosky, and J. Halleran, *Lidar 101: An introduction to LiDAR technology, data and applications*, 2012.

[42] P. Lewis and S. Hancock, "LiDAR for vegetation applications," *UCL, Gower St, London, UK*, 2007.

[43] W. Krabill, J. Collins, L. Link, R. Swift, and M. Butler, "Airborne laser topographic mapping results," *Photogrammetric Engineering and Remote Sensing*, vol. 50, no. 6, pp. 685–694, 1984.

[44] J. L. Bufton, J. B. Garvin, J. F. Cavanaugh, L. A. Ramos-Izquierdo, T. D. Clem, and W. B. Krabill, "Airborne LiDAR for profiling of surface topography," *Optical Engineering*, vol. 30, no. 1, pp. 72–78, 1991.

[45] N. Pfeifer and C. Briese, "Geometrical aspects of airborne laser scanning and terrestrial laser scanning," *International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences*, vol. 36, no. 3/W52, pp. 311–319, 2007.

[46] G. Sithole and G. Vosselman, "Report: ISPRS comparison of filters," *ISPRS commission III, working group*, vol. 3, 2003.

[47] K. Kraus and N. Pfeifer, "Determination of terrain models in wooded areas with airborne laser scanner data," *ISPRS Journal of Photogrammetry and remote Sensing*, vol. 53, no. 4, pp. 193–203, 1998.

[48] M. E. Romano, "Innovation in LiDAR processing technology," *Photogrammetric Engineering and Remote Sensing*, vol. 70, no. 11, pp. 1201–1206, 2004.

[49] Q. Chen, "Airborne LiDAR data processing and information extraction," *Photogrammetric engineering and remote sensing*, vol. 73, no. 2, p. 109, 2007.

[50] Q. Chen, P. Gong, D. Baldocchi, and G. Xie, "Filtering airborne laser scanning data with morphological methods," *Photogrammetric Engineering and Remote Sensing*, vol. 73, no. 2, pp. 175–185, 2007.

[51] A. N. Strahler and A. H. Strahler, *Environmental Geoscience: Interaction between Natural Systems and Man*. New York, USA: Wiley, 1973.

[52] M. Kass, A. Witkin, and D. Terzopoulos, "Snakes: Active contour models," *International Journal of Computer Vision*, vol. 1, no. 4, pp. 321–331, 1988.

[53] C. Li, C. Xu, C. Gui, and M. D. Fox, "Level set evolution without re-initialization: A new variational formulation," in *IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*, vol. 1, 2005, pp. 430–436.

[54] E. Sojka, J. Gaura, T. Fabián, and M. Krumnikl, "Active contours without edges and without reinitialisation," in *IEEE World Congress on Nature & Biologically Inspired Computing (NaBIC)*, 2009, pp. 914–917.

[55] X. F. Wang, D. Huang, and H. Xu, "An efficient local chan–vese model for image segmentation," *Pattern Recognition*, vol. 43, no. 3, pp. 603–618, 2010.

[56] X. Ren and J. Malik, "Learning a classification model for segmentation," in *Proc. 9th Int'l. Conf. Computer Vision*, vol. 1, 2003, pp. 10–17.

[57] R. Achanta, A. Shaji, K. Smith, A. Lucchi, P. Fua, and S. Susstrunk, "SLIC superpixels compared to state-of-the-art superpixel methods," *IEEE Transactions on Pattern Analysis and Machine Intelligence.*, vol. 34, no. 11, pp. 2274–2282, 2012.

[58] K. Kraus and N. Pfeifer, "Advanced DTM generation from LiDAR data," *International Archives Of Photogrammetry, Remote Sensing and Spatial Information Sciences*, vol. 34, no. 3/W4, pp. 23–30, 2001.

[59] A. L. Neuenschwander, M. M. Crawford, L. A. Magruder, C. A. Weed, R. Cannata, D. Fried, R. Knowlton, and R. Heinrichs, "Terrain classification of LADAR data over Haitian urban environments using a lower envelope follower and adaptive gradient operator," in *SPIE Defense, Security, and Sensing*, International Society for Optics and Photonics, vol. 7684, 2010, pp. 768 408–768408–9.

