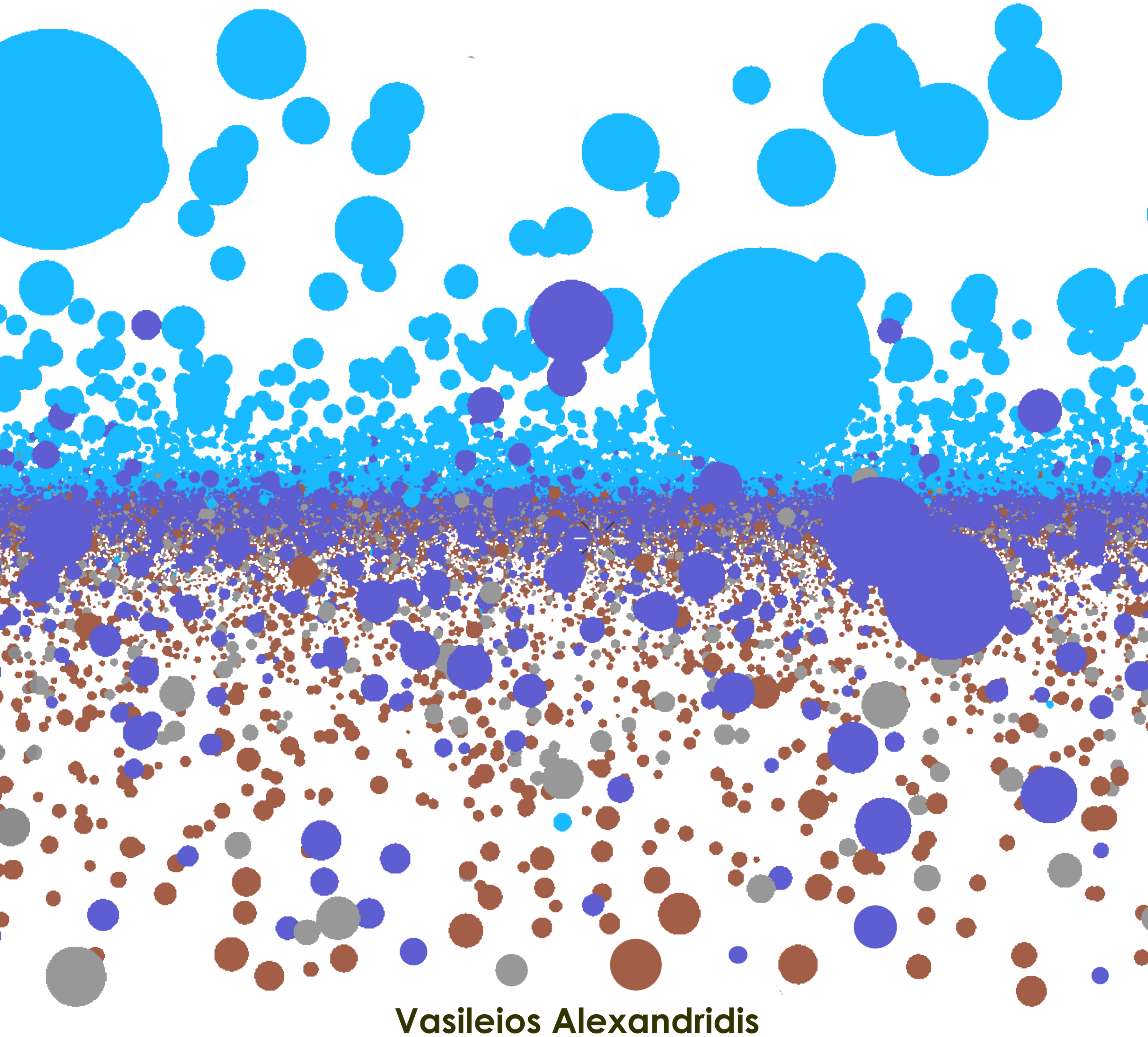
MSc Thesis in Geomatics
Faculty of Architecture and the Built Enviroment

Automatic detection of waterbeds in shallow muddy water-bodies in the Netherlands using green LiDAR



Vasileios Alexandridis
August 2020

AUTOMATIC DETECTION OF WATERBEDS IN SHALLOW MUDDY WATER-BODIES IN THE NETHERLANDS USING GREEN LIDAR

A thesis submitted to the Delft University of Technology in partial fulfillment of the requirements for the degree of

Master of Science in Geomatics for the Built Environment

by

Vasileios Alexandridis

June 2020

Vasileios Alexandridis: *Automatic detection of waterbeds in shallow muddy water-bodies in the Netherlands using green LiDAR* (2020)

The work in this thesis was carried out in the:

3D geoinformation group
Department of Urbanism
Faculty of the Built Environment & Architecture
Delft University of Technology

Deltares

Supervisors:	Prof.dr. Jantien Stoter
	Dr. Ravi Peters
	Ir. Maarten Pronk (Deltares)
Co-reader:	Balázs Dukai

ABSTRACT

Bathymetric Airborne LiDAR technology is used to map the depth of water bodies. It uses a green light sensor which is able to penetrate the water surface and reach the bottom part of the interesting water areas.

However, water conditions affect the capability of the green laser penetration. Factors such as the water clarity, the water turbidity (waves) and the vegetation are some of the crucial restrictions for green light to penetrate the water; particularly in shallow inland water areas.

This research examined the capability of green LiDAR data to improve the bathymetric surveys in case of muddy and shallow inland Dutch water bodies. The potential of green LiDAR increases as the monitoring of water depths is getting easier, faster and more efficiently in terms of cost than manual GPS measurements.

The main challenges of this thesis are concentrated both on the existence of various sparse and dense parts in the point-cloud and on the limitations of the data in terms of quality due to the not ideal water conditions.

Specifically, this thesis presents a workflow with required procedures that aim to process a raw green LiDAR point clouds of water bodies and then classify them into three classes: water surface, underwater and bottom points. Pulse and Neighbourhood based algorithms were implemented in order to perform a classification process with high level of automation. Point characteristics such as intensity, number of returns, return number were analysed per pulse. Voxelization was used as a spatial method to divide the 3D space into water columns (3D Voxels).

The spatial distribution of the water points into the water columns was examined based on different factors such as elevation, density, intensity. By comparing and partially combining those methods the detection process was improved to deal with shallow and muddy water bodies. A classification confidence value was calculated and stored for each potential bottom point. The resulting output is a classified green LiDAR point cloud based on the confidence values. Using elevation, density and confidence values, raster DTMs with multiple bands were created for each water body.

To sum up, this thesis proposed an efficient workflow to process and automatically classify green LiDAR water-body data using both voxel and pulse based methods.

ACKNOWLEDGEMENTS

First of all, I would like to thank all the people who guided and taught me during my graduation thesis. Special thanks to my supervisors Ravi Peters and Jantien Stoter for their guidance, efficient help and productive meetings and feedback, throughout this project. Profound gratitude to Ravi, who never hesitated to offer important help whenever I faced up problems and guided me to the right direction. Also, I would like to thank Edward Verbree for his valuable feedback in the final period of my graduation.

Many thanks to Deltares, and specifically Maarten Pronk, for giving me the opportunity to conduct my graduation thesis in collaboration with them and fulfil my research interests. Maarten has always provided to me detailed and well-understandable feedback. We had many constructive discussions and his existing knowledge on the point clouds' topic helped me completing this thesis.

Furthermore, I want to thank Giorgio Santinelli (Deltares), the manager of green LiDAR project, for his helpful feedback and advices. Also, special thanks to both Giorgio (Deltares) and Maarten (Deltares) for handing me the required collection of green LiDAR data, which have allowed me to do this research true.

Finally, I would like to thanks my parents and friends, who always support and encourage me to do my best and give 100% of myself. Thanks to all my teachers fellow classmates and co-"riders" in Geomatics, for their support and accompany in the entire master program.

CONTENTS

LIST OF FIGURES

LIST OF TABLES

List of Algorithms

ACRONYMS

DEM	digital elevation model	14
DTM	digital terrain model	14
DT	Delaunay triangulation	14
VD	Voronoi diagram	14
GDAL	Geospatial Data Abstraction Library	47
TIN	triangular irregular network	14
LIDAR	Light Detection And Ranging	ix
ALS	Airborne Laser Scanning	2
ASPRS	American Society of Photogrammetry and Remote Sensing	8
ALB	Airborne LiDAR Bathymetric	xv
GLFW	green LiDAR full waveform	xi
DWSM	digital water surface model	19
NWSP	near water surface penetration	20
kNN	k-Nearest Neighbors	34
VFLP	Virtual First and Last Pulse	22
ME	Mean Error	5
SD	Standard Deviation	5
RMSE	Root Mean Square Error	5
NIR	Near-infrared	27
RN	Return Number	9
NR	Number of Returns	9
EEC	exclusive economic zone	28
GDAL	Geospatial Data Abstraction Library	47
saga	SAGA System for Automated Geoscientific Analyses	47
grass	GRASS Geographic Resources Analysis Support System	47
nap	NAP Normal Amsterdam Level	xii

1 INTRODUCTION

The Netherlands, a flat and low-altitude country where four rivers merge into a delta area, has over 50% of its total area protected by dikes against floods. The existence of a well-organized infrastructure with ditches and pumping stations can cope with all the dangers posed by a wet and shallow country (Vázquez et al., 2017). Ditches possess an important role for the drainage of the whole country of Netherlands and their total length is estimated at around 30.000km.

Part of this system is the so called Dutch polder model. The country's water boards are responsible for managing the regional water system, maintaining the water level, protecting the water quality and supervising regional flood infrastructure. In order to control this complex and well-balanced water system with 237.000 km of canals and ditches consisted of sand, peat and clay (Vázquez et al., 2017), accurate bathymetric information plays a vital role in the water management.

The regional water systems that are maintained by water boards consist of muddy and shallow water bodies (water depth around 50cm to 3-4 meters). In order to acquire accurate bathymetric data for these shallow inland water bodies, an efficient and cost-effective way is demanded. A technique such as *echo-sounding* is not suitable because of the shallowness of water or the presence of obstacles above and below the water surface. Another technique is to use the *gauging rod*, but it is preferred only for small area surveys and not for large ones (Vázquez et al., 2017).

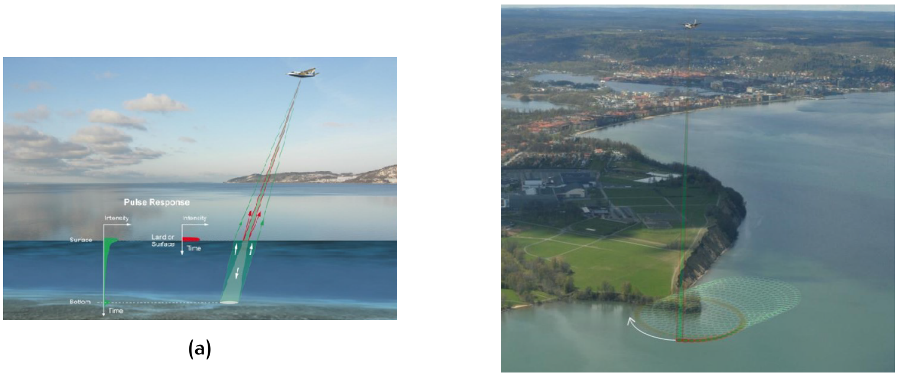
However, the airborne LIDAR bathymetry (ALB) technology has been successfully used in recent decades in deep waters and clear coastal wetted areas (Figure 1.1). This technology can be used to improve the bathymetric surveys in the case of muddy and shallow inland Dutch water bodies. That is why several water boards in collaboration with private companies are examining the potential of bathymetric LIDAR through pilot project. Particularly, green bathymetric LIDAR is used in order to monitor the water depths efficiently, easier and cheaper than manual measurements (i.e. GPS measurements).

The green LIDAR pulse uses a wavelength of 532 nm that propagates into the water and can be reflected from the bottom surface of the water body (Mandlburger et al., 2015). Its laser can penetrate greater depth measurements compared to image based methods. Consequently errors due to shadows or surface disturbance do not exist. Other factors such as sun angle and shining water surface do not affect the ALB and the data collection is not limited only during desired light conditions (Hilldale and Raff, 2008).

These acquired green LIDAR data can in theory be used in order to distinguish the 3D geometries of the waterbeds of the shallow water bodies in Netherlands. That is a complex and challenging problem as the bottom sediments of Dutch water canals are mainly contained dark sands, peat, clay and mud. Many of the sediments contain organic matter which makes them look like dark colour and decreases their reflectivity to the ALB technology. Also, they present high concentrations of chlorophyll during the seasons and usually have a soft layer of sludge that attenuate the laser signal. These general conditions of the water and bottom surface of the shallow water bodies increase the complexity for ALB measurements.

Bathymetric data is essential for applications related to agriculture, floods protection and maintaining water supply during drought periods (Vázquez et al., 2017). The knowledge of water depths in water bodies provides useful information for dredging and water transportation purposes, used to maintain the water quantity and quality of the water.

The complexity of the ALB measurements in combination with the existing need for more bathymetric data make bottom detection from Airborne Laser Scanning (ALS) data still critical problem. This thesis considers the problem of automatically detecting the bottom areas (points) of muddy water bodies from ALS point clouds.



(a)

(b)

Figure 1.1: Bathymetric scanners (AeroData, 2015)

1.1 RESEARCH MOTIVATION

Previous studies have been conducted to detect waterbeds in water bodies using bathymetric LIDAR data in case of Netherlands. However, the developed methods did not succeed to detect bottom points with high certainty and accuracy (see Section 3.2). Further research needs to be carried out by applying other techniques and algorithms.

The high need for both classified green LIDAR point clouds and produced DTM raster outputs with multiple bands containing information such as depth values is certain. Deltares is a company that is currently interested in collecting and analysing green bathymetric LIDAR data, especially for shallow water areas, and then creating automatic processes to effectively get valuable end products (e.g. DTMs). This is why this thesis is carried out in close collaboration with them.

Deltares is a Dutch technological institute for applied research in the field of water, soil and infrastructure. Other institutes and organizations (e.g. Water boards, Rijkswaterstaat) benefit from accurate and dense bathymetric LIDAR data. Pilot projects have been run under the collaboration of Deltares and them. Particularly, aeroplanes flew and collected LIDAR data over the water areas (i.e. shallow and muddy water canals) throughout the Netherlands. Big data acquisitions costs a lot and requires good organization, as it should be done once. However, there is not an automatic operation that using these data detects the waterbeds of shallow and muddy water-bodies in the Netherlands.

Pulse and Neighbourhood-based methods could improve the detection process and deal particularly with shallow and muddy water bodies. Specific point characteristics such as intensity, number of returns, return number can be analysed both per pulse and in the whole dataset area. Also, the spatial analysis of the dataset using voxelization approach could provide promising outputs. Voxelization is a spatial method that based on the creation of volumes (water columns - 3D voxels) using the 3D points in the 3D space of the dataset. Therefore, this research needs to fill this gap by implementing algorithms that enhance the ability to classify the water bodies and then detect the interesting bottom points.

1.2 OBJECTIVES & RESEARCH QUESTIONS

This thesis explores the possibilities of automatically detecting the waterbeds of shallow and muddy water-bodies in Netherlands using green airborne LIDAR data. The main goal is to classify the given dataset of water points into three classes: water surface, underwater and bottom points based on some criteria (e.g. point density, distance between points) This can by done by applying:

- pulse based methods (i.e. taking into account the point's characteristics such as intensity, return number, number of returns)
- voxel based methods

This study focuses on how different spatial algorithms can be used, and how to possibly combine them to come to a final classification. The effectiveness of the proposed method will be assessed in terms of confidence of the classified points; especially for the bottom points. The confidence value of a potential bottom point is determined according to the amount of points (i.e. density) in its local neighbourhood and its distance (i.e. depth) from the water surface point and its intensity value. These values are used to assess how confident is a point to be bottom or not (see reference methodology section). It is not the actual goal to provide a method that will just automatically classify the point cloud, but to provide a more accurate and automatic bottom detection tool. Therefore, the corresponding main research question is:

Question 1. *Can the bottom points of shallow and muddy water-bodies in the Netherlands be automatically detected using ALB?*

As seen in Figure 1.2, this question incorporates the whole process from a raw airborne LIDAR dataset to a final classified point cloud using an automatic procedure. Besides this main research question, it is also significant to investigate the performance of the proposed methods, the properties of the dataset and the visualisation approach. These details are formulated in the following sub-questions:

Question 2. *Can pulse and/or neighbourhood based methods - in a green airborne LIDAR - be used to classify and detect the bottom points?*

Question 3. *What is the influence of different voxel resolutions for classification, in terms of accuracy and computation load?*

Question 4. *How does the various point cloud quality (i.e. density, outliers) affect the classification process?*

Question 5. *Can a confidence value of water points be calculated?If it's possible, how?*

1.3 SCOPE OF RESEARCH

This thesis will not deal with the detection of water courses from an unclassified green LIDAR dataset. The provided datasets (i.e. topo-bathymetric) contain urban structures (e.g. buildings, bridges) and vegetation that are going to filtered them out in the pre-processing step. This will happen by just using the Top10NL dataset with the water boundaries, whereas a hard-coded threshold will be used for the z dimension. Only if necessary, ground filtering methods will be run to extract the ground points which correspond to water-bodies' points in this study (Ledoux et al., 2019).

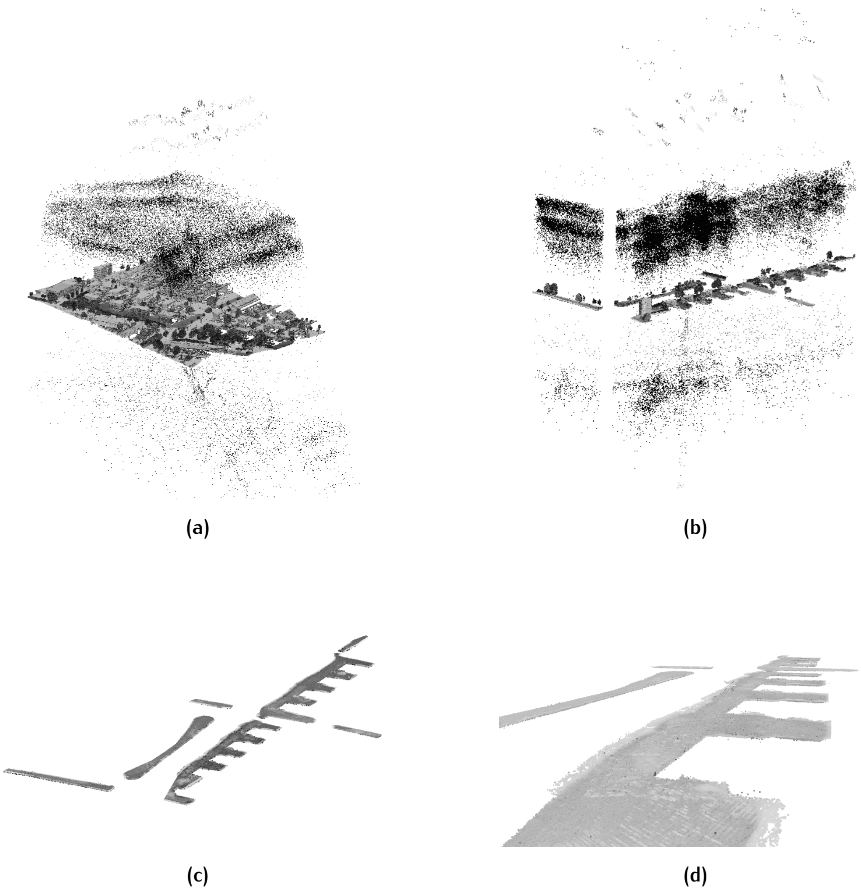


Figure 1.2: Illustration of topo-bathymetric LIDAR data in different stages. 1.2a Raw airborne LIDAR dataset with noisy points (i.e. points in the air space between airplane and ground surface). 1.2b Cropped based on the extent of water canals 1.2c Clipped on specific height level threshold 1.2d Interesting unclassified water areas

Furthermore, this research uses a group of discrete airborne LIDAR points. The point data correspond to high intensities in the waveform of the laser pulse interacting with the water surface and the water particles in the water column. This means that may be 1 up to 4-5 returns per pulse, where each correspond to an individual reflection event from an object on or beneath the water surface (J. Sumnall et al., 2015). The given raw datasets have already been processed by the point cloud collector in order to derive-identify proximal peaks (i.e. returns) from the full waveform. The analysis and interpretation of the full profile of a return signal is complex and demands high computational level. That is why the provider (i.e. the lidar company) does this pre-processing to the derived raw data and then delivers them.

The main focus of this study is to automatically distinguish the bottoms of the shallow water-bodies by applying different methods. By detecting those interested bottom points, the original water points can be classified into the 3 classes: water surface, underwater and bottom points. LIDAR data handling techniques like filtering methods based on the points' characteristics (e.g. return number, number of returns, intensity value) and neighbourhood based methods will be implemented (i.e. voxelization). The later ones take advantage of specific spatial search criteria that are tested on the 3D extent of a voxel.

Also, the research aims to develop an efficient workflow in terms of accuracy and execution time that should be applicable on various green LIDAR datasets in the Netherlands; specifically for muddy and shallow water areas. Important to mention that the input point clouds may have various densities and are not classified at all.

If few ground truth data (GPS measurements) are provided, the validity of the achieved results can be checked by computing statistics. For instance, the Mean Error (ME), the Standard Deviation (SD) and the Root Mean Square Error (RMSE) can be calculated for evaluating the closeness agreement between the observed and resulted data.

The various point cloud quality in terms of density and errors might affect the classification process of the LIDAR data and needs to be investigated. Also, 3D geometry and 2D rasters (DTM surface) can be created using the classified point cloud in order to visually check the geometry and classification of water bodies, but also to validate the results comparing with few ground truth data.

Moreover, this study aims to develop an efficient method that is applicable on large data sets. To get high speed for the computational analysis of the point cloud, all the algorithms are implemented in Julia programming language. It is a novel language that combines the functionality of quantitative environments (e.g. R, Python) with the speed of programming languages like C++ to solve big data and analytics problem. The developed prototype in Julia[1] can be used by Deltares to run further experiments with other LIDAR data sets.

1.4 THESIS OUTLINE

This thesis is structured as the following chapters:

- Chapter 2 provides the necessary theoretical background of this thesis. It covers the fundamentals of LIDAR and especially the green bathymetric ones.
- Chapter 3 introduces and analyses existing approaches in the field of mapping water bodies using LIDAR data. Characteristics and limitations of those methods are described.
- Chapter 4 explains the methodology of this research. It demonstrates the proposed workflow and the conceptual ideas behind it. The challenges and the potential solutions are described.
- Chapter 5 focuses on the implementation and experiments of the proposed methods. It transfers the conceptual ideas to code. The provided datasets and the used tools are described here.
- Chapter 6 presents results, discussions and validations of the proposed method.
- Chapter 7 provides the conclusions of the research. All the research questions are answered here. Conclusions and future ideas are presented.

[1] https://julialang.org

2 THEORETICAL BACKGROUND

This chapter provides an overview of the theoretical background for this thesis. Firstly, the fundamental basics of airborne LIDAR are given in Section 2.1, to understand how the point clouds are captured, which are their properties and the existing errors. In Section 2.2, a further explanation on the use of green LIDAR for bathymetric purposes is given. Moreover, the environmental factors that can affect the green laser pulse's transmission are presented. Following, processing techniques that are used to extract valuable information from a point cloud are described in Section 2.3. Lastly, different techniques that are used in this research to represent the point cloud are discussed in Section 2.4.

2.1 AIRBORNE LIGHT DETECTION AND RANGING

In the mid 60s, the laser scanners (LIDAR) began to be known as they were placed on aeroplanes and used for airborne applications. However, only few decades ago LIDAR became a popular method for acquiring accurate geospatial measurements after the introduction of GPS (Sharma, 2019). It has evolved and recognized as a geospatial technology with many advantages in a variety of applications (e.g. topography, bathymetry, agriculture, archaeology).

LIDAR is a remote sensing method that uses light in the form of a pulsed laser to measure distances of objects on the earth's surface (Sharma, 2020). The light pulses generate the three - dimensional information (i.e. points) about the shape of earth's surface and its surface characteristics. The generated set of points (i.e. point cloud) can contain up to some thousands of points for just a small -few kilometres- area. The quick and automated data capturing make it usable for large areas of the real world.

The three basic components of the instrument are the scanner, laser and GPS receiver. Other elements such as optics have crucial role in the data collection procedure, too. The LIDAR systems are divided into two categories based on their functionality: Airborne & Terrestrial systems (Sharma, 2019). Following, there are two types based on the application purposes; the topographic and bathymetric ones. A topographic LIDAR uses a near-infrared (1064nm) laser to capture the ground, while bathymetric one uses green light (532nm) that can penetrate the water surface and measure sea-floor (Sharma, 2020).

2.1.1 Discrete Return and Full Waveform LIDAR

In a LIDAR system, a laser pulse (light) is emitted from the laser light source and travels to the ground. When it hits on the ground or on other objects (e.g. trees, buildings), an amount of the light energy is reflected and returns back to the LiDAR sensor, where it is recorded (Wasser, 2020). Some part of the energy may continue towards the ground surface and multiple reflections can be recorded from just one light pulse.

Specifically, the system measures the time that takes for the light to hit on an object and return back. That time is used to calculate the travelled distance, and then the distance is converted to elevation. The final measurements are done by taking into account the x,y,z location (GPS measurement) of the light source.

The light energy distribution that returns to the sensor creates a ”waveform”. That amount of returned energy is the so called intensity. Areas with more light energy present peaks in the waveform and often correspond to objects like branches of trees. This return may be recorded in two ways:

- The Discrete Return system identify peaks in the waveform and stores a point (i.e. returns) at each peak location in the waveform curve. This kind of system can record 1 up to 4 returns per laser pulse. (Wasser, 2020)
- The Full Waveform system captures the distribution of returned energy (i.e. waveform) the full profile of a return signal. The handling, interpretation and analysis of the waveform is quite more complex. However, more information may be captured than to discrete system, but more data does not always translate into better information for data analysis (Ussyshkin and Theriault, 2011).

A group of discrete return LiDAR points is known as a *LiDAR point cloud*. In this study all the LiDAR datasets are acquired based on this format.

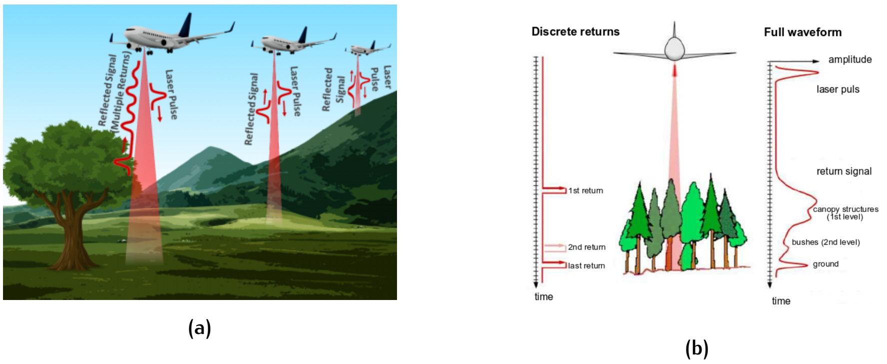


Figure 2.1: Airborne LiDAR systems. 2.1a An emitted laser pulse and the reflected signal (waveform); in case of forest, bare ground and branches of tree (LiDARNews, 2018), 2.1b differences between discrete return and full waveform systems; in a forest application (Ferraz et al., 2009)

2.1.2 Properties of LIDAR

A LiDAR point cloud is usually stored into *.las*, an American Society of Photogrammetry and Remote Sensing (ASPRS) data format. In addition, the *.laz* format has been developed, which is a compressed version of *.las*. It is often used to transfer large amounts of LiDAR data.

The attributes of a point cloud can vary, depending on the collection and process procedure. The point’s attributes are defined in the metadata and they are specified based on their LAS point data format. The LAS standard, currently at version 1.4, is maintained by the ASPRS organisation (more info (ASPRS, 2013)). Attributes such as x, y, z coordinate values and probably intensity are the primary elements of a point. Other point clouds include attributes like classification index, representing the type of an object where the point belongs to (e.g. vegetation, water, building). The classification is a complex procedure and various algorithms are needed to detect the classes. Some LiDAR products may be classified as ground and non-ground, or other may further processed to distinguish the type of infrastructure (e.g. buildings, factories).

A point cloud dataset; specifically *.las/.laz* formats, consists of a *header block* and a *point records* part. The header part is at the beginning of the file and supplemental information (e.g. metadata) are placed there. Then, the point records part contains all the points with their attributes.

The points are stored in an array, where each row represents a point and the columns correspond to each attribute (e.g. x, y, z coordinates). Also, there are other important attributes based on the point data format such as intensity, Return Number (RN), Number of Returns (NR), classification and GPS time. In particular, the intensity value represents the magnitude of the pulse return. This value is usually included in the LIDAR system and stored to the data, otherwise it must be set to zero. The return number is the pulse return number of a pulse. A pulse can have many returns and they marked in a sequence of return. Specifically, the first return has a RN of one, the second of two and so on up to five returns. The number of returns represents the total number of returns for a given pulse. For instance, a point may have RN of two with a total number of three returns. Moreover, the classification represents the *class* attribute of a point. If the points has never been classified is set to zero, otherwise takes number from 1 to a few dozen. Each number corresponds to a class; part of the real world (e.g. buildings, agriculture, water). The GPS time is stored for every points and is the standard GPS time (satellite GPS time) in the new *.las* versions. The original of standard GPS time is defined as midnight of the morning of January 6, 1980 (ASPRS, 2013).

2.1.3 Errors in airborne LIDAR

As a LiDAR system contains various sensor components, many different sources of errors exist. A few important errors are presented in the Table 2.1, as stated in (Lohani, 2010).

	Errors	Description
1	Sensor's position	Due to errors in GPS, IMU and GPS-IMU integration.
2	Laser's angle	The laser instrument is not perfectly aligned with the aircraft (roll, pitch, yaw axis). The scanner angle may have error.
3	Laser's range	Due to time measurement error and wrong atmospheric correction
4	(in) LiDAR data	Due to complexity of an object space, e.g., terrain with steep slopes
5	Multipath (2.2b)	Laser may reflect in specular surface and pulse's direction changes Then, a wrong point is capture (see Figure 2.2b))
6	Refraction(2.2a)	The laser travels in the air and encounters surface like water. Part of it passes from one medium (air) into the other (water). Then, it continues on a new straight path

Table 2.1: Errors during the collection of LiDAR data

Two of the most significant errors are the multipath reflection and refraction effect. The first is caused when the laser pulse may reflect in a surface with low or high reflectance (A) (e.g. wall of a building), then the pulse direction changes and hits other object(B). This will result in a point measurement which was never measured by the LiDAR, thus error in the LiDAR data (see Figure 2.2b). The second, refraction effect (see Figure 2.2a), is caused when a part of the light array is reflected on a surface (water) and returns back to the atmosphere, while the transmitted light ray bends (refraction). In addition to the refraction of the laser beam, the slowdown effect (as the speed of light in the water is smaller) occurs. As a result, the underwater point (C) is written with incorrect coordinates (B) in the point cloud file.

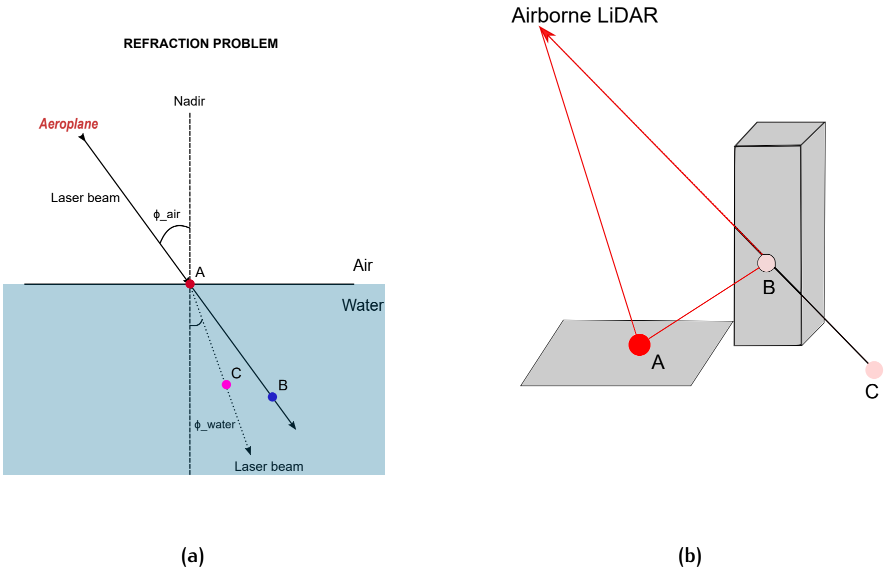


Figure 2.2: Laser scanning errors. 2.2a Refraction and slowdown effects where the underwater point (B) is written with incorrect coordinates, while it should in position (C). 2.2b Multipath effect where a laser pulse is reflected of an object (B) before it returns to the scanner. The travelled path is way longer. This results in an another point location below ground surface (C).

2.2 AIRBORNE BATHYMETRIC LIDAR

The technology of airborne LiDAR for mapping terrestrial topography has grown significantly the last decades. However, the use of airborne lasers to measure bathymetry is both promising and challenging. Bathymetric LiDAR operates in a way that is similar to topographic LiDAR. It emits green laser beams with a wavelength: of 532 nm (see Figure 2.3). This wavelength penetrates the water and gets reflected by the bottom surface in the water. The bathymetric LiDAR is also called *green LiDAR*.

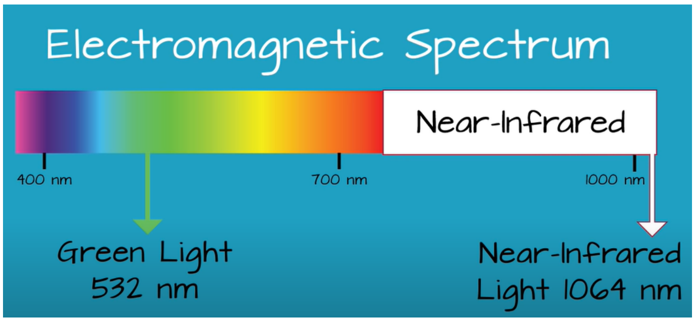


Figure 2.3: LiDAR wavelengths in the visible and near-infrared regions of the electromagnetic spectrum (Wasser, 2020)

2.2.1 Operational properties

The ALB system uses a high energy laser pulse with wavelengths in visible green and sometimes also near-infrared (1064nm) in order to detect both the bottom sediment and the reflection from the water surface (see Figure 2.4). The green wavelength is preferred as it is capable to penetrate the water surface and may reach the bottom, while the near-infrared is used to get measurements both around the topography of coastal waters and water levels (Vázquez et al., 2017). Particularly, ALB allows collecting data in areas, that usually are not suitable for sonar technology due to the shallowness of the waters or underwater obstacles.

Many limiting factors in ALB can influence the strength and shape of the laser pulse that returned to the aircraft, and therefore the data collection procedure. In bathymetry, the laser beam needs to pass through the atmosphere, the air/water interface and the water column both in the transmitted and returned way (Vazquez, 2017b). For instance, the water can attenuate the light energy with depth by causing absorption or scattering effects to the light. As seen in Figure 2.4, the laser travels in the water column and expands in a cone form due to the scattering effect by the water particles.