[60] T. H. Stevenson, L. A. Magruder, A. L. Neuenschwander, and B. Bradford, "Automated bare earth extraction technique for complex topography in light detection and ranging surveys," *Journal of Applied Remote Sensing*, vol. 7, no. 1, p. 073 560, 2013.

[61] J. Shan and A. Sampath, "Urban DEM generation from raw LiDAR data: A labeling algorithm and its performance," *Photogrammetric Engineering and Remote Sensing*, vol. 71, no. 2, pp. 217–226, 2005.

[62] P. Kumar, C. P. McElhinney, P. Lewis, and T. McCarthy, "An automated algorithm for extracting road edges from terrestrial mobile LiDAR data," *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 85, pp. 44–55, 2013.

[63] L. D. Cohen, "On active contour models and balloons," *Computer Vision and Image Understanding (CVGIP)*, vol. 53, no. 2, pp. 211–218, 1991.

[64] Y. Choe, S. Ahn, and M. J. Chung, "Fast point cloud segmentation for an intelligent vehicle using sweeping 2D laser scanners," in *9th International Conference on Ubiquitous Robots and Ambient Intelligence (URAI)*, 2012.

[65] L. A. Magruder, M. E. Wharton, K. D. Stout, and A. L. Neuenschwander, "Noise filtering techniques for photon-counting LADAR data," in *SPIE Defense, Security, and Sensing*, International Society for Optics and Photonics, 2012, 83790Q.

[66] K. Horan and J. P. Kerekes, "An automated statistical analysis approach to noise reduction for photon-counting LiDAR systems," in *Proceedings of IEEE International Geoscience and Remote Sensing Symposium IGARSS*, 2013, pp. 4336–4339.

[67] M. S. Moussavi, W. Abdalati, T. Scambos, and A. Neuenschwander, "Applicability of an automatic surface detection approach to micro-pulse photon-counting LiDAR altimetry data: Implications for canopy height retrieval from future ICESat-2 data," *International Journal of Remote Sensing*, vol. 35, no. 13, pp. 5263–5279, 2014.

[68] J. Zhang, J. Kerekes, B. Csatho, T. Schenk, and R. Wheelwright, "A clustering approach for detection of ground in micropulse photon-counting LiDAR altimeter data," in *Proceedings of IEEE International Geoscience and Remote Sensing Symposium IGARSS*, 2014, pp. 177–180.

[69] D. Gwenzi and M. A. Lefsky, "Prospects of photon counting LiDAR for savanna ecosystem structural studies," *ISPRS International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, vol. 1, pp. 141–147, 2014.

[70] P. Montesano, J. Rosette, G. Sun, P. North, R. Nelson, R. Dubayah, K. Ranson, and V. Kharuk, "The uncertainty of biomass estimates from modeled ICESat-2 returns across a boreal forest gradient," *Remote Sensing of Environment*, vol. 158, pp. 95–109, 2015.

[71] C. Xu, D. L. Pham, and J. L. Prince, "Image segmentation using deformable models," *Handbook of Medical Imaging*, vol. 2, pp. 129–174, 2000.

[72] S. Liu and Y. Ye, "Point cloud segmentation using gradient vector flow snake," in *IEEE International Conference on Information Science and Technology (ICIST)*, 2011, pp. 1114–1118.

[73] C. Xu and J. L. Prince, "Gradient vector flow: A new external force for snakes," in *IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*, 1997, pp. 66–71.

[74] V. Caselles, F. Catté, T. Coll, and F. Dibos, "A geometric model for active contours in image processing," *Numerische Mathematik*, vol. 66, no. 1, pp. 1–31, 1993.

[75] B. Vemuri and Y. Chen, "Joint image registration and segmentation," in *Geometric Level Set Methods in Imaging, Vision, and Graphics*, Springer, 2003, pp. 251–269.

[76] H. P. Ho, Y. Chen, H. Liu, and P. Shi, "Level set active contours on unstructured point cloud," in *IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*, vol. 2, 2005, pp. 655–662.

[77] M. Weber, A. Blake, and R. Cipolla, "Initialisation and termination of active contour level-set evolutions," in *IEEE workshop on Variational, Geometric and Level Set Methods in Computer Vision*, 2003, pp. 161–168.

[78] A. Morar, F. Moldoveanu, and E. Gröller, "Image segmentation based on active contours without edges," in *IEEE International Conference on Intelligent Computer Communication and Processing (ICCP)*, vol. 30, 2012, pp. 213–220.

[79] S. Lankton and A. Tannenbaum, "Localizing region-based active contours," *IEEE Transactions on Image Processing*, vol. 17, no. 11, pp. 2029–2039, 2008.

[80] W. Wen, C. He, and M. Li, "Transition region-based active contour model for image segmentation," *Journal of Electronic Imaging*, vol. 22, no. 1, p. 013 021, 2013.

[81] M. Awadallah, S. Ghannam, A. Abbott, and A. Ghanem, "Active contour models for extracting ground and forest canopy curves from discrete laser altimeter data," in *13th International Conference on LiDAR Applications for Assessing Forest Ecosystems*, 2013, pp. 129–136.

[82] M. Awadallah, L. Abbott, and S. Ghannam, "Segmentation of sparse noisy point clouds using active contour models," in *Proceedings of the IEEE International Conference on Image Processing (ICIP)*, 2014, pp. 6061–6065.

[83] J. Shi and J. Malik, "Normalized cuts and image segmentation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, no. 8, pp. 888–905, 2000.

[84] A. P. Moore, J. Prince, J. Warrell, U. Mohammed, and G. Jones, "Superpixel lattices," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2008, pp. 1–8.

[85] A. P. Moore, S. J. Prince, and J. Warrell, "'Lattice Cut'-constructing superpixels using layer constraints," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2010, pp. 2117–2124.

[86] L. Vincent and P. Soille, "Watersheds in digital spaces: An efficient algorithm based on immersion simulations," *IEEE Transactions on Pattern Analysis and Machine Intelligence.*, no. 6, pp. 583–598, 1991.

[87] A. Levinshtein, A. Stere, K. N. Kutulakos, D. J. Fleet, S. J. Dickinson, and K. Siddiqi, "Turbopixels: Fast superpixels using geometric flows," *IEEE Transactions on Pattern Analysis and Machine Intelligence.*, vol. 31, no. 12, pp. 2290–2297, 2009.

[88] M. Van den Bergh, X. Boix, G. Roig, and L. Van Gool, "SEEDS: superpixels extracted via energy-driven sampling," *International Journal of Computer Vision.*, vol. 111, no. 3, pp. 298–314, 2015.

[89] J. Yao, M. Boben, S. Fidler, and R. Urtasun, "Real-time coarse-to-fine topologically preserving segmentation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015, pp. 2947–2955.

[90] J. Papon, A. Abramov, M. Schoeler, and F. Worgotter, "Voxel cloud connectivity segmentation-supervoxels for point clouds," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2013, pp. 2027–2034.

[91] M. Schoeler, J. Papon, and F. Worgotter, "Constrained planar cuts-object partitioning for point clouds," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015, pp. 5207–5215.

[92] T. Hamedani, R. Zarei, and A. Harati, "Superpixel based RGB-D image segmentation using Markov random field," in *International Symposium on Artificial Intelligence and Signal Processing (AISP)*, IEEE, 2015, pp. 89–94.

[93] Z. Zhang, "Microsoft Kinect sensor and its effect," *MultiMedia, IEEE*, vol. 19, no. 2, pp. 4–10, 2012.

[94] A.-V. Vo, L. Truong-Hong, D. F. Laefer, and M. Bertolotto, "Octree-based region growing for point cloud segmentation," *ISPRS Journal of Photogrammetry and Remote Sensing.*, vol. 104, pp. 88–100, 2015.

[95] M. C. Bakkay, M. Arafa, and E. Zagrouba, "Dense 3D SLAM in dynamic scenes using Kinect," in *Pattern Recognition and Image Analysis.* Springer, 2015, pp. 121–129.

[96] V. Kurlin, "A fast and robust algorithm to count topologically persistent holes in noisy clouds," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2014, pp. 1458–1463.