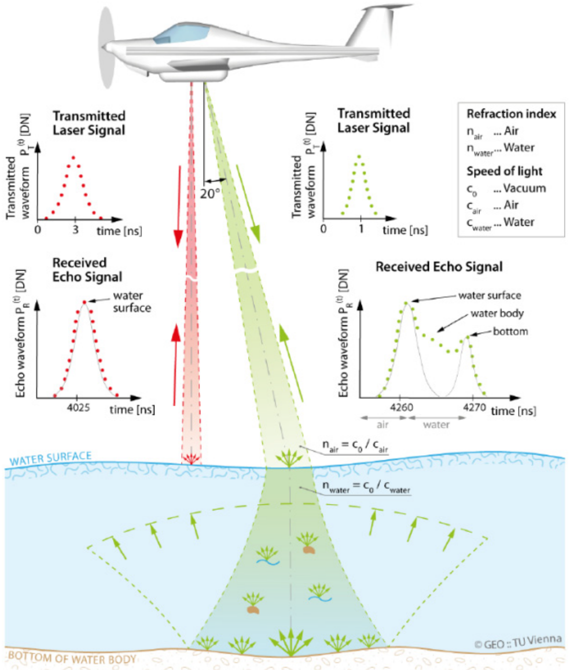


Figure 2.4: Airborne LiDAR bathymetric in operation (Mandlburger et al., 2015)

2.2.2 Environmental factors

Many ALB systems use only green LiDAR as it is suitable for the waterbed detection of water bodies. Even if the green LiDAR wavelength can penetrate the water surface of water areas and can potentially reach the bottom part, many factors can negatively influence the direction, strength and shape of the returned laser pulse to the aircraft. For instance, the laser pulse's transmission is affected by various environmental conditions (e.g. water clarity, suspended sediments, organic particles, water turbidity (waves), vegetation) (Guenther et al., 2000). Also, the composition and the roughness of the bottom play important role on the reflectivity of the laser beam.

- **Water clarity** is one of the most important factors that limits the depth penetration of the laser pulse. It causes absorption (energy reduction) and scattering of the pulse. The water clarity can be measured using Secchi depth method, which is the depth at which a standard black and white disk is lowered into the water until no longer can be seen by the observer.

- **Organic particles & Suspended sediments** increase the scattering effects of the laser pulse. The amount of organic materials in the water and the quantities of the suspended organic and inorganic particles influence the redistribution of the laser pulse's energy (scattering) back to the airborne receiver. For example, the presence of mud over the bottoms of water-bodies cause the absorption of the laser signal rather than the reflection (Vazquez, 2017a).

- **Water turbidity (waves)** increases the backscattering effect and causes the lack of bottom returns depending also on the season of the year when the flight done, as the turbidity varies (Vazquez, 2017a).
- **Vegetation under water** affects negatively the ALB measurements because they can block the laser pulse to reach the bottom sediments.

2.3 PULSE & NEIGHBOURHOOD BASED METHODS

As mentioned in Chapter 1, accurate bathymetric data for large water areas are essential for the Dutch water managers to maintain the water quantity and quality of the water canals. Thus, the acquisition of bathymetric green LiDAR data can primarily be used to detect the waterbeds of water bodies. Indeed, this detection procedure demands further analysis and processing of the point cloud points through spatial algorithms. Many algorithms operate on a single discrete point without using any context information. That makes them ineffective for distinguishing any features in the point cloud. Thus, there are algorithms that take into account the neighbours of a points. Those algorithms can be based either on the grouping of points per pulse or on geometric attributes of points in a local neighbourhood (e.g. sphere, cylinder or water column). These methods are also called *grouping LiDAR data handling techniques* and commonly used to compute a variety of LiDAR metrics (Koma, 2017). Two significant techniques are:

- **Pulse-based method**: uses specific attributes (e.g. return number, number of returns) of points in order to group them into a pulse form; knowing that the dataset is sorted already by the GPS time. This method tries to reconstruct the original pulse (full waveform) that has been discretized into multiple points.

  Particularly, a water surface point usually present *Number of Returns* bigger than 1 (i.e. a sequence of multiple points) and *Return Number* equal to 1 (i.e the first point of the laser pulse). The corresponding underwater points needs to follow up the water surface in the file with *Return Number* and *Number of Returns* bigger than 1. Also, there are few exceptions such as the case where a pulse contains just one point (e.g. RN = NR = 1), which may correspond to the water surface.

Index	Return Number	Number of Returns
6	2	2
7	1	1
8	2	2
10	1	3
11	2	3
12	3	3

Table 2.2: Water-surface point (blue) and underwater points (red)

For instance, the 10th point has RN=1 and NR=3 indicating that could be a water-surface point. The following 11th point has RN=2 and NR=3 and is a corresponding underwater point, while the 12th point has RN=NR=3 and probably is a bottom point.

- **Neighbourhood-based method**: uses a defined search shape in a neighbour points. The search shape can be a sphere, cylinder, a voxel or nearest neighbours. Within the neighbourhood, it is possible to derive descriptive or spatial characteristics for the features within the shape.

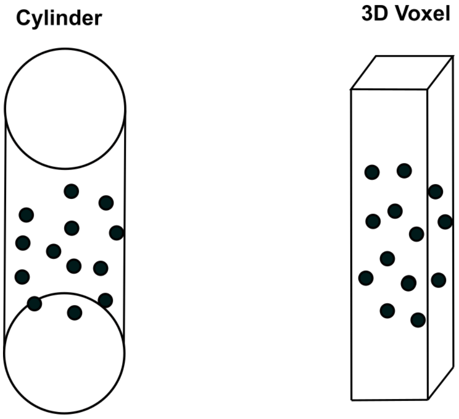


Figure 2.5: Cylinder and 3D Voxel; group of points are located in them

2.3.1 Voxelization

Voxelization of LiDAR data is a process where the entire point cloud can be divided into a collection of 3D regular cubes, which can be called *voxels*. Each point of the point cloud is allocated to 3D voxels, and voxel values are assigned based on the attribute values of the LiDAR point inside the corresponding voxel (Wang et al., 2018).

An Axis-Aligned Bounding Box (AABB) is used to define the 3D extent of the point cloud, where:
AABB = $(x, y, z)|x_{min} <= x <= x_{max}, y_{min} <= y <= y_{max}, z_{min} <= z <= z_{max}$, where ($x_{max}$,$y_{max}$,$z_{max}$) and ($x_{min}$,$y_{min}$,$z_{min}$) are the maximum and minimum values of the bounding box, respectively. The AABB can be divided into uniform 3D voxels based on the *voxel resolution*.

The *voxel resolution* is the most important parameter during the voxelization of a point cloud. If the resolution is high, the number of voxels that contain no points become larger. But if the resolution is low, then more points fall into a voxel and the loss of information is increased. In order to reduce the redundancy and the information loss, an appropriate resolution should be selected. If the LiDAR data are well-distributed and form a regularized grid, then the horizontal resolution can be determined from the equation $\Delta x = \Delta y = \sqrt{A_{xy}/n}$, where δx and Δy are the voxel resolution in x and y axes respectively, and A_{xy} is the horizontal projected area of the points. The vertical resolution Δz is determined by the equation: $\Delta z = min[\sqrt{A_{xz}/n}, \sqrt{A_{yz}/n}]$, where A_{xz} and A_{yz} are the projected areas of the points in xz and yz planes.

Based on the voxel resolution, the bounding box (AABB) is divided into rows (r), columns (c) and layers (l) and they will be stored into a 3D array. The LiDAR points are distributed to the voxels using the formulas

$$r_i = [\frac{x_i - x_{min}}{\Delta x}], c_i = [\frac{y_i - y_{min}}{\Delta y}], l_i = [\frac{z_i - z_{min}}{\Delta z}] \tag{2.1}$$

Afterwards, descriptive and spatial statistics can be computed for every voxel in order to classify the LiDAR point cloud into water surface, underwater and bottom points (Fig. 2.5). In particular, the number of points that fall into each voxel can be summed, while the minimum, maximum,mean and the standard deviation of z values can be calculated (Habel et al., 2018).

Moreover, skewness and kurtosis of the data can measured in order to indicate points' distribution inside the voxel. Skewness is a measure of symmetry, or more specifically, the lack of symmetry. The distribution of the data is symmetric if it looks the same to left and right of the centre point in a histogram. Kurtosis is a measure of whether the data are heavy or light tailed relative to the normal distribution. High kurtosis means heavy tails (or outliers), whereas low kurtosis indicates lack of outliers.

2.4 3D & 2D DIGITAL REPRESENTATIONS

Airborne LIDAR constitutes a tool that can also provide accurate digital terrain model (DTM) of the earth's surface. Using a big-finite number of points-measurements with specific accuracy we can approximate the 3D surface of an area by creating its 3D model. DTM are popular due to the applicability in many fields such as surveying, landscape architecture, agriculture, road design etc. Several techniques have been used to derive DTMs with LiDAR systems, such as Delaunay triangulation (DT) and Voronoi diagram (VD).

DT or triangular irregular network (TIN) and VD are fundamental data structures for the 3D representation of terrains and for their processing (e.g. interpolation) (Ledoux et al., 2019). Additionally, a point cloud can be represented as a 2.5D grid ('rasterized') and then exported to a raster image (e.g. geotiff).

2.4.1 TIN triangulation

TIN or DT is a vector-based representation of the physical land surface or sea bottom. It is consisted of irregularly distributed nodes and edges that arrange a network of triangles. The triangular network of vertices (3D points) connected with edges form a triangular tessellation. A TIN used to represent terrain is called digital elevation model (DEM).

The vertices of TIN are formed by the points of a LiDAR dataset, creating the 3D representation. An advantage of using TIN over the rasterized DTM is that the points of the TIN are represented with a vertex in the model. However, TIN may suit less than a raster DEM for GIS applications with analysis of surface's slope, tends to be more expensive to build and process and processing tends to be less efficient due to its complex data structure (ArcMap, 2016).

TIN can be based on the DT, which conforms triangles to certain constrains. This happens because the output TIN may have long, thin triangles which are not desirable. The resulting triangulation satisfies the DT criterion, if no vertex lies within the interior of any of the circumcircles of the triangles in the network. If this criterion is satisfied everywhere in the triangulation, then the minimum interior angle of all triangles is maximized. As a result, the long and skinny triangles are eliminated (see Fig. 2.6).

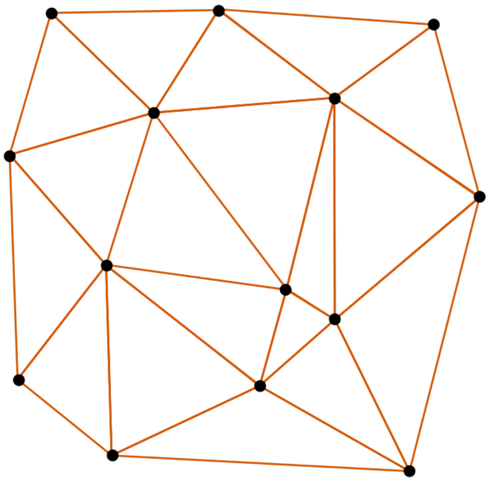


Figure 2.6: Triangulated irregular network; a set of irregularly distributed points

2.4.2 Rasterization

Rasterized LIDAR data provide valuable output products using points' attributes such as elevation, intensity or classification indexes to represent the space in a 2D model. A raster model is an array of cells (or pixels) organized in rows and columns. The value of a pixel can represent a discrete or continuous phenomenon that corresponds to an actual location in the earth. For instance, a pixel with specific elevation corresponds to a real place on earth (Guiotte et al., 2019).

The structure of raster data is simple and really useful for a wide range of applications. Rasters are suited for representing data that change dynamically across a surface area. They can store the continuity of data as a surface, such the previous example of elevation data. The advantages of raster are the simple structure of the format (row,columns), the further spatial and statistical analysis, the ability to store uniformly different types of data (e.g. points, lines, polygons) and the fast way to represent various complex datasets into different band layers. However, in contrast with TIN or DT, the exact point locations will be not retained in the raster cells. Thus, some spatial inaccuracies will exist in the raster structure due to the raster cell too. Raster datasets can be very large in size as the resolution increases and the pixel size decreases. There is also potential loss of precision as the data structured into a grid-cell format.

The space partitioning of LIDAR data into a regular grid works the same as the 2D rasterization of data previously mentioned (see 2.7). The spatial resolution or voxel size (corresponding to pixel) affects the voxel structure. Many geographical features will not be retained when a large voxel size is chosen and other voxels may not have any features. On the other side, a small voxel size may contain too less information (points) to extract valuable results for the contained points.

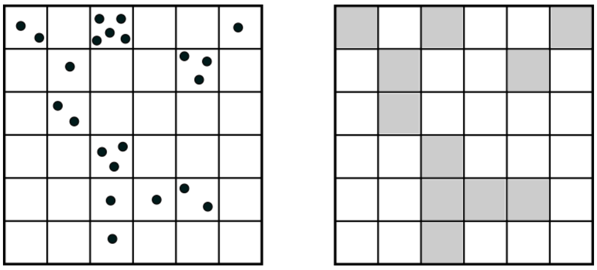


Figure 2.7: 3D points irregularly distributed into a grid, then stored into a raster structure

Moreover, some rasters have a single or multiple bands (i.e. layers of data). Each band is represented by a single matrix of cell values. In case of multiple bands, the raster contains multiple matrices with same spatial extent that represent the same spatial area. For instance, a DEM is a single-band raster, where each cell only one value representing surface elevation. When there are multiple bands, every cell has more than one value associated with it (e.g. elevation, classification codes).

In this thesis, I may use rasters with pixel (voxel) size to represent the bottom surface and do further analysis on the raster bands. Multiple bands may be used in order to associate points' characteristics in different bands. Also, TIN representation might be used to visualize the water-bed's surface into a 3D model.

3 RELATED WORK

Several studies have been done in the field of mapping river and shallow water body bathymetry using green LiDAR. Green LiDAR is widely used for coastal surveys. Many studies focus on water depth measurement quality and limits, in particular for rivers and surface water, and take advantage of this technology. Some of them are concentrated on the mapping of the water-bodies acquired from LiDAR data either green or green with additional ones (e.g. near-infrared (NIR)). Then, methods are applied to classify the water points into water surface and bottom points.

Some studies only deal with the detection of the water regions from a point cloud. This is not part of this study, but the implemented methods (e.g. fuzzy logic concept) give insight for the classification part of the proposed methodology of this research. This chapter describes other studies for mapping water bodies using LiDAR data in Section 3.1, using only green in Section 3.1.1 or additional data in Section 3.1.2. Thereafter, previous bathymetric surveys for Dutch shallow inland water bodies are presented in Section 3.2. In Section 3.3 methods to filter and classify points are thoroughly described. This chapter ends with a conclusion Section 3.4 to overview the methods that match with the goal of the thesis and form the gaps of the research.

3.1 CLASSIFYING WATER BODIES USING LIDAR DATA

In general, the methods for mapping the shallow water-bodies can be separated based on the use of LiDAR data. Either only green LiDAR or green combined with additional LiDAR data (e.g. NIR).

3.1.1 Using only green LiDAR

Throughout the year, many studies have been developed in order to analyse the 3D geometry of water bodies. The detection of the water surface and subsequently the bottom parts using only green LiDAR data.

Allouis et al. (2015) introduced a specific green LiDAR full waveform GLFW model in order to detect the minimum depth detectable estimation H_{inf}. The H_{inf} assessment is done on four steps: 1. *GLFW modelling* for a specific set of LiDAR system and river parameters, 2. *bathymetry estimation* from the GLFW model with approximation methods, 3. *determination of minimum H_{inf}* from the approximated model, 4. *confidence interval* computation using the Monte Carlo method.

This methodology is targeted on low deep waters and only focuses on the usage of surface and bottom returns from the green LiDAR signal in order to determine the minimum depth. In order to develop a suitable model for rivers, Allouis et al. (2015) did some assumptions. The water surface has a non-zero slope in the longitude, the turbidity of water is homogenous without algae and the roughness of the water surface is uniform. Afterwards, using the GLFW (see Figure 3.1) the bathymetry H_{inf} from peaks detection of water surface and bottom parts was estimated using an approximation method. The method is based on a combination of Gaussian laws fitted in a least squares algorithm. The water depth $\hat{H}$ from the model can be estimated.

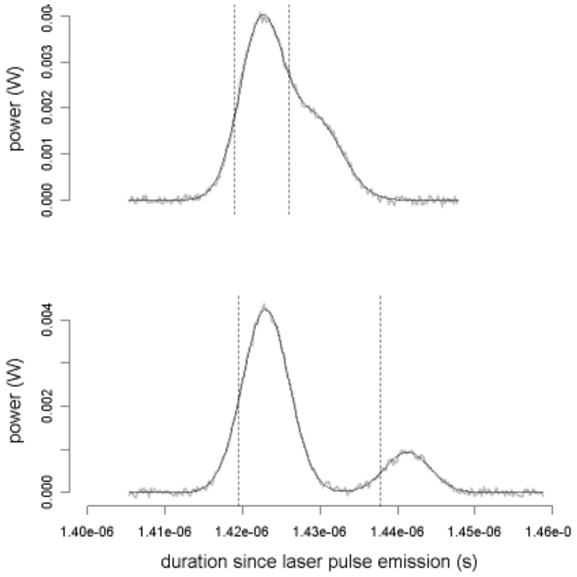


Figure 3.1: A GLFW model using fitted Gaussian density functions; Actual surface position (left) and bottom position (right) are the vertical dotted lines (Allouis et al., 2015)

Mandlburger et al. (2015) proposed a method to determine the water surface using only the available reflections from green LiDAR, and then classify the water echoes. The classification of the water points into three classes: 1. *surface* 2. *body* 3. *bottom points* relied on full waveform features and spatial features based on local neighbourhood. When the green laser interacts with the water, the signal either reflects back due to the water surface or scatters at water particles (i.e. backscatter). The classification procedure is difficult particularly in deep waters. The density of points near to the bottom of the river drops significantly as the laser pulse passes through the water column. The low point density in the bottom part causes misinterpretation of the volume backscatter echo as a ground point.

The surface returns are identified by their low reflectance values and small water depth. Points that have higher reflectance near the water surface and their majority belong to a spherical 3D neighbourhood are classified as water surface, too. The backscatter points are detected by their low reflectance and sparse point spacing.

To avoid misclassification of water points, especially the water bed points at the maximum depth, three thresholds were used: 1. *water depth* 2. *object reflectance* 3. *neighbourhood definition*. The water depth threshold is related to the maximum penetration depth based on Secchi depth of the water body. The Secchi depth measurement is done by a circular disk 30cm in diameter. This disk used to measure the turbidity of the water by lowering it down in the water. The depth where the disk is no longer visible is taken as measure of transparency. Also, the reflectance threshold depends on environmental conditions such as the attenuation with atmosphere and the water column. The neighbourhood threshold is used for the dimension of the search (e.g. spheres, cylinders). The dimension of the search shape differs based on the overall point density of the point cloud. As seen in Figure 3.2, the LiDAR points were classified based on their reflectance value, their distance from the water surface (i.e. water point's depth) and their spatial distribution.

Also, surveyed reference points were measured in order to validate the classified point cloud. Those field points were categorized into classes such as river bed, road, vegetation, forest, as all these categories exist in the study area. Then, the height differences between the LiDAR and reference points were calculated and accuracy was estimated. The height discrepancies were ranged between few centimetres.

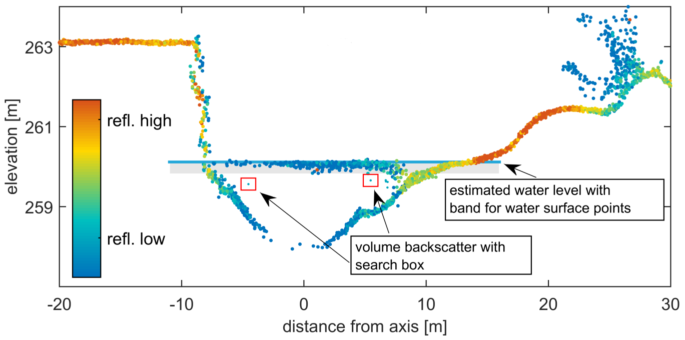
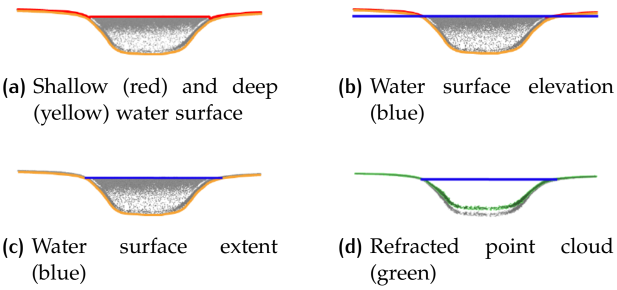


Figure 3.2: Classification of water echoes based on reflectance value, water depth and spatial distribution (Mandlburger et al., 2015)

Another study on the classification of water surface and bottom points was done by Andersen et al. (2017). Specifically, a methodology to process green LiDAR data in tidal environments was proposed. The water surface detection is based on determining depth and surface extent by creating a digital water surface model (DWSM). The water surface elevation is acquired by the water surface points, while the extent is determined by inferring the intersection of water surface and the surface topography.

In particular, a 2D grid was adjusted on the point cloud extent with 0.5m cell size. Then, the shallow and deep surface were extracted from the point cloud by selecting the highest point and lowest point per grid cell, respectively. The shallow surface displays the topography along with the water surface, whereas the deep surface displays the topography and the riverbed (if it is reached by the laser). Also, the shallow surface was used to determine the actual water depth. In order to better simulate the water level, the average elevation of 2m cells was used as the estimated water surface elevation. Thus, the water points were classified into water surface and bottom ones, while they have been corrected from the refraction effect. All these steps appear in Figure 3.3.



(a) Shallow (red) and deep (yellow) water surface

(b) Water surface elevation (blue)

(c) Water surface extent (blue)

(d) Refracted point cloud (green)

Figure 3.3: Water surface and bottom detection. 3.3a Extract both surfaces using 0.5m grid size. 3.3b Determine the water surface elevation. 3.3c Determine the water surface extent. 3.3d Refract the point cloud. (Andersen et al., 2017)

Kinzel et al. (2013) proposed algorithms that used to detect the location of a water surface and bottom return in waveforms for shallow depths (<1m) and deeper depths. The elevation of the bottom is found by firstly correcting the refraction effect and taking into account the slowdown effect in the speed of light. Also, the attenuation of laser pulse and the backscatter effect due to the water turbidity.

Thus, after correcting the waveform from these effects, the first algorithm searches for the most significant peak (maximum peak (MP)) which corresponds to the bottom return in waveforms returned from shallow waters (¡1m). Also, the last peak (LP) algorithm, like the MP, first corrects the waveform and then finds all inflection points in the waveforms (using 1st derivative), and then selects the last peak based on a threshold value. This reduces the change to select a stronger peak due to the turbidity of the water instead of a weaker bottom reflection.

3.1.2 Using additional LiDAR data

The way to acquire water depth measurements using bathymetric LIDAR data is to compute the two time moments between the surface and bottom returned peaks, as seen in Figure 3.4b). Using only green LiDAR on very shallow waters (<2m) is quite challenging and difficult to extract the water surface and bottoms positions, as they are typically mixed in the green signal (Allouis et al., 2010). This means that the two peaks (water surface, bottom) are so near in the time frame, only few ns time difference, that makes hard to easily distinguish them.

For depths lower than 2 meters, the difficulty is the discrimination of the two mixed peaks. For that reason, Allouis et al. (2010) proposed the use of two methods: the first one used green and near-infrared (NIR) wavelength (1064 nm) and the second the red wavelength Raman signal (647 nm) in a regression model based on principal component analysis (PCA) in order to estimate the depth. In the first one, the NIR wavelength is reflected on the water surface, and thus it is easy to distinguish dry land from water surface. In the second one, the Raman wavelength is useful to locate the air/water interface when facing incorrect surface detections due to land reflection or to undesired targets such as birds. Consequently, both NIR and Raman signals helped to accurately measure the water surface position and water column, respectively in the methods (see Figure 3.4). Also, if depth measurements are missing in muddy shallow waters in green waveforms, Raman signal can be used instead.

The NIR and green method gave satisfactory results without using reference data, while the method based on Raman regression model needs large amount of reference data.The Raman signal is sensitive to water characteristics like temperature, turbidity.

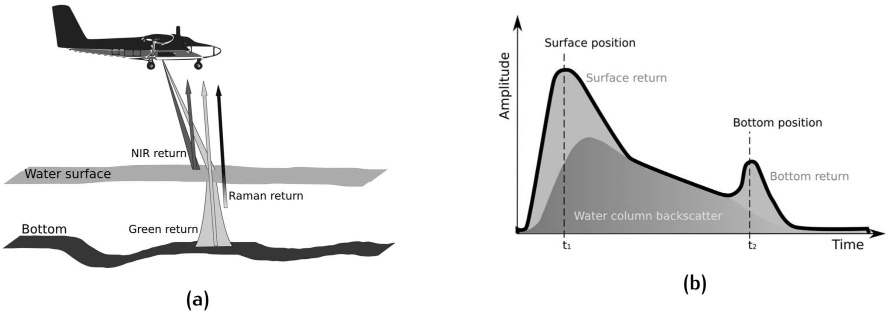


Figure 3.4: 3.4a Bathymetric LIDAR with green, NIR and Raman wavelengths. 3.4b Green LIDAR waveform showing the surface and bottom position. (Allouis et al., 2010)

Zhao et al. (2017) proposed a method to accurately detect water surface and water bottom heights combining green LiDAR and corrected by the near water surface penetration (NWSP) model. The NWSP is the phenomenon where the first return can not exactly correspond to the water surface but reflects a penetration level in the water column. However, the use of integrated infrared (IR) and green LiDAR solve this phenomenon and improves the accuracy, but it cost a lot and adds extra weight to the ALB system. That's why only the green LiDAR is preferred to be used. In this case, if the NWSP model can be accurately estimated, the green LiDAR can obtain accurate measurements. The model can be build using LiDAR and hydrological ground truth data and then apply further statistical analysis.

$$\Delta d = \beta_1 \phi + \beta_2 \phi^2 + \beta_3 F + \beta_4 F^2 + \beta_5 Turb + \beta_6 Turb^2 + \beta_7 \tag{3.1}$$

The NWSP model can be expressed as an equation (see Eq.3.1), where different parameters are taken into account. Particularly, the water turbidity (*Turb*), the incident angle (ϕ) of laser beam to the local vertical and the laser spot diameter (F) on the water surface. Also, it is important to mention that the results of this method are affected by the given reference water surface height data (IR data) and the water turbidity.

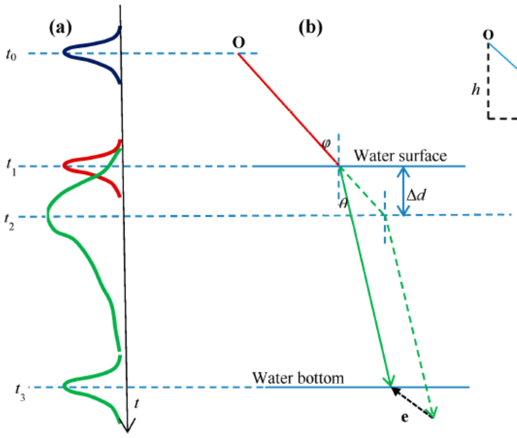


Figure 3.5: Bathymetric LiDAR principle; red and green colours represent the infrared (IR) and green laser pulses, respectively(Zhao et al., 2017)

In Figure 3.5, the propagation way of the laser pulse with green and IR wavelengths is presented. At the time t_1 the IR laser hits the water surface, while the time t_2 the green laser. The water bottom points (time t_3) are captured only by the green laser. The time delay Dt_{12} between the two lasers is calculated using ϕ: incidence angle in the air, θ: refracted angle in the water and Δd: vertical distance between the water surface height from green laser and measured water surface from hydrological data.

3.2 BATHYMETRIC SURVEYS IN SHALLOW DUTCH INLAND WATER BODIES

Bathymetric data have a crucial role for the water managers and Dutch water boards as they give valuable information to control groundwater level in the country (see also in Chapter 1). The knowledge of water depths especially in water bodies is useful for dredging and water transport purposes (Vazquez, 2017a). In order to obtain these data, there is not any effective and cost-efficient method. Even echo-sounding is not suitable everywhere due to the shallowness of the water (<1m) and the presence of under water obstacles.

On 2015, the waterboards collected an ALB dataset with data acquired from both near-infrared and green laser pulses (AeroData, 2015). In order to validate the results, water depth measurements were done from a boat that was navigated in the shallow water canals. Factors such as the presence of vegetation in the water and sludge as bottom sediments block the laser signal to reach the real bottom. Also, it was hard to discriminate between the two peaks in the waveform in case of really shallow and muddy water bodies.

Another study was conducted in 2017 (in a laboratory (Vazquez, 2017a)). Vazquez (2017a) tested the use of yellow wavelength (590nm) for lower attenuation in the waters in combination with short laser pulses for shallow measurements and Single-Photon Counting or Supercontinuum lasers in laboratory conditions. The main goal of this research was to develop a new technique for bathymetric measurements in

Dutch shallow inland waters. This was done, because the previous project (AeroData, 2015) didn't manage to capture the water points in small shallow ditches, due to the shallowness and turbidity of the waters. The study performed measurements with green and yellow wavelengths in the laboratory and compared the results. In addition, it aimed to understand how the properties of the Dutch waters can affect the laser pulse transmission and to define the optimal time window for ALB measurements.

The results were promising for the very shallow inland waters (i.e. under 30cm depth). Bottom returns in dark sands and sludge were captured, so as measurements in sediments can be achieved. Also, the results demonstrated that the sludge presents higher scattering values than any other bottom soil (e.g. sand), while the absorption levels are low and same in both wavelengths. Also, the Secchi depth measurements can provide accurate measurements regarding the water turbidity. The best time period for bathymetric measurements in the inland waters is between February and March, since parameters such as water vegetation and algae took into consideration.

In addition, the comparison between yellow and green measurements showed that green wavelengths failed to capture deep waters with high turbidity, whereas the yellow presented less failures. However, the role of bottom sediments is more important than the differences between the wavelengths. Moreover, the low energy per pulse makes difficult to get airborne measurements in deeper and turbid waters. The Single-Photon Counting (time-of-flight approach) presents better measurements (more points) in turbid waters and the Supercontinuum lasers gave a potential for more accurate measurements, since they have a multi-wavelength setup (i.e. green with NIR wavelengths in one laser). The laboratory data were validated with field (water) measurements.

Even if few pilot projects have be done to collect ALB data, the automatic and accurate classification of the water points into water surface, underwater and bottoms points is still a remaining issue. The effectiveness of the various combinations of wavelengths (e.g. green, NIR, both of them) have been studied in case of the Dutch water bodies and the different environmental factors have been analysed too.

Moreover, a following study used these bathymetric data in order to assess the water depth and detect the bottom points of the waterbeds (van Tol, 2019) and conducted in cooperation with Deltares. The proposed methodology was done in steps: first the noisy points were filtered out using a height threshold, then the remaining points were classified into ground and low points. In order to distinguish the water surface and sediments points, the echoes and the intensity values were used. All this process were done using *HydroVish* and *Terra Scan Solid* software. The results were not so satisfying as the classification process was based on cross sections analysis without taking into account the whole group of 3D points, and further research needs to be done.

3.3 PULSE-BASED VS. NEIGHBOURHOOD-BASED METHODS

There are many point cloud algorithms that aim to compute features or detect interesting points (e.g. lamp posts or other objects). These techniques use either the points of a laser pulse or the points within a local neighbourhood (e.g. 3D Voxel).

Grouping per pulses approach has a vital role in the LIDAR data, as it can used to detect and extract features. Mahphood and Arefi (2019) proposed a method to detect structures (e.g. buildings) from point cloud, depending on a method called Virtual First and Last Pulse (VFLP). This method concentrates on the height difference between the virtual first pulse (VFP) and virtual last pulse (VLP). In this method, the main assumption is the zero scan angle which corresponds in a vertical laser beam direction. By calculating this height difference, the vertical features (e.g. building walls) can be extracted. This method is effective to detect structures such as buildings and eliminate other features such as trees. Another study used the return numbers and number of returns from point cloud's characteristics just for ground filtering purposes. When the LiDAR dataset had multiple returns, the last returns may be used for filtering the ground level (Meng et al., 2010).

Neighbourhood-based methods aim to extract information from a local neighbourhood of points. For instance, the division of point cloud into a regular grid cells (voxel) helps to group a set of points in order to further analyse their spatial distribution. Grid representations and voxels can be used to speed up point cloud classification (Plaza-Leiva et al., 2017). Other studies such as (Boerner et al., 2017), proposed the voxel structure for ground segmentation; able to cluster the point cloud into ground and non-ground points.

Also, green LIDAR is used to evaluate the extent, the density and the height of aquatic reed beds at a lake (Corti Meneses et al., 2017). In order to detect the bottom of the reed, a classification process was applied to the point cloud. First, the water points have shifted due to refraction effect and therefore had to be corrected. Then, the points were selected based on an infinite vertical cylinder. For those points, statistics were calculated for identifying patterns based on the rank and point density parameters. The rank was the relative height position within this vertical neighbourhood, while the point density is the amount of points per square meter in the cylinder (Corti Meneses et al., 2017).

(Boerner et al., 2017) proposed a method to cluster the point cloud data into ground and non-ground points using voxel structure. First, the ground voxels are identified with a region growing algorithm, and then the non-ground voxels are searched in order to filter out remaining ground segments. In particular, the algorithm uses octree data structure to create the voxel structure and divide the 3D space into voxels. It searches the lowest ground voxels in a local neighbourhood and get its point with the lowest z value. In order to assure that the ground voxels are properly selected, the attributes of points fall into a voxel are used. When the return number and the number of returns are equal, then the point is the last one in a pulse. Therefore, the voxels which do not contain mostly last points of pulses are marked as non-ground voxels. Also, it uses local gradient between neighbouring voxels to identify the ground voxels (e.g. in a flat ground the gradient is zero) using a threshold value.

3.4 summary & conclusions

Many studies recognize that classifying the water points of a bathymetric LIDAR dataset is a complex and hard-demanding procedure. Some attempts are made and discussed why they can not be applied in the case of this research.

A few studies claimed to successfully classify the point cloud and then detect the bottom points of water bodies just using green LIDAR [Allouis et al. (2015) ; Mandlburger et al. (2015); Andersen et al. (2017); Kinzel et al. (2013)]. However, not every author followed the same approach to distinguish the surface and bottom returns. Allouis et al. (2015) used the green LIDAR waveform and applied a simulation approach (Monte Carlo). Others based on the full green waveform and spatial features in a local neighbourhood by using external thresholds Mandlburger et al. (2015). Most potential is shown in methods of Andersen et al. (2017) and Kinzel et al. (2013). Both authors applied refraction correction to the water points. The first author detected the water surface by fitting a line, and then detected the bottom surface used the remaining lower points. The other one detected the peaks from the green waveforms, the 1st corresponds to the water surface and the last to the bottom using threshold values.

Furthermore, few others combined the green LIDAR data with additional wavelengths such as NIR (1064nm) or Red (647nm) [(Allouis et al., 2010); Zhao et al. (2017)]. Both approaches used NIR data to enhance the detection of water surface, and then only use the green ones to detect the waterbeds. In this study, the provided data are only green bathymetric LIDAR and therefore the methods of these studies can not be treated here.

In addition to these studies, pilot projects with further studies have been carried out for the case of shallow inland water bodies in the Netherlands. Two different projects; one with field airborne bathymetric measurements (AeroData, 2015) and one with laboratory tests (Vazquez, 2017a) were done to identify the suitable combination of LIDAR wavelengths for the shallow and muddy Dutch water areas. Particularly, those wavelengths that can accurately capture water points by overcoming the existing environmental water conditions (e.g. sludge, algae, turbidity). Also, van Tol (2019) did an first analysis of the point cloud and tried to classify it without the desired results.

The processing and analysis of point cloud data is a demanding procedure. Many studies used methods that extracted valuable information from the laser pulses (e.g. first and last pulse returns) (Mahphood and Arefi, 2019), while other author used them for ground filtering purposes (Meng et al., 2010). However, not any study was found that used the pulse based approach in green LIDAR to classify the water points according to their characteristics (i.e. RN, NR). This study will explore the potential of exploiting those characteristics for the classification procedure.

Different studies used different approaches for constructing a neighbourhood of points, such as (Plaza-Leiva et al., 2017), (Corti Meneses et al., 2017) and (Boerner et al., 2017). Plaza-Leiva et al. (2017) divided the 3D space into voxels and grouped the points per voxel to further analyse them, while Corti Meneses et al. (2017) used the vertical cylinder shape to group them and then classify according the height and point density. Boerner et al. (2017) used octree to create the voxel structure in order to distinguish the ground and non-ground voxels. From the ground voxels extracted the lowest point corresponding to the ground. All in all, voxelization shows great potential to reduce the computational cost and to enhance the classification procedure. Consequently, the voxelization seems to be an interesting method and will be implemented in this study.