[97] J. Yuan, E. Bae, and X.-C. Tai, "A study on continuous max-flow and min-cut approaches," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2010, pp. 2217–2224.

[98] T. F. Chan, S. Esedoglu, and M. Nikolova, "Algorithms for finding global minimizers of image segmentation and denoising models," *SIAM Journal on Applied Mathematics.*, vol. 66, no. 5, pp. 1632–1648, 2006.

[99] P. Getreuer, "Rudin-Osher-Fatemi total variation denoising using split Bregman," *Image Processing On Line.*, vol. 2, pp. 74–95, 2012.

[100] M. Ester, H.-P. Kriegel, J. Sander, and X. Xu, "A density-based algorithm for discovering clusters in large spatial databases with noise.," in *Proceedings of the 2nd International Conference on Knowledge Discovery and Data Mining (KDD'96)*, 1996, pp. 226–231.

[101] T. N. Tran, K. Drab, and M. Daszykowski, "Revised DBSCAN algorithm to cluster data with dense adjacent clusters," *Chemometrics and Intelligent Laboratory Systems.*, vol. 120, pp. 92–96, 2013.

[102] G. Azzopardi and N. Petkov, "A CORF computational model of a simple cell that relies on LGN input outperforms the Gabor function model," *Biological Cybernetics*, vol. 106, no. 3, pp. 177–189, 2012.

[103] N. Wiener, *Extrapolation, interpolation, and smoothing of stationary time series*. MIT press Cambridge, MA, 1949, vol. 2.

[104] L. Vincent, "Morphological grayscale reconstruction in image analysis: Applications and efficient algorithms," *IEEE Transactions on Image Processing*, vol. 2, no. 2, pp. 176–201, 1993, ISSN: 1057-7149.

[105] J. Canny, "A computational approach to edge detection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. PAMI-8, no. 6, pp. 679–698, 1986, ISSN: 0162-8828.

[106] A. Golovinskiy and T. Funkhouser, "Min-cut based segmentation of point clouds," in *IEEE 12th International Conference on Computer Vision Workshops (ICCV Workshops)*, 2009, pp. 39–46.

[107] N. Otsu, "A threshold selection method from gray-level histograms," *Automatica.*, vol. 11, no. 285-296, pp. 23–27, 1975.

[108] P. Axelsson, "DEM generation from laser scanner data using adaptive TIN models," *International Archives of Photogrammetry and Remote Sensing*, vol. 33, no. B4/1; PART 4, pp. 110–117, 2000.

[109] D. R. Martin, C. C. Fowlkes, and J. Malik, "Learning to detect natural image boundaries using local brightness, color, and texture cues," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 26, no. 5, pp. 530–549, 2004.

[110] P. Baldi, S. Brunak, Y. Chauvin, C. A. Andersen, and H. Nielsen, "Assessing the accuracy of prediction algorithms for classification: An overview," *Bioinformatics*, vol. 16, no. 5, pp. 412–424, 2000.

[111] NASA, G-LiHT, [Online]. Available: `http://gliht.gsfc.nasa.gov/` (visited on 07/05/2016).

[112] A. Schick, M. Fischer, and R. Stiefelhagen, "Measuring and evaluating the compactness of superpixels," in *21st International Conference on Pattern Recognition (ICPR)*, 2012, pp. 930–934.

Appendices

Appendix A

Additional Qualitative Results

In this appendix we are showing a comparison between our proposed algorithms (PCDS-Yao, PCDS-SLIC and PCDS-SEEDS) and the state-of-the-art algorithms, Yao and SLIC and SEEDS. The original source code implementation for the algorithms downloaded from the authors' websites. The experiments were performed on both synthetic and real LiDAR (MPL, MABEL and MATLAS) datasets. The weights and parameters for the traditional algorithms were tested and carefully chosen empirically to produce the best qualitative results. The weights that produced best results for Yao were $\lambda_{pos} = 5$, $\lambda_{app} = 5$ and $\lambda_b = 50$. The value of λ_b was chosen to be relatively high in order to keep the boundaries of the superpixels as short as possible (because these algorithms suffer from the problem of weaving around points creating very long and irregular superpixel boundaries). In the SLIC experiment the only weight is the compactness factor. On one hand when the compactness factor is selected to be too high, the superpixel boundaries become very stiff causing the final superpixels to remain without changing its initial location. On the other hand, if the compactness factor is selected to be too small the superpixels boundaries become very long and tend to "snake" around points in the point cloud. We chose this parameter to be a middle value (i.e. 10).