Study	Input LiDAR data	Waveform	Approach
(Allouis et al., 2015)	green	Full	GLFW model
(Mandlburger et al., 2015)	topobathymetric	Full	Classification water surface and waterbeds returns
(Andersen et al., 2017)	topobathymetric	Full	2D grid cells; highest point water surface, lowest point waterbed
(Kinzel et al., 2013)	green	-	Maximum peak (MP) and Last Peak (LP) algorithms
(Allouis et al., 2010)	NIR, green, Raman	Full	Combine NIR and green to detect the water surface and bottom parts
-	-	-	Raman waveforms using PCA to estimate the depth
(Zhao et al., 2017)	green	Discrete	Determine water surface and waterbed ,use the NWSP model for validation
(Vazquez, 2017a)	yellow	-	Laboratory tests between green and yellow wavelengths
(van Tol, 2019)	green	Discrete	Classification of point cloud using HydroVish and Terra Scan

Table 3.1: Summary of water classification cases using various ALB data (1/2)

Tables 3.1, 3.2 summarize the general findings of the various water classification cases using various wavelengths. Comparing those studies with my study case few differences are spotted. Regarding the input data, the waveform form, the main approach, the parameters, the comments-assumptions and the existence or not of ground truth data. Some of them used only green LIDAR data, others combined them either with near-infrared (topo) or Raman measurements, or tried another wavelength. Moreover, most of them had full waveform LIDAR and only a few had discrete LIDAR, as in my study. Different approaches were implemented in order to classify the water areas. Some created models based on specific parameters (see Parameters in Table 3.2) and others based the analysis either on pulse's characteristics or lowest/highest point of a 2D grid cell. Also, the majority of the studies used ground truth data to validate the results.

Study	Parameters	Comments - Assumptions	GT Data
(Allouis et al., 2015)	LiDAR system (power, time function,angle), river parameters (surface, bottom)	non-zero slope in longitude	No
	-	homogeneity of turbidity	No
	-	uniform rough water surface	No
(Mandlburger et al., 2015)	water depth, object reflectance, neighbourhood definition	using only green LiDAR	Yes
(Andersen et al., 2017)	water surface is horizontal	using only green LiDAR	Yes
(Kinzel et al., 2013)	threshold value, maximum number of points	-	Yes
(Allouis et al., 2010)	-	very shallow waters (<2m)	Yes
	-	very shallow waters (<2m)	Yes
(Zhao et al., 2017)	scanning angle, spot laser diameter, turbidity	shallow waters, hydrological data	Yes
(Vazquez, 2017a)	sludge, water turbidity	shallow waters	-
(van Tol, 2019)	-	shallow and muddy inland waters	Yes

Table 3.2: Summary of water classification cases using various ALB data (2/2)

In addition, the Table 3.3 presents all the pulse and neighbour-based methods of this chapters. Even if few studies were used to process different LIDAR data, their methodology was essential to define the current workflow (see Chapter 4) of this study.

Study	Method
(Mahphood and Arefi, 2019)	Pulse
(Meng et al., 2010)	Voxel
(Plaza-Leiva et al., 2017)	Voxel
(Corti Meneses et al., 2017)	Cylinder
(Boerner et al., 2017)	Voxel

Table 3.3: Summary of pulse and neighbour-based methods

To conclude, this study will try to establish and apply an automatic classification workflow that will be based on both the characteristics of the points and their spatial distribution in the 3D space. In my study, compared to the others, I will only use green discrete LIDAR. Also, ground truth data will also be used to validate the output.

4 METHODOLOGY

As described in Chapter 3, different approaches already exist to classify the water surface and bottom parts of water bodies using green LIDAR. Even while many studies tried to classify the water points using green LIDAR and sometimes with additional data, classifying and detecting water beds remains a challenging issue. In Chapter 3, various attempts with few promising results are described. A few studies, particularly, for shallow inland water areas in the Netherlands are shown and discussed. However, an improvement can be made in order to automatically classify those points using spatial algorithms.

Pre-processing steps are necessary to extract the appropriate extent of water bodies. The pulse and neighbourhood-based methods presented in this chapter aim to establish an automatic classification process for the water bodies. Using points characteristics, pulse grouping and voxelization to further analyse and classify the points.

This chapter begins with an overview of all the essential pre-processing procedures in Section 4.1, how the areas of shallow water canals are extracted (Section 4.1.1) and the further sorting criteria (Section 4.1.2). Following the pulse and voxel based methods presented in Section 4.2 and Section 4.3, respectively. The Section 4.3.3 describes how the confidence values for the potential bottom points are defined. Lastly, the Section 4.4 presents the ways to visualize the classified point cloud. Figure 4.1 gives a summary of the whole classification process of the green LIDAR.

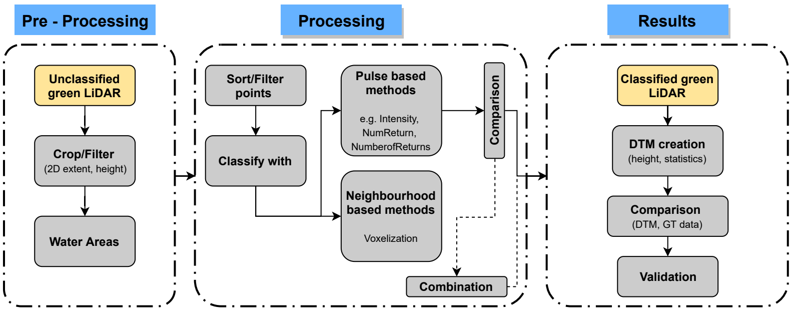


Figure 4.1: An overview of the workflow

4.1 OVERVIEW OF PRE-PROCESSING PROCEDURES

This study uses a raw and unclassified airborne topo-bathymetric LIDAR dataset, as presented in Chapter 1 and notably in Figure 1.2. In general topo-bathymetric LIDAR uses two wavelengths; the Near-infrared (NIR) (1064nm) and green (532nm) visible electromagnetic spectrum. In this thesis, only the green will be used since it can capture the water areas that we are interested in. Millions of points have been captured in both topography and water bodies. Also, some thousands of points are probably outliers (i.e. noisy points), which have been recorded due to external environmental conditions (e.g. air particles, reflection).

As a result, the pre-processing procedure (see Figure 4.1) is an essential first step to filter out outliers. Also many undesired points, such as urban structures (e.g.

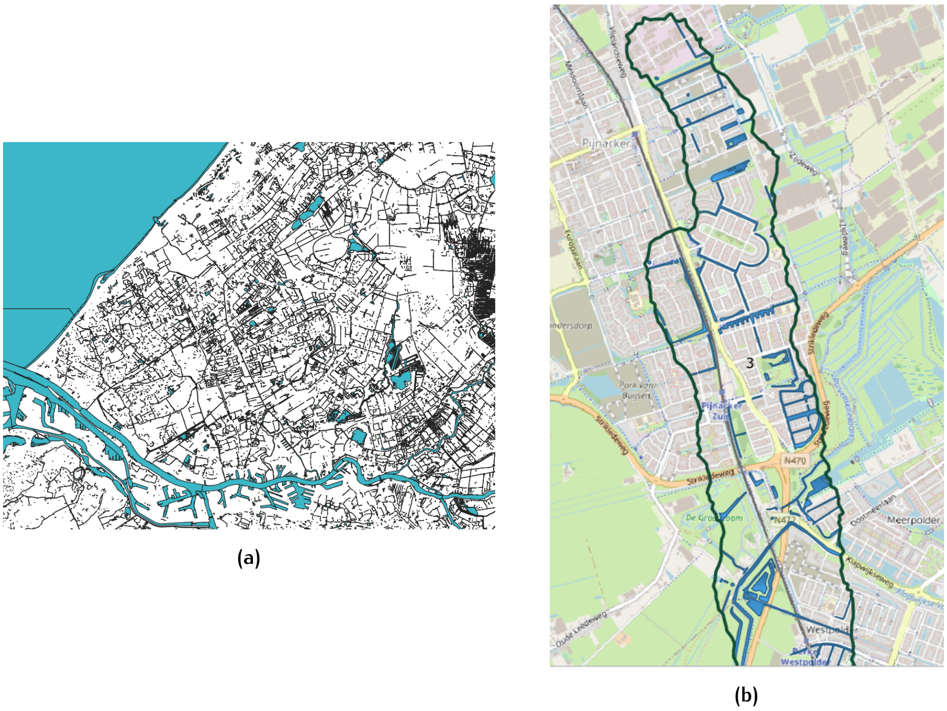
buildings, bridges) and vegetation, need to be removed from the given raw point cloud. Implemented tools can be used to remove the errors or duplicate points in this step (see Chapter 5). Thereafter, the filtering and cropping steps aim to extract only the water areas of interest out of the whole dataset.

4.1.1 Extract shallow water bodies

Airborne bathymetric LIDAR covers a wide range area and capture millions of points. These huge datasets need to be meticulously processed in order to obtain only the valuable information. As a result, all the unnecessary points are removed and the shallow water bodies can be extracted by applying a quick and effective technique.

The main idea is concentrated on the use of an external datasets with the exact shape of the water bodies. Since the area of the provided datasets cover big parts of the country, the use of an official digital base map; called *TOP10NL*, freely provided by Dutch Kadaster is chosen.

The *TOP10NL* is a digital topographic base map of the Kadaster. It is uniform, consistent and a map basis for the whole extent of the Netherlands. It consists of various topographic elements which were categorized in classes such as road sections, buildings, terrain. The most significant class is the *water part* as it contains all the registered water areas in the country. This means rivers, water canals, lakes, water polders and sometimes parts of the exclusive economic zone (EEC) in the sea (see Figure 4.2).



(a)

(b)

Figure 4.2: Top10NL dataset; 4.2a Registered water parts. 4.2b Inland water bodies in a small region.

Additionally, the extent of the various datasets (see Chapter 5) cover inland water parts. In most cases, the water areas are actual inland water bodies in the urban environment and not any river or sea parts as seen in Figure 4.2b. Moreover, this dataset was created by combining aerial photographs, field records and other external sources. This means that the shape and size of the water areas may differ from the reality due to low accuracy of used sources. Even though these differences are not so big (just few meters), the actual shape of the water bodies will differ and few undesired points will remain in the extracted water body point cloud. However, these few points can be filtered out in the next steps and will not affect the final classified output.

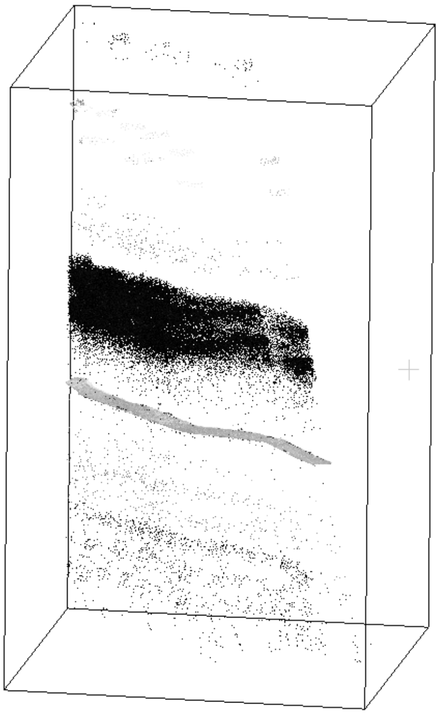


Figure 4.3: 3D view of a water body; existence of outliers in the higher and lower z level

Obtaining the 2D extent of a water body using the TOP10NL polygons, there are still many outliers and unnecessary points (e.g. buildings) in the z level (see Figure 4.3). In order to eliminate them, a threshold range in the z level is used and only the water points are extracted, as seen in Figure 4.4. This threshold range varies per dataset and is selected by visually inspecting it. Then, the up and low z values of the range is used to crop and store only the water areas of every dataset. More details are seen in Table 6.1.

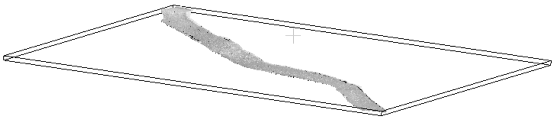


Figure 4.4: 3D view of a water body; cropped in the z extent

4.1.2 Sort criteria

As the shallow water bodies have been selected in the Section 4.1.1, the points can be processed based on their characteristics. An important factor for the LIDAR data is the quality checking (Isenburg, 2019). The main quality check is concentrated on the completeness and correctness of the discrete returns of each laser pulse. In particular, the existence of all returns in the file and the correct numbering of them, respectively (Isenburg, 2019). Thus, the completeness and correctness is an important issue that needs to be taken into account for every LIDAR dataset.

Every laser pulse usually collects from one up to five returns. Each of these returns contain an exact GPS time stamp that corresponds to the time the point-return was captured. By having these time stamps for all the points of the point cloud, the set of points per pulse can be recovered. Also, other problems in the set such as missing returns, duplicate returns or inconsistencies with the values of number of returns can be detected and removed. Existing tools (e.g. LAStools) are used in this research and restore the quality of the data.

Knowing the time stamp of each point, the point cloud can be sorted by GPS time. The group of points per pulse can be selected and further processed to distinguish which point possibly corresponds to the bottom surface of a water body. Note that the grouping procedure can be based also in the return number RN and number of returns NR, as explained in Table 2.2. However, even if the previous problems could be removed from the dataset, it is more secure to additionally use the GPS time stamp for the grouping procedure. Mainly because the used functions (e.g. lassort, lasreturn) try to restore the data without indicating how exactly they do it.

4.1.3 Refraction correction

Another issue is the refraction and slowdown effect of a laser pulse, when it is transmitted in the air interface and hits on the water surface (see Section 2.1.3). Both horizontal and vertical errors are introduced in the point data, resulting in points that are deeper and further away from the nadir than the real measurement (Parrish et al., 2019). The slowdown effect occurs because the speed of light in the water is getting smaller due to the transition from air to water interface. As a result, the distance between the underwater point and the water surface point (1st laser hit) is smaller. This is presented in the Figure 2.2a in the theory Chapter 2.

However, this refraction effect does not practically affect the classification procedure of this study; specifically the detection of the bottom points. Even if the geometrical position of the underwater points is a bit shifted, their relative spatial distribution may remain similar (see Figure 4.5). Thus, the refraction correction is not going to be applied in this study.

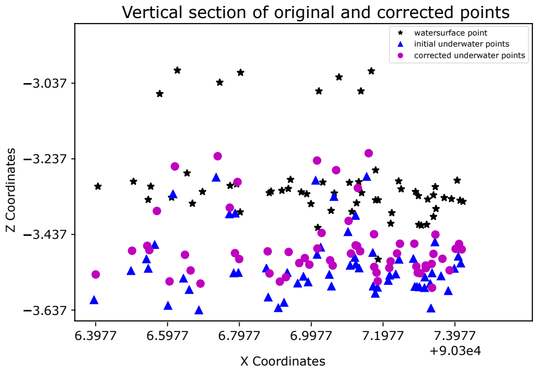


Figure 4.5: Vertical section of original and corrected water points due to refraction and slowdown effect

4.2 PULSE-BASED METHOD

Since the LIDAR points have been restored and sorted per GPS time, then the points of every pulse can be identified. Next, the waveform of a pulse approximation can be generated by using the discrete intensity values of its corresponding points, which is quite useful for the classification of the water bodies.

When the laser beam starts penetrating the water surface, the intensity value starts decreasing over distance due to multiple effects (e.g. due to the reflectivity of the water interface). The water surface point has a high intensity value as it is the first point that laser pulse hits on, while the following underwater points present gradually decreasing values. As the pulse goes deeper to the water column, it can possibly reach either the bottom surface or hit on any other objects or structure or it is absorbed completely. This reflection on a solid object, except the water particles, is translated to a small peak in the intensity graph. In other words, after the discretization of the waveform that reflection will correspond to the last point of that current laser pulse. The last point of the pulse can be interpreted in this study as a potential bottom point of the water body, since that small peak in the intensity graph has been presented (see Figure 4.6).

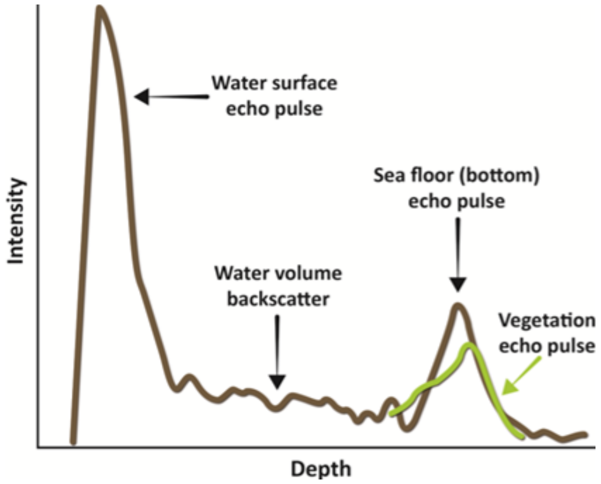


Figure 4.6: The three components of echoes from water area of green LiDAR consist of water surface return, water volume backscatter and bottom return (IQmulus, 2019)

It is important to mention that the laser beams of the given raw point cloud are not stored as a full waveform, but as discrete values that correspond to the stored points of every laser beam. For instance, the first intensity value (first point) of a pulse in the point cloud corresponds to the water surface echo pulse and the following values to the underwater points. The last value (last point) may indicate the sea floor (bottom) echo pulse, but it might be noise, too. Thus, the intensity value of a point varies per pulse as the environmental conditions (see Section 2.2.2) are not the same for the entire area of a point cloud. For example, seaweed and mud may be present in specific parts of the dataset.

The set of points per pulse needs to be selected based on certain criteria. As described also in previous Section 4.1.2, the GPS time stamp and the characteristics: RN and NR are known for each point. The recommended grouping procedure per pulse combines all these elements of points, especially, the GPS time, the RN and NR. In this study, the intensity value is not included in this grouping procedure since it may not have the expected format, as seen in Figure 4.6. In Section 4.2.2, the expected and undesired cases are discussed, while the pulses that reach the bottom can be distinguished. This is a challenging step as many errors (i.e. wrong NR of a point) may still remain in the characteristics of the points that will affect the grouping method. More detailed steps are described in the next Section 4.2.1.

4.2.1 Grouping per pulse

To identify the points of a pulse, an algorithm that iterates through the points is established. For every point, RN, NR and GPS time are retrieved. Since the points are sorted by their time stamp, time differences between following points are calculated. Then, the points can be categorized into three tags: 1. firstpoint 2. midpoint 3. lastpoint as seen in Algorithm 1.

In Algorithm 1, many different cases are considered in order to ensure the correctness of the grouped points. In particular, the common occasion is when all the points of a pulse exist in the dataset, they can be recognized and grouped. However, there are a few cases where points have been remained during the previous filtering procedure (see Section 4.1.1 and Section 4.1.2). For instance, the last point of a pulse remains, while the other points were out of the z threshold range and eliminated. Also, the case where a pulse captured just one point on the water surface. All these case studies taken into account were removed.

Algorithm 1: Grouping per laser pulse

```
Data: Point cloud
Result: List of points per laser pulse
// Initialize of a list with pulses
pulses ← [];
n ← 0 ;
for $p \in dataset$ do
    rn ← return number (p) ;
    nr ← number of returns (p) ;
    dt ← gps time(p) - gps time(previous p);
    if $rn == n$ and $nr > 1$ then
        if $rn == 1$ then
            pulse[p] ← $firstpoint$;
        else if $rn < nr$ and $dt < 1e\text{-}7$ then
            pulse[p] ← $midpoint$;
        else if $rn == nr$ and $dt < 1e\text{-}7$ then
            pulse[p] ← $lastpoint$;
        if $rn == nr$ then
            n ← 0 ;
        end
    // Case: one point!
    else if $nr == rn == 1$ then
        pulse[p] ← $firstpoint$;
    // Case: one point!
    else if $nr >= 1$ and $rn == 1$ then
        pulse[p] ← $firstpoint$;
    // Case: individual points of a pulse e.g. rn=3, nr=3
    else if $nr == rn$ and $nr != 1$ then
        pulse[p] ← $lastpoint$;
    n += 1
end
```

4.2.2 Identifying bottoms

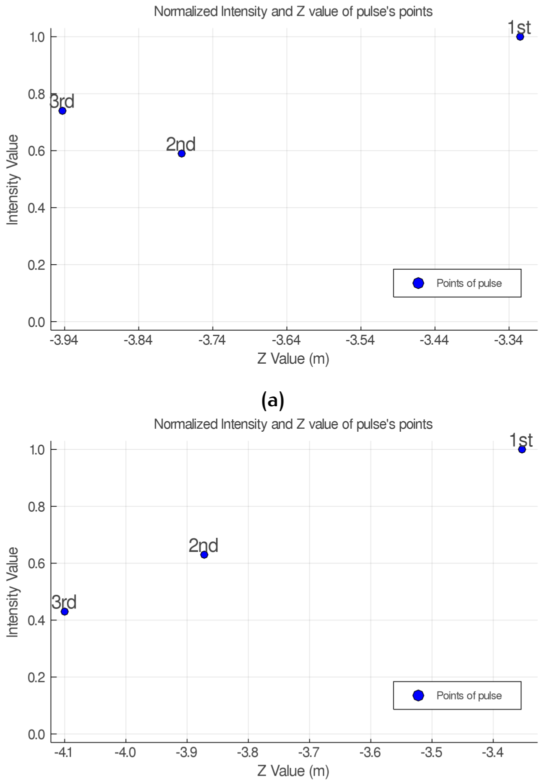
Since the points have been successfully grouped per pulse, the next step is to identify which pulses have reached the bottom of a water body. Due to the environmental factors, as described in Section 2.2.2, the laser pulse's transmission is affected. Also, the roughness and composition of the bottom have a crucial effect on the reflectivity of the pulse. For example, the bottom might have an extra layer of sludge on it. The laser pulse will reflect on this layer and not on the real bottom. In case the bottom surface is bumpy, the laser pulse might capture a point not on the real bottom due to its reflectivity.

Thus, a pulse may present either an expected or an undesired behaviour, as it penetrates the water surface of a water body and tries to reach the bottom part.

According to other studies, as presented in Chapter 3, the intensity of the pulse's waveform changes with respect to the depth. When the laser beam hits on the water surface, that time, the intensity value is quite high. As the beam pass through the water, the value decreases until to hit on an object or ideally on the bottom surface. Then, it shows a peak in the intensity value. This can be easily seen in Figures 4.6 and 3.4b, and presents the ideal expected pulse form.

The expected and undesired behaviour can be distinguished based on the intensity values of the discrete returns (points) of a pulse as they z value increases. For every pulse, the intensity values are normalized in order to be comparable between different pulses.

- **Expected**: the intensity of the 1st point (water surface) is high, the following points have lower intensities and the last one (bottom surface) a bit higher than the previous one (see Figure 4.7a)
- **Undesired**: the intensity of the 1st point (water surface) is high, following point present lower value while the last one (bottom surface) is lower than the previous one (see Figure 4.7b)



(a)

(b)

Figure 4.7: Normalized Intensity and Z value of points of a pulse; 4.7a Expected behaviour. 4.7b Undesired behaviour.

To select only the pulses with the expected behaviour, the intensity values of their points are used. Specifically, in the case of triple or more points in a pulse, the intensities of middle and last points are checked. When the ratio of intensity values between *last point/ middle point* is bigger than 1, then the last point has a peak and can be described as a potential bottom point. As for the double of points in pulses, the last point is considered directly as the bottom point.

4.3 NEIGHBOURHOOD-BASED METHOD

A neighbourhood-based method determines the position of an interesting point using some similarity measures between the point and its neighbourhood. As mentioned in Section 2.3, a search shape is used to select a neighbour of points. This shape may differ in many cases according to its application. For this study, the proposed shape is a 3D rectangular cube, called *voxel*. It offers an efficient and organized data model where the space division into voxels is a straight-forward procedure. Also, finding neighbouring cells and analysing the spatial relations of points is more easy. This voxel-based approach is computationally efficient in comparison to algorithms like k-Nearest Neighbors (kNN). The kNN identifies the k-nearest neighbours of a point of interest based on a query (e.g. distance). Its main drawback is its sensitivity to the scale of data as it may become slow for processing huge bathymetric data compared to the voxel-based method.

4.3.1 Voxelization

The first step is the creation of a 2D regular grid. The process to create this structure is based on the space partitioning of the point cloud. The 3D bounding box of the dataset is defined and all the points' coordinates are stored, but their topological relations are not defined. In a 3D grid, voxel locations are defined indirectly by their location in the grid. However, a grid may contain voxels with and without data points. In this process, all the empty voxels will not be stored to save memory space and maximize the processing time.

Another important factor is the *spatial resolution* or *voxel resolution*, i.e. the size of the voxel (see also Section 2.3.1). This resolution determines also the amount of voxels contained in the 2D grid. If the resolution is lower, the number of voxels is smaller and more voxels contain points. On the other case, the opposite happens.

The choice of the proper voxel resolution depends on three parameters:

- **Density** the amount of points in the dataset. When the point density is low and voxel resolution too high (lower voxel size), many empty voxel exist and gaps between them may be present. On the contrary, using bigger voxel size, more points will fall into the same voxel and lower level of detail will exist.
- **Area** the extent of area of the points in the 3 planes (i.e. A_{xy}, A_{yz}, A_{xz}) in correspondence with a certain number of points.
- **Processing time** should be kept to reasonable level. The processing time increases with the larger size and extent of a point cloud.

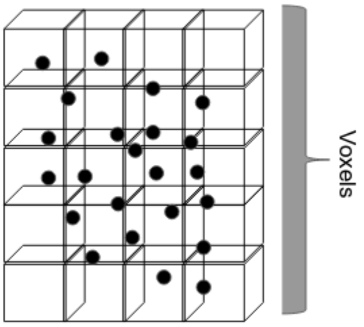


Figure 4.8: 3D Voxel; with (i,j,k) coordinates

The voxelization algorithms can be used for geospatial applications using point datasets. Nourian et al. (2016) described a voxelization algorithm for point cloud data. The algorithm creates a bounding box larger or equal to the bounding box of the dataset. It checks how many voxels fit in each dimension. Then, it creates a 3D array with all the voxels (i,j,k) and stores the points that fall into every voxel. The number of rows and columns is determined by the voxel size and the minimum (lower left) and maximum (upper right) points in the dataset.

Inspired by this study, the voxelization approach is changed for this case. Instead of assigning (i,j,k) coordinates for every voxel, the space division is done into voxels (also called *water columns*) were the k parameter is every time the z extent of the bounding box. This is an important modification of the algorithm, as the voxel's points in the whole z extent are essential for the further classification process. The detailed steps are presented in the following Algorithm 2.

Algorithm 2: Voxelization algorithm

```
Data: Point cloud
Result: Points per voxel
voxel_size ← constant value ;
// Extend the bounding box by voxel in all sides
new_min_point ← min_point - voxel-size;
new_max_point ← max_point + voxel-size;
// Number of voxels in x,y dimensions
nx ← (new_max_point (x) - new_min_point (x)) / voxe-size(x);
ny ← (new_max_point (y) - new_min_point (y)) / voxe-size(y);
// Initialize a 2D array with voxels
voxels ← [nx,ny];
// Compute voxel (i,j) for each point
dictionary [point] ← (i,j);
for every point ∈ dataset do
    i ← (point(x) - new_min_point (x))/ voxe-size(x);
    j ← (point(y) - new_min_point (y))/ voxe-size(y);
    dictionary ← (i,j);
end
// all the points with the same i,j coordinates are contained in the same voxel
voxels ← (i,j);
```

See Algorithm 2, the algorithm starts with identifying the new lower left and upper right points that define the new extent of the bounding box. Then, the number of rows and columns of the 2D array are calculated. Every point in the dataset has a unique combination of (i,j) coordinates, which correspond to the voxel where it belongs to. All the points with the same (i,j) are contained in the same voxel. Thus, the points are grouped per voxel.

4.3.2 Histogram per Voxel

Since the point cloud contains sparse and dense parts, it is obvious that the distribution of points, in the z level, of a voxel varies compared to the neighbour voxels. This happens when the laser beam hits on the water surface and then passes through the water to reach the bottom. Its direction and intensity value inevitably change. Therefore, the amount of points that a pulse can capture during its transmission in the water varies. It is clear that more points will be measured in the water surface as they can belong to the first returns of a pulse. As the laser pulse goes deeper, less points will be captured and possibly the last point may be at the bottom surface. Also, other factors (see Section 2.2.2) can affect the LIDAR collection procedure, and therefore the amount of captured data. Specifically, the top and right side view of a voxel are presented in Figure 4.10. The points cover almost all the parts of the 2D extent of the voxel, while from the right side is shown that many points are accumulated near the water surface and just few in the lower part.

Thus, considering all the points that fall into a voxel, the next step is to analyse their spatial distribution in order to detect the point that may possibly correspond to the bottom surface (see Figure 4.9). The approach is to create a histogram that captures the point distribution with respect to the z axis and has a fixed bin size (e.g. 0.1m) per voxel. This is helpful to determine on which depth large concentrations of points are located in a voxel.

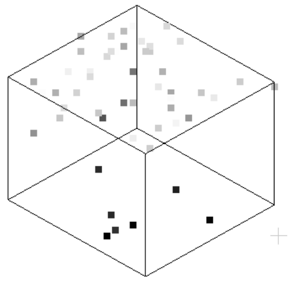
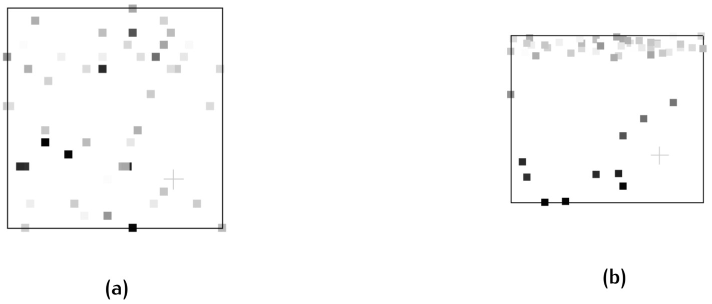


Figure 4.9: 3D Voxel with distributed points



(a)

(b)

Figure 4.10: 4.10a Top view. 4.10b Right side view.

For example, Figure 4.11 illustrates two cases of voxels. In both cases, there is a high peak near the water surface (i.e. bin with highest z level) where there are a lot of points compared to other regions in the voxel. However, they do not have the same distribution of points in the deeper parts of the voxels. In particular, the number of points gets smaller as the z decreases in the voxel (Fig.4.11), but not any second peak is presented in the figure. On the other hand, a second peak is shown at the deepest part of the voxel (Fig.4.11). This means that there is a quite big number of points captured by the laser pulses and could be part of the waterbed.

They may also not have the same number of bins, as in this Figure 4.11, where the Figure 4.11(a) has five bins, whereas the Figure 4.11(b). The number of bins is related to both the constant width size and to the fixed vertical extent of the voxel (i.e. z value of the bounding box). Thus, in case there are no points in the deepest part, then no bin appears in the graph. The same can happen in the middle z level of the voxel, and then no bin will exist (see Figure 4.11(b)).

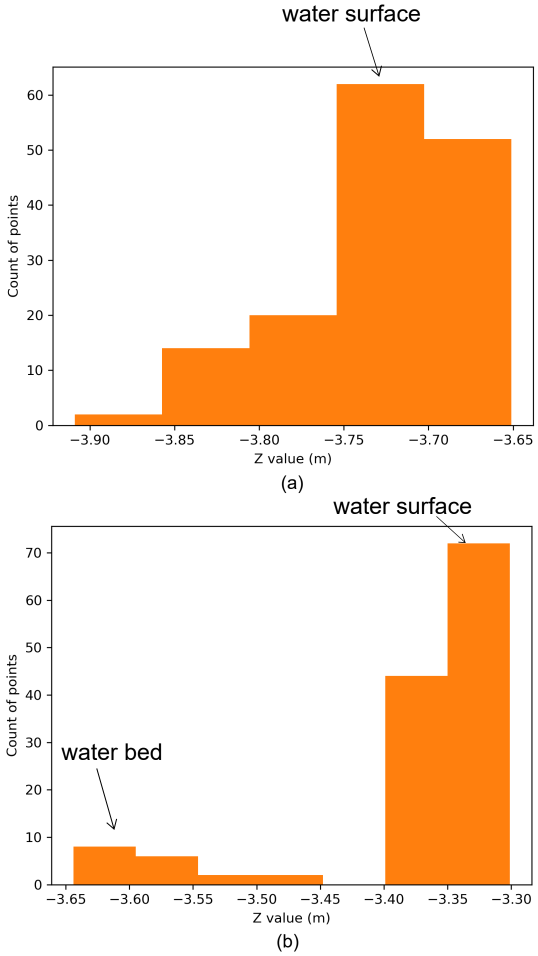


Figure 4.11: Histogram with bins (size=0.1m); Count of points with respect to Z value in a voxel; (a) Voxel with one peak (water surface) (b) Voxel with two peaks, water surface and water bed

This process is to be done for all the acquired voxels of a water body. For each voxel, a histogram is constructed. The highest peak and the 2nd high peak (if it exists) can be detected, but the location and height of the peaks may vary greatly per graph. For instance, all the bins have the same height or there are only two peaks with the same height. Therefore, it is complicated to determine how effectively the peaks may correspond to the water surface and the bottom part, respectively. Further parameters need to be taken into account and are presented in the following Section 4.3.3.

4.3.3 Definition of Confidence values

Provided that all the points of a voxel have been grouped into separate bins, the proposed algorithm aims to automatically extract the relevant points of the interesting bins. First, the algorithm needs to identify the peaks in the graph. A peak is defined when a bin is surrounded by lower bins or has one lower neighbour. For instance, the histogram (see Figure 4.12) presents three peaks: the highest (right bin with higher z) and two others; one in the middle and another at the end (left bin with lower z).

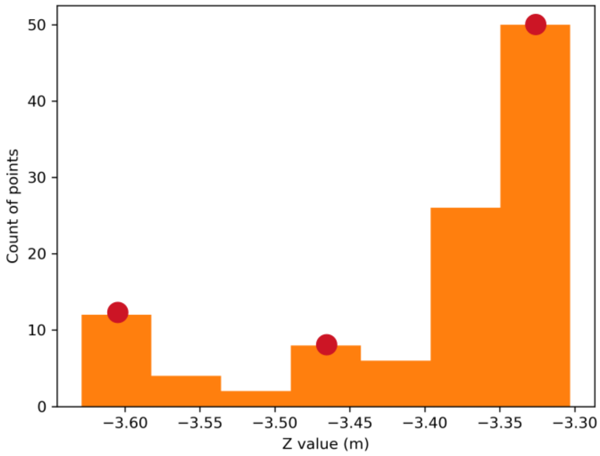


Figure 4.12: Peaks (red dots) of a histogram

Considering the peaks have been detected, the relevant points of the 1st and 2nd high peak can be extracted. However, three parameters are taken into consideration before the extraction of the points. The ***density***, the ***intensity*** and the ***distance*** are important to evaluate the situation of a histogram. In particular, the *density* is defined as a fraction between the amount of points in a bin and the total number of points in the corresponding voxel ($density = bin\_points/all\_points$). The *intensity* of point is related to the return strength of the laser pulse that generated the point. It is based on the reflectivity of the object hit by a laser pulse. In addition, the *distance* is specified as the absolute z difference between the two interesting peaks ($distance = 1rst\_peak\_z/2nd\_peak\_z$). In case of more than two peaks, the 2nd highest peaks is selected. Those three parameters will define the confidence level of a point to be a potential bottom point.