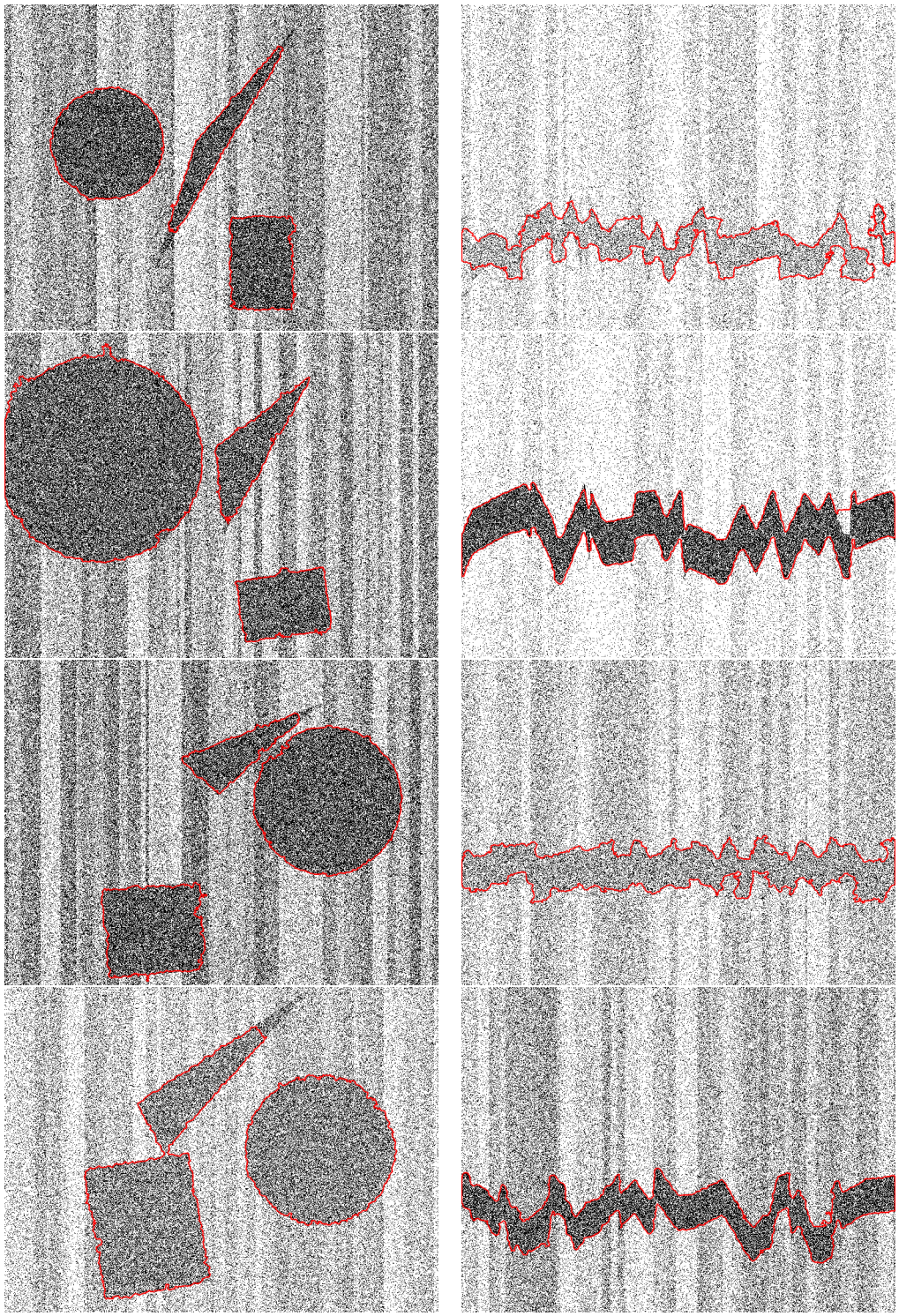


Figure A.1: Examples of the result of a full segmentation obtained by PCDS-Yao for the synthetic clouds.