Algorithm 3: Create and process data of histogram

```
Data: A VoxelGrid V
Result: Interesting points of every voxel: Inter
for voxel ∈ V do
    // Extract points per voxel
    n ← number of points in voxel;
    points ← get all the points of a voxel;
    // Initialize histogram parameters
    bin_size ← constant value ;
    nbins ← (maximum(points) - minimum (points)) / bin_size;
    // Create histogram for every voxel
    counts, bins ← histogram (points, nbins);
    peaks ← findpeaks(counts);
    distance ← constant negative value;
    // Define the cases
    // Just one peak
    if length(peaks) == 1 then
        z_value ← peaks[1];
        density ← amount of points in the bin / n;
        if value <= mean(bins) then
            bottom_point ← lowest point in the bin;
            Inter ← [density, distance, intensity] ;
        else
            high_point ← highest point in the bin;
        end
    // Multiple peaks
    else if length(peaks) > 1 then
        first_peak ← peaks[1];
        second_peak ← peaks[2];
        z_value ← first_peak;
        density ← amount of points in the bin / n;
        if z_value <= mean(bins) then
            bottom_point ← lowest point in the bin;
            Inter ← [density, distance, intensity] ;
        else
            z_value ← second_peak;
            density ← amount of points in the bin / n;
            distance ← first_peak - second_peak;
            Inter ← [density, distance, intensity] ;
        end
end
```

As seen in Algorithm 3, there are multiple cases that the algorithm needs to take into account in order to process the data of histograms. For instance, voxels with only one peak, where the peak may be below or above the *mean_z* value of the voxel. For this case, if the peak is below and near to the left side of the histogram is assumed that it may correspond to the bottom area. The opposite happens when the peak is near to end of the graph, where the points of water surface are concentrated. The most common case is when multiple peaks exist. Then, if the 1st peak is above the *mean_z*, it presents the water surface, whereas the 2nd peak below the *mean_z* display the bottom surface.

The main steps of the Algorithm 3 are summarized below:

1. The algorithm iterates over every voxel, finds its peaks and the corresponding points.

2. For every voxel, a histogram is created with dynamic parameters (i.e. bin size, number of bins) based on the number of points in the voxel.
3. The peaks are identified.
4. If peak is one and its z values is below the *mean_z*, the lowest point of the bin is extracted.
5. For that point, *density* and *distance* values are calculated.
6. For that point, the *density*, *distance* and its *intensity* value are stored. (see Algorithm 4)
7. For that point, a confidence value is defined based on three elements: normalized density (NormDensity), normalized distance (NormDistance) and normalized intensity (NormIntensity) values. The results are stored in a dictionary with key: the voxel ID of the lowest point.

In case of more peaks, the lowest point of the 2nd highest peak (bin) is defined as bottom point.

As far as the confidence value, the Algorithm 4 illustrates the three thresholds: *density*, *distance* and *intensity* that taken into account. Thus, the main idea is to check three things: 1. the bulk of points in the interesting bin 2. the distance between the detected water surface and bottom part. 3. the intensity of the lowest point

In order to have a comparable confidence value for all the voxels in the point cloud, the three elements are normalized. In particular, the maximum values of density, distance and intensity of all the lowest points are extracted. Then, the density, distance and intensity values of every point divided with the corresponding maximum ones, and the results varied from 0 to 1.

In the next Chapter 5, many experiments have been performed to arrive at the most promising threshold values. Different classes that were used to define the level of confidence did not present promising results. The ranges of the threshold values were neither realistic nor dynamic for the real datasets.

Algorithm 4: Confidence value

```
Data: normalized Density: (NormDensity), normalized Distance:
        (NormDistance), normalized Intensity: (NormIntensity),
density threshold ($\tau_{den}$), distance thershold ($\tau_{dis}$), intensity thershold ($\tau_{inten}$)
Result: Confidence value: Conf
if $Den < \tau_{den}$ & $Dis < \tau_{dis}$ & $Inten < \tau_{inten}$ then
    Conf ← high confidence ;
else
    Conf ← low confidence ;
end
```

Therefore, a more dynamic and robust solution for the threshold values is proposed. The median and mean values of the *NormDensity*, *NormDistance* and *NormIntensity* are calculated in order to get insight about the range in the entire dataset.

It is important to mention what is the difference between these values. The *mean* is the average, where all the numbers are added and then divided by their total number. The *median* is the middle value of all the numbers. In this case, the median is selected to be used for the *NormDensity* and *NormIntensity*, and the mean for the *NormDistance*. The reason is that the median can be more realistic for these data, as extreme values may exist and can certainly affect the mean value. The median is not used for the *NormDistance*, because the points extracted from a histogram with one peak had zero distance. That's why they affect the median value and not the mean used.

The **threshold values** are defined as follows: (τ_{den}): median of NormDensities, (τ_{dis}): mean of NormDistances and (τ_{inten}): median of NormIntensities, as seen in

Table 4.1. Eight classes (1-8) are defined, where one (1) corresponds to the highest confidence and eight (8) to the lowest one. The threshold parameters are combined based on their significance in this procedure.

In particular, both density and intensity are bigger than their median values in the first two classes. Greater density means a bigger amount of points in the bin of the selected point, while greater intensity is interpreted as more chances of getting points at the bottom. The distance is bigger than its threshold value for the first class, as the bigger distance between the two peaks in the histogram means that second peak's point are deeper in the z axis of the voxel. Next, the following three classes (3-6) have distances bigger than the threshold since it is important as explained before. Then, the density parameter is chosen to play a more critical role than the intensity. That's why the density is higher than its threshold in the class (3), while the intensity in the class (4). In class (5), both are smaller than the critical value. Afterwards, the same order follows in the next three classes (5-8), where the distance remains below its critical value.

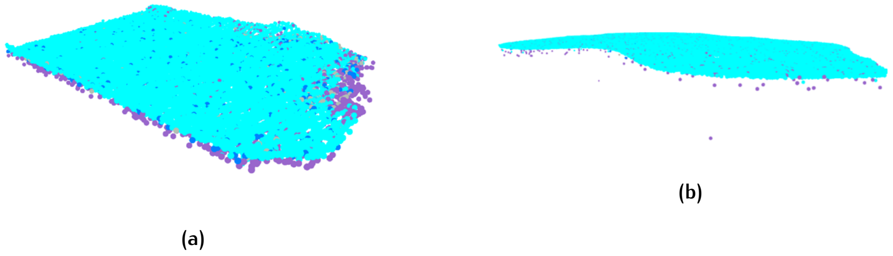
Density (τ_{den})	**Distance** (τ_{dis})	**Intensity** (τ_{inten})	**Confidence Value**
$> \tau_{den}$	$> \tau_{dis}$	$> \tau_{inten}$	1
$> \tau_{den}$	$<= \tau_{dis}$	$> \tau_{inten}$	2
$> \tau_{den}$	$> \tau_{dis}$	$<= \tau_{inten}$	3
$<= \tau_{den}$	$> \tau_{dis}$	$> \tau_{inten}$	4
$<= \tau_{den}$	$> \tau_{dis}$	$<= \tau_{inten}$	5
$> \tau_{den}$	$<= \tau_{dis}$	$<= \tau_{inten}$	6
$<= \tau_{den}$	$<= \tau_{dis}$	$> \tau_{inten}$	7
$<= \tau_{den}$	$<= \tau_{dis}$	$<= \tau_{inten}$	8

Table 4.1: Confidence values based on τ_{den}, τ_{dis} and τ_{inten} thresholds

4.4 VISUALIZATION OF A POINT CLOUD

Since the water surface and bottom points have been detected, the next step is to present this new information. It can be either integrated in the original point cloud as an extra attribute or exported in a new output product (e.g. raster/TIN).Both ideas can be implemented as they offer different capabilities.

In particular, the class information (e.g. water surface, underwater, bottom) of a point can be stored in the classification attribute. Then, using 3D viewers the points can be colourized and presented based on those unique classification codes. For example, the 4.13 presents two example water bodies that have been classified and colourized based on their classification codes. The water surface points are shown with light blue colour, the underwater with dark blue and the bottom points with purple.



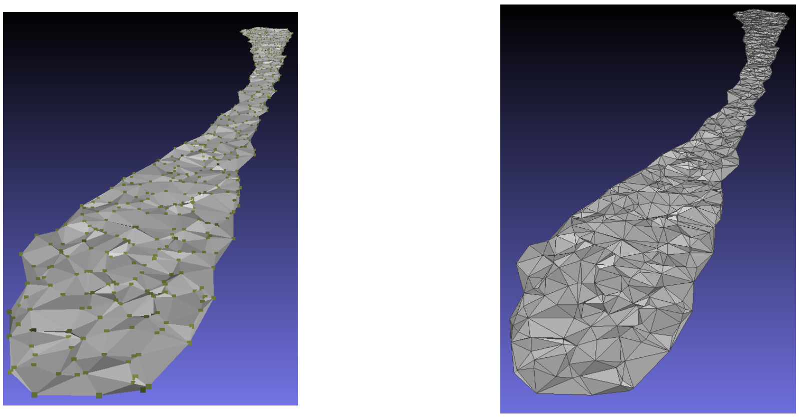
(a)

(b)

Figure 4.13: Classified water bodies; water surface (light blue), underwater (dark blue) and bottom (purple) points 4.13a . 4.13b .

Moreover, the classified point cloud can be exported into a 2D raster output as explained in Section 2.4.2. Specifically, the grid structure can contain filled or empty grid cells based on the existence of a detected bottom point in that region of the dataset. Also, the bottom points can be interpolated and then exported to a raster in order to have a uniform surface, if gaps exist due to no existence of data. Also, other characteristics such as the height, intensity and confidence value information can be visualized through a raster output. More outputs are presented and discussed in the Chapter 6.

Another approach is to create the 3D bottom surface (DTM) of a water body by triangulating the bottom points. For instance, the bottom points are the vertices of the TIN in order to reconstruct the 3D geometry of the waterbed (see Figure 4.14).



(a)

(b)

Figure 4.14: Triangulated bottom points of a water body 4.14b Points are vertices of the triangles. 4.14a Triangles of the bottom surface.

5 IMPLEMENTATION, EXPERIMENTS AND COMPARISON

This chapter explains the implementation of the pulse and voxel-based methods introduced in Chapter 4. This achieved by running number of experiments on various datasets. Firstly, the Section 5.1 describes the areas, the datasets, the tools and software that are used in the implementation. The Section 5.1.2 explains how the implementation was done and in what degree this procedure is automated.Then, Section 5.3 presents some experiments that were done to create and test the classification code and they were not promising. Section 5.4 describes the comparison and the potential combination of the two methods. Lastly, the Section 5.5 displays the quality validation of the results using ground truth data.

5.1 DATASETS & TOOLS

5.1.1 Areas and Datasets

In this thesis, several airborne bathymetric LIDAR datasets are used. The datasets are provided by Deltares and correspond to six different regions in the Netherlands. Several water boards had organized a pilot project in cooperation with Deltares [1], Stowa [2] and Waternet [3]. They tried to examine the potential of green LiDAR for the shallow and muddy Dutch water-bodies. These six areas are located in the western, middle and north-eastern part of the country, which all together represent a variety of environments. For instance, rivers, parts of sea, inland muddy water canals, deeper and shallower waters. In this thesis, inland shallow water bodies and river parts will be the test cases for the classification method. That project had to show which areas in the Netherlands can de detected with green LiDAR, if it worked or not and the future potential of these data. The acquired data from this project will be processed and used in order to run and test the implemented algorithms.

Moreover, the LIDAR sensor characteristics have a vital role for the quality of this study. The Riegl VQ-880-G laser system was used, as it is a fully integrated airborne laser scanning system for combined topobathymetric surveying (RIEGL, 2016). This system has integrated GNSS, IMU and cameras, while its scanner pattern is circular (see 5.1). Also, it has an effective measurement rate is up to 550.000 measurements per second and a scan angle of 40 degrees. It is widely efficient for applications such as coastline and shallow water mapping and surveying for hydraulic engineering (RIEGL, 2016).

[1] https://www.deltares.nl/en/
[2] https://www.stowa.nl/english
[3] https://www.waternet.nl

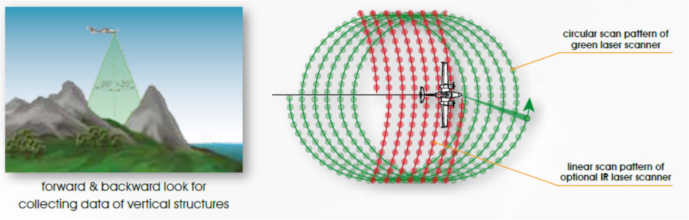

forward & backward look for collecting data of vertical structures
20° 20°
circular scan pattern of green laser scanner
linear scan pattern of optional IR laser scanner

Figure 5.1: Scan pattern of Riegl VQ-880-G; field of view ±20° and circular pattern for the green laser scanner (RIEGL, 2016)

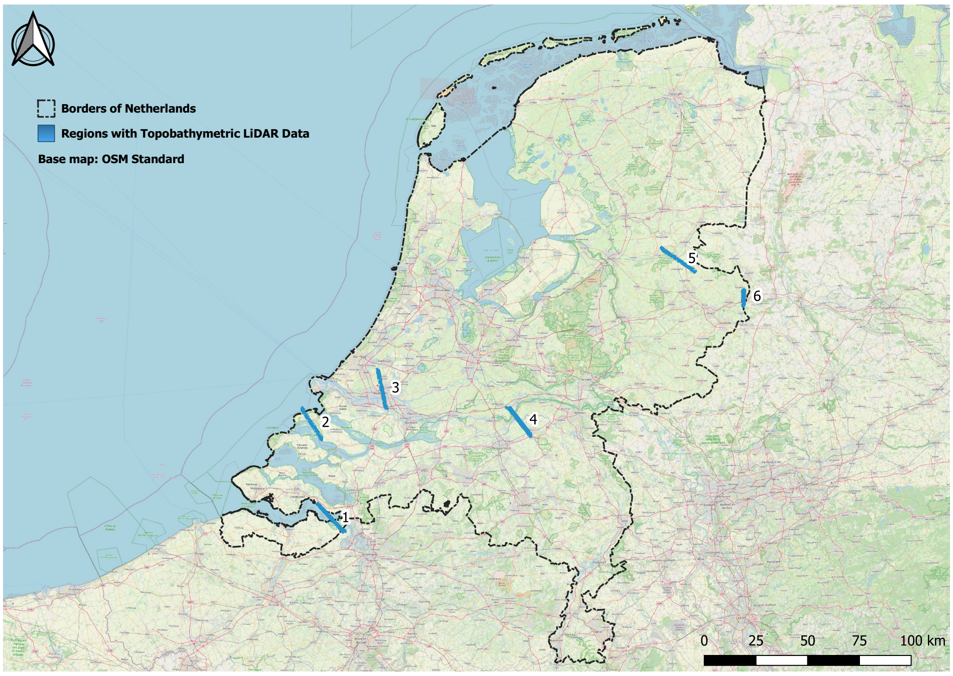
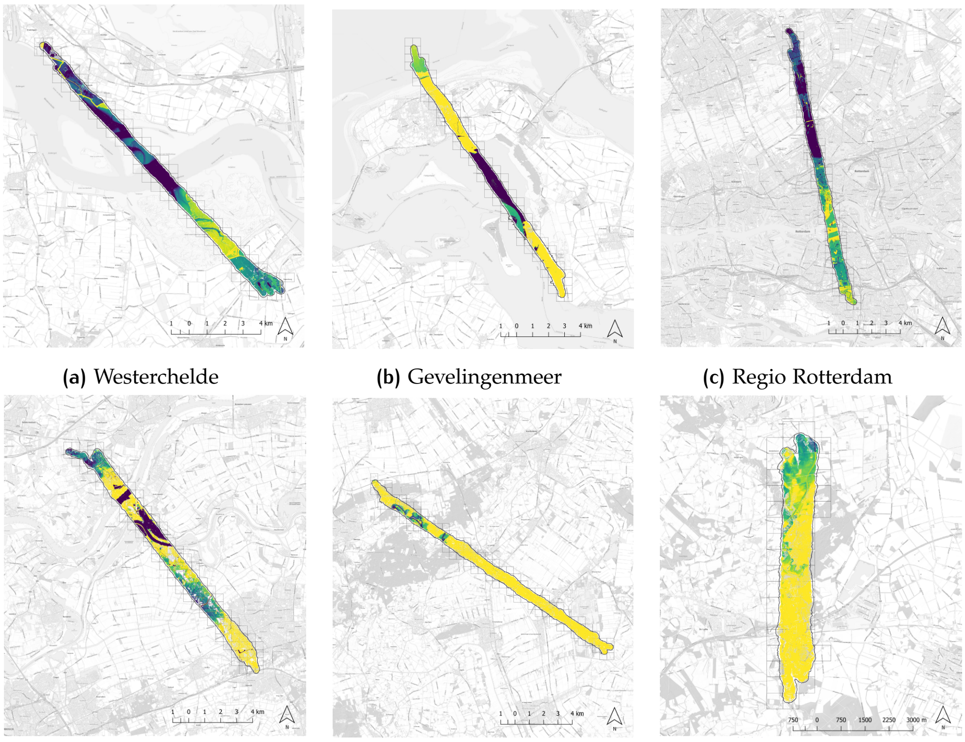


Figure 5.2: Location of the six regions in the Netherlands

The six different point cloud datasets are displayed on a background map, as shown in Figure 5.3. The various colours indicate the water depth in specific parts of each region. In particular, dark blue colour illustrates the water areas whereas yellow one the non-existence of water. The water bodies from all these regions were extracted into separate *.laz* files using Top10NL dataset (more details in Section 4.1.1). Then, they were used as experiments to run the implemented algorithms and eventually to classify them.



(a) Westerchelde

(b) Gevelingenmeer

(c) Regio Rotterdam

(d) Oss

(e) Regio Vechtgebied

(f) Regio Dinkelgebied

Figure 5.3: Datasets of six different regions in the Netherlands (van Tol, 2019)

Furthermore, the characteristics of these datasets have been calculated by (van Tol, 2019). The measured surface (per km^2), the average point density (points per m^2) and maximum soil depth (m) differ per location.

Location	Measured surface (km²)	Average Point Density (points per m²)	Maximum soil depth (m)	Number of points
Westerschelde	14,263	37,33	-5.04	1.136.543.997
Gevelingenmeer	12,304	58,89	-5,24	1.391.822.923
Regio Rotterdam	13,458	17,63	-2.46	1.721.510.390
Oss	18,263	15,70	-2,30	1.764.699.685
Vechtgebied	14,041	23,167	-1,08	1.631.270.918
Dinkelgebied	7,527	52,63	-1,12	1.105.047.875

Table 5.1: Characteristics of green LiDAR datasets from six different regions in the Netherlands (van Tol, 2019)

Moreover, the various water boards have conducted ground truth measurements in those regions by selecting depth and transparency data. Those measurements have been captured one day after the flight over that regions. These reference data are going to be used in the last steps of this study for validation of the results of the implemented algorithms. For instance, the field depth measurements were made in a few profile sections of a water body as seen nn Figure 5.4. Several water points were measured on that sections (see Figure 5.5). The z values of points are measured with respect to the nap vertical datum. Those data are going to be compared with the LIDAR point data in that areas in order to calculate the z differences.

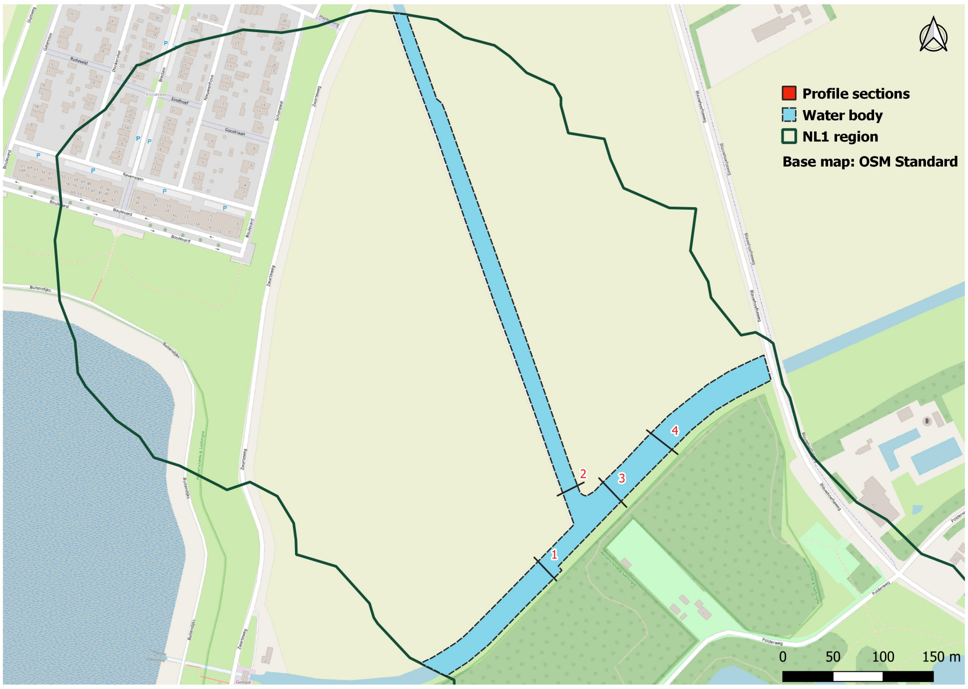
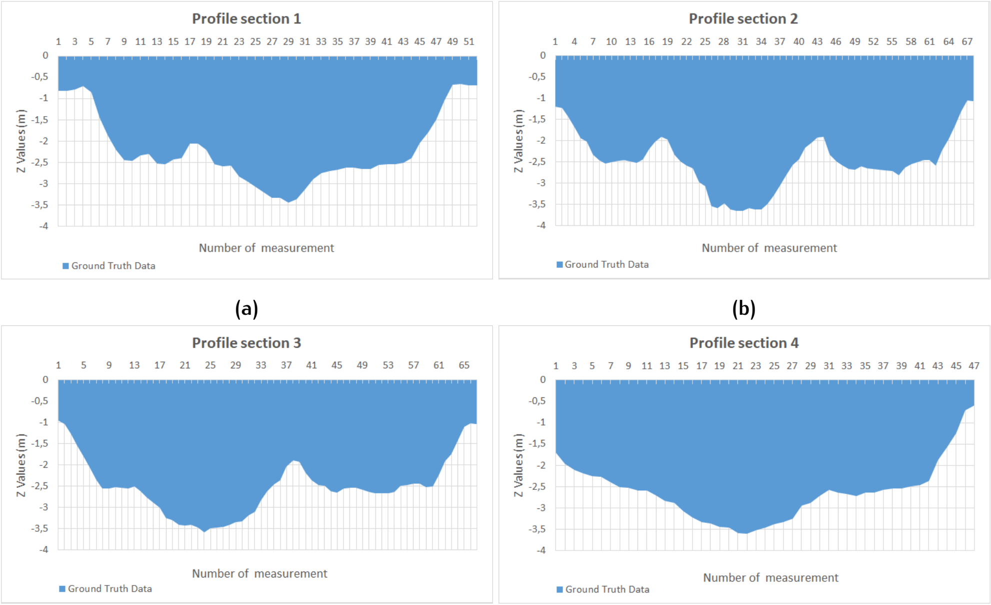


Figure 5.4: Ground truth measurements; Profile sections (1,2,3,4) in a water body

It is important to mention that the field measurements may contain errors in their quality as they can be affected by the water conditions. The possible presence of a layer of sludge above the ground is an important factor. Many points can be measured as bottom points because of this layer. Also, the turbidity of the water and the underwater vegetation (e.g. algae) implies difficulties to the process. As a result, even though field measurement are provided, it may be difficult to trust the quality assessment due to the previous factors.



(a)

(b)

(c)

(d)

Figure 5.5: Profile sections of water body as seen in Figure 5.4. The profile sections present the depth (z values) measurements of several bottom points with respect to the nap vertical datum

5.1.2 Tools and Software

A large number of tools were used in this study to implement the methodology described in Chapter 4. In this Section 5.1.2 listed the software were used to process and visualize the dataset. A number of algorithms are used in this thesis that were not implemented by me. Furthermore, many Julia and Python packages are used to structure the algorithms and process the data, which are presented here as well.

All the algorithms in this study are developed in a laptop with 64-bit Window10 operating system. The details of the laptop are:

Model	Processor	RAM	CPU
HP ZBook 15 G5	Intel(R) Core(TM) i7-8750H GPU @ 2.20 GHz	32 GB	Quadro P1000

Table 5.2: Hardware characteristics

Programs

In this thesis, I used the following programs for processing, converting and visualising my dataset.

- **QGIS or Quantum GIS** [4]: is an open source and free cross platform geographic information system (GIS) application that used to view, edit and analyse geospatial data. Also, there is an integrated toolbox with Geospatial Data Abstraction Library (GDAL), SAGA (saga) and GRASS (grass) geographic systems.
- **LAStools** [5]: is a software suite with a collection of tools for processing LIDAR data. It is also available as a LIDAR processing toolbox in QGIS.
- **CloudCompare** [6]: is an open source and processing software for 3D point cloud and meshes.
- **Displaz** [7]: is a cross platform viewer for displaying LIDAR. It is flexible for exploring large datasets, and easy to visualize them by creating custom point visualizations based on their attributes.
- **MeshLab** [8]: is an open source system for processing, editing and visualizing 3D meshes

Implemented Algorithms

Some implemented algorithms, that are not created by me, are used in this study to process the data.

- **lasindex**: creates a *.lax* file for a given point cloud file *.laz*, that contains spatial indexing information. The presence of this file in the directory of the *.laz* file speeds up the access to the point data whenever a spatial query is used.
- **lasclip**: clips an input point cloud file based on a shapefile file with one or more polygons. It can clip all the point that fall into the polygons and store points to separate point cloud files. It was used to crop the interesting water bodies using the Top10NL water boundaries.
- **lassort**: sorts the points of a point cloud file based on an attribute. In this study, it is used to sort them based on their GPS time.

[4] http://www.qgis.org.
[5] https://rapidlasso.com/lastools/
[6] https://www.danielgm.net/cc/
[7] https://github.com/c42f/displaz
[8] https://www.meshlab.net

- **lasreturn**: finds sets of points of a laser pulse with incomplete returns or fixes the NR based on the GPS time stamp.

Julia and Python packages

The Julia and Python languages are used in this thesis for data processing and analyses. A number of libraries have been used, which are presented here:

- **LasIO** [9]: Julia package for reading and writing LAS lidar format
- **LazIO** [10]: extends LasIO with Laszip integration
- **FileIO** [11]: Main Package for IO, loading all different kind of files
- **StaticArrays** [12]: provides package for statically sized arrays for Julia
- **Dataframes** [13]: run with tabular data
- **Plots** [14]: for data visualization purposes
- **ArchGDAL** [15]: A high level API for GDAL, to rasterize the lidar data
- **GeoArrays** [16]: Simple geographical raster interaction
- **Numpy** [17]: stores the data into arrays and matrices and provides a collection with mathematical functions
- **Rasterio** [18]: reads and writes gridded raster datasets
- **Startin** [19]: a Delaunay triangulator for processing TINs

5.2 IMPLEMENTATION OF ALGORITHMS

To test the proposed method as described in Chapter 4, an algorithm with many sub-algorithms is implemented. This developed algorithm should run for every ALB dataset, process the data and return the classified output. This is possible with Julia programming language (used by Deltares), as many packages for point cloud processing already exist. All the packages are freely available, where few of them are in this Deltares GitHub repository. As far as my implemented algorithms can be found in this repository: Green LiDAR —Automatic Detection of Waterbeds.

5.2.1 Implementation of algorithms

This section presents the main parts of the implemented algorithm based on the proposed classification model. The first step involves the pre-processing proceedings (see Section 4.1.1, Section 4.1.2). Then, the two separate classification methods are developed: the pulse and neighbour (voxel) based.

The pulse-based method is structured and grouped points per pulse as introduced in Section 4.2.1. Intensity vs Z values plots were created for every pulse and they categorized into expected and unexpected (see **??**). These plots visualized the

[9] https://github.com/visr/LasIO.jl
[10] https://github.com/evetion/LazIO.jl
[11] https://github.com/JuliaIO/FileIO.jl
[12] https://github.com/JuliaArrays/StaticArrays.jl
[13] http://juliadata.github.io/DataFrames.jl/stable/
[14] http://docs.juliaplots.org/latest/
[15] https://github.com/yeesian/ArchGDAL.jl
[16] https://github.com/evetion/GeoArrays.jl
[17] https://numpy.org
[18] https://rasterio.readthedocs.io/en/latest/
[19] https://github.com/hugoledoux/startin

differences into the behaviour of the laser pulses. This process is shown in Figure 5.6, but it wasn't used into the final step of the classification procedure. The voxel-based method involves creating the voxel grid data structure. All the points that fall into a voxel are extracted and processed separately. A histogram is created per voxel with a fixed bin size. In particular, the bins that correspond to the water and bottom surface are selected, respectively. The lowest point of all the points in the bin (2nd peak) is classified as bottom point. Whereas the highest point of the points in the bin (1st peak) is categorized as the water surface.

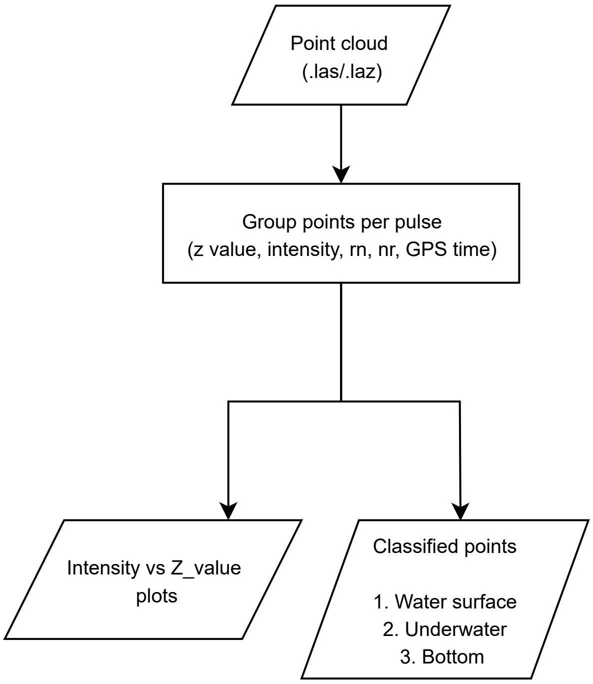


Figure 5.6: Pulse-based method

The neighbourhood-based method; voxelization structure is initially created from a *.las* or *.laz* file as described in Section 4.3.1. The process creates a data structure which describes the data points that are contained in each voxel (see Algorithm 2). Then, a histogram is created per voxel which depicts the amount of points per bin with respect to the z values (see Algorithm 3). The points of the bin which presents the 2nd highest peak in the histogram were extracted. The lowest one from them was assumed as potential bottom point. Thus, a confidence value was assigned to estimate how likely or not is to be a bottom point (see Algorithm 4). As mentioned in Section 4.3.3, the confidence value was based on three parameters: density, distance and intensity. More details about further experiments can be seen in Section 5.3.

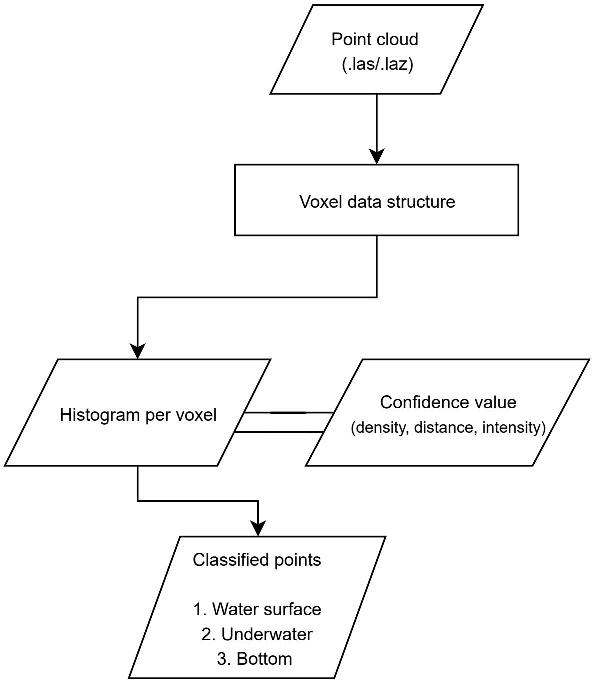


Figure 5.7: Voxel-based method

5.3 EXPERIMENTS WITH NOT PROMISING RESULTS

This section describes the performance of few implementations that failed to give any valuable information during the classification procedure.

5.3.1 Pulse-based classification

During the pulse-based method, the intensity value of points was tested to be used and to enhance the classification; especially the grouping per pulse procedure. Graphs with intensity value of a point with respect its z information were created. As described in Section 4.2.2, the reconstructed waveforms of pulses usually didn't display the expected curve; where the intensity of the last return is higher than that of the previous point (see Figure 4.7b).

In order to research the behaviour of pulses, pulses with three returns were plotted in the same Figure 5.8 and 5.9. The waveforms of pulses do not follow any similar pattern. In particular, there are cases where the intensity value of last return (blue point) of a pulse is higher or lower than that of its previous (red point). Only in cases where the intensity of the last point presents a peak compared to the previous one can it be considered as reliable element of a point. Thus, all points that pass this test are identified and considered as potential bottom points during the grouping per pulse procedure.

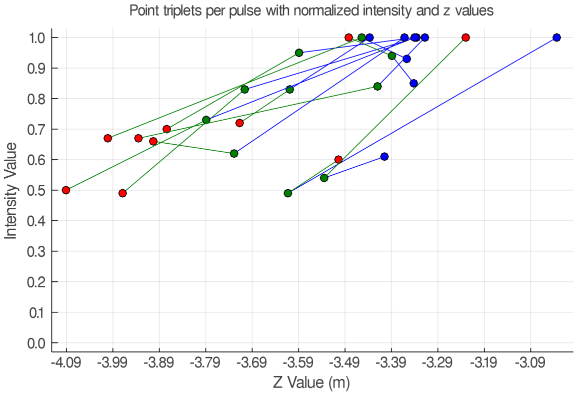


Figure 5.8: 10 pulses; point triplets per pulse with normalized intensity and z values. First return (blue point), middle return (green point) and last return (red point)

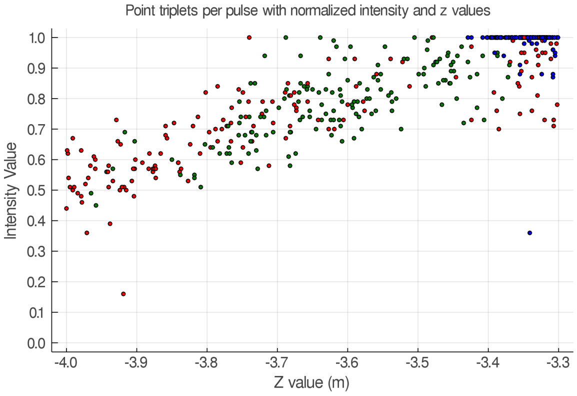


Figure 5.9: More than 100 pulses; per pulse with normalized intensity and z values. First return (blue point), middle return (green point) and last return (red point)

Figure 5.10 illustrates the correspondence of intensity and return number value for set of points in a pulse. It is evident also that the intensity of the last return (red point) varies per pulse. The waveforms of pulses do not follow any similar pattern using only the return number with respect to the intensity factor.

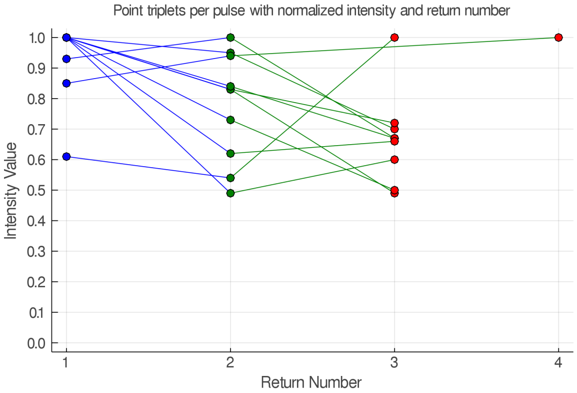


Figure 5.10: 4 pulses; point triplets per pulse with normalized intensity and return number. First return (blue point), middle return (green point) and last return (red point)

Following the implementation of these experiments, both the intensity and return number were considered to be important factors in the pulse-based approach.

5.3.2 Threshold values in voxel-based method

Table 5.3 displays the eight different classes (from 1 to 8), which are defined based on specific combinations of different thresholds values. The range of density (0 to 1) split into four equal groups. The