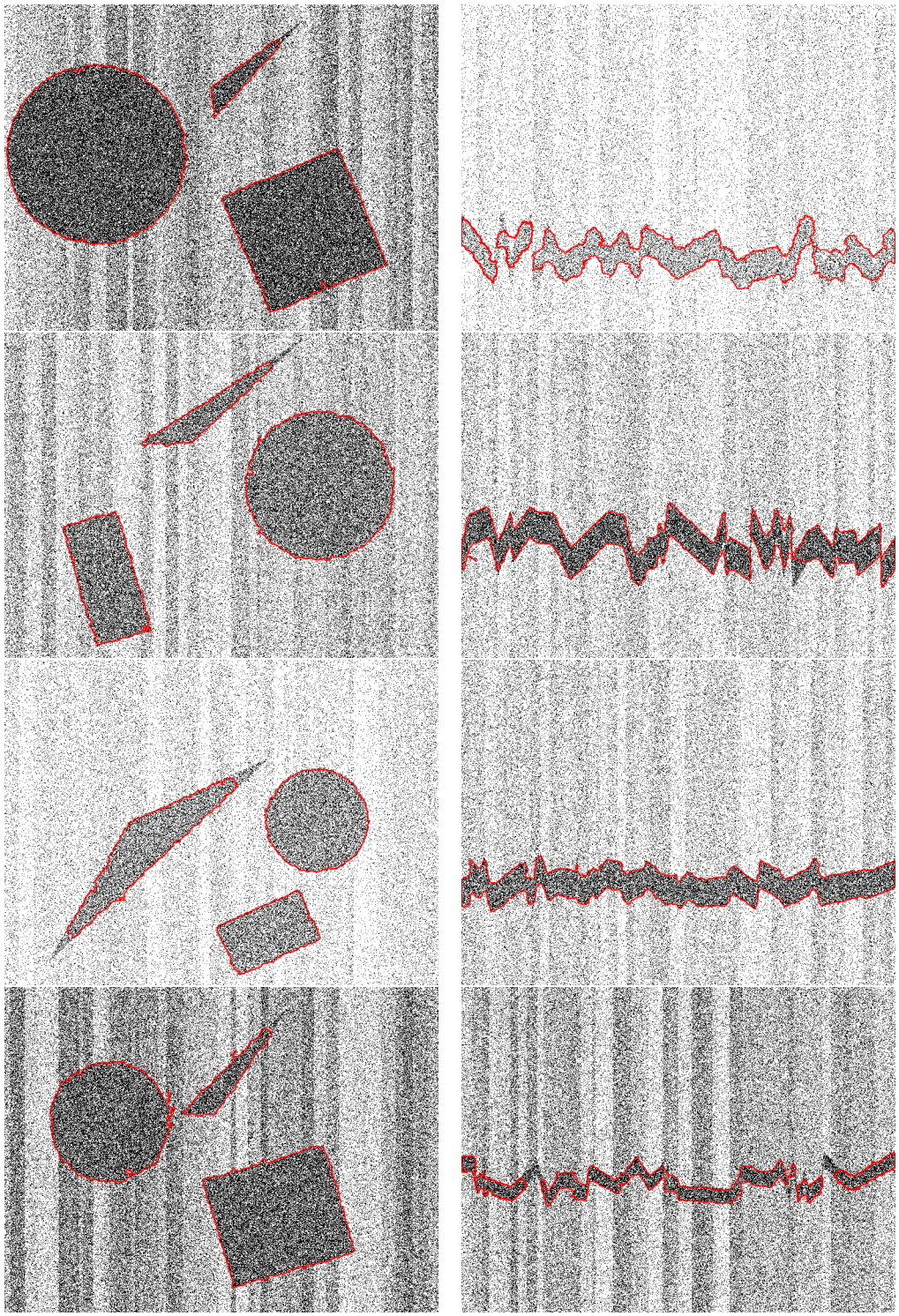


Figure A.2: Examples of the result of a full segmentation obtained by PCDS-Yao for the synthetic clouds.

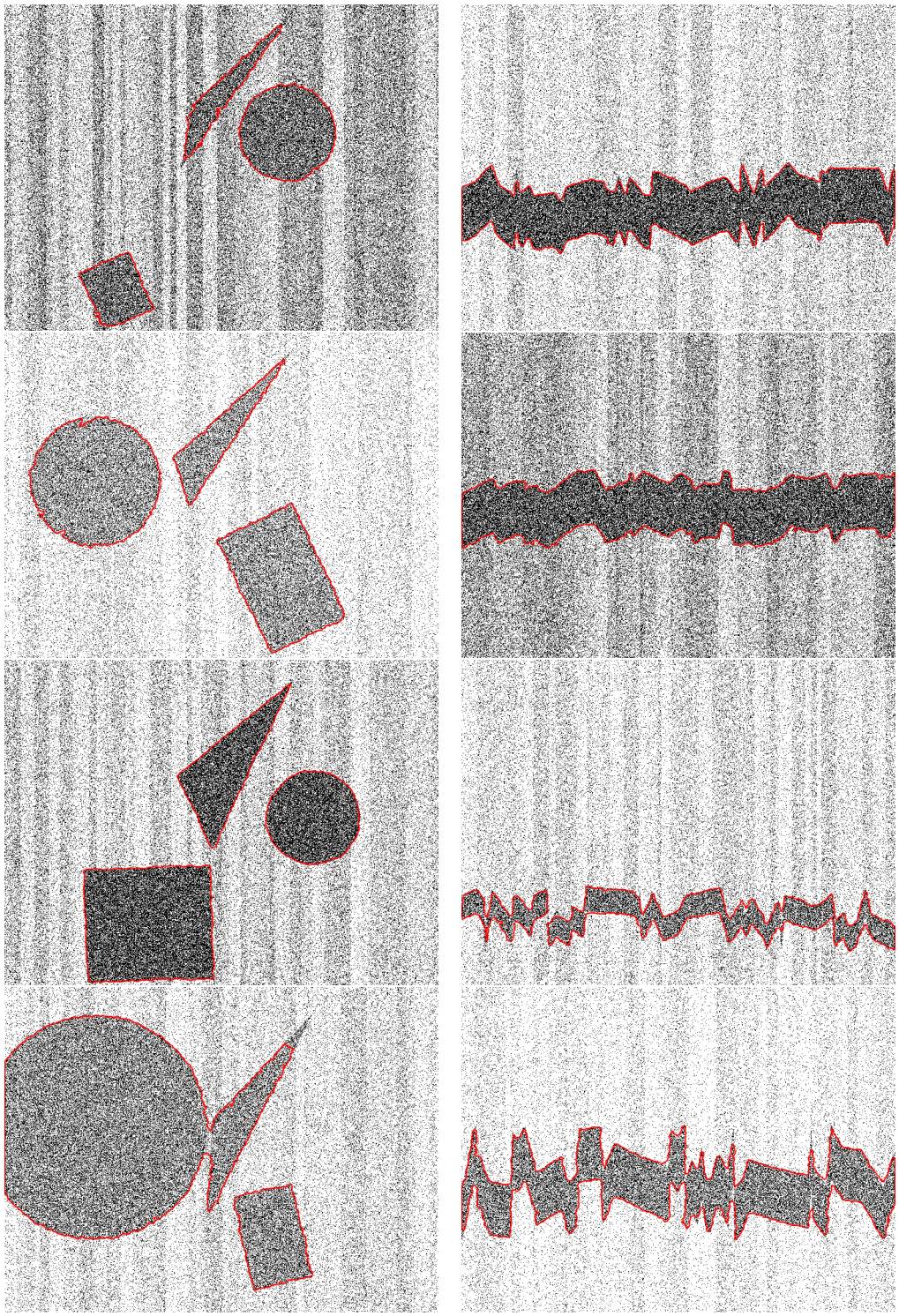


Figure A.3: Examples of the result of a full segmentation obtained by PCDS-Yao for the synthetic clouds.

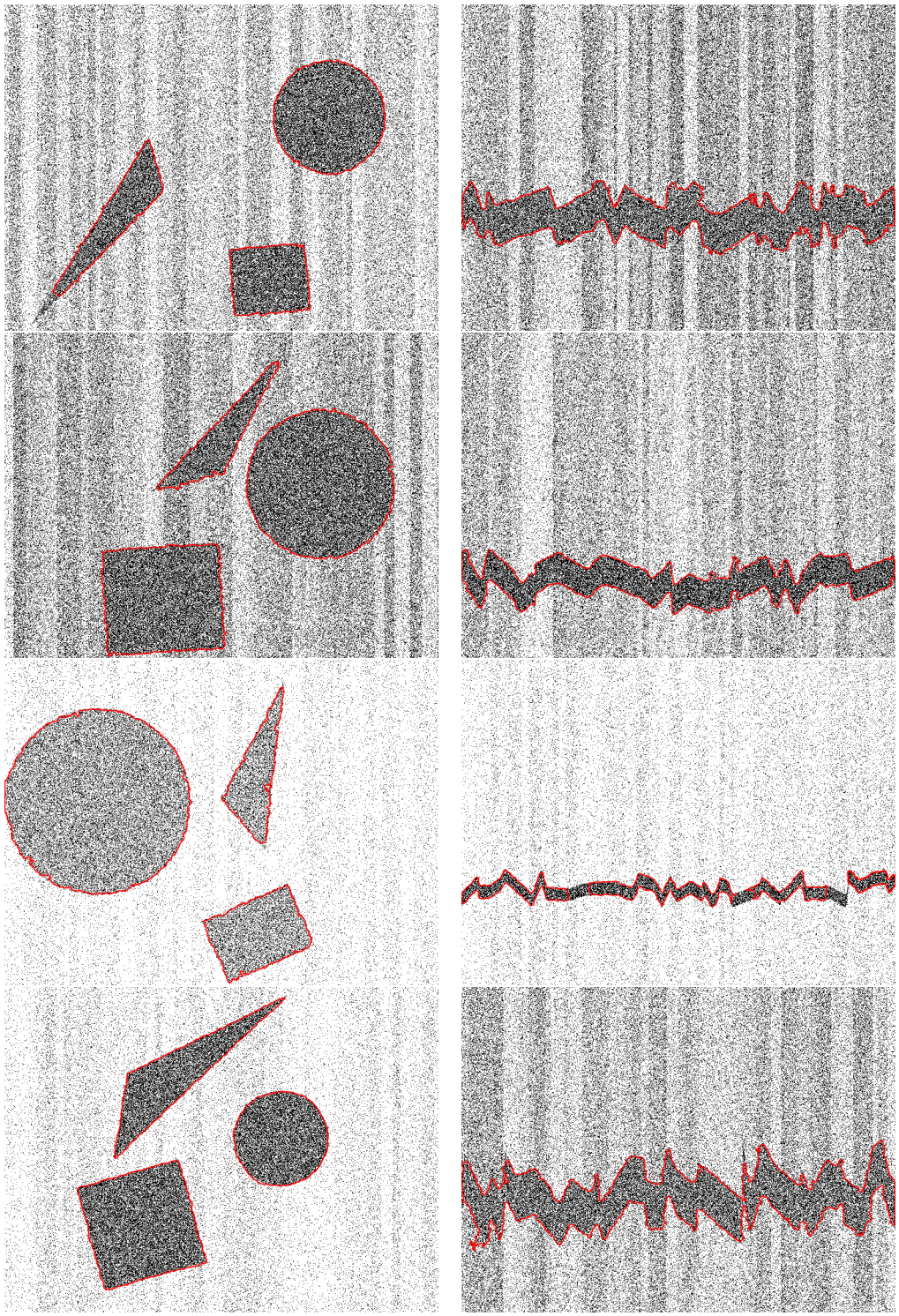


Figure A.4: Examples of the result of a full segmentation obtained by PCDS-Yao for the synthetic clouds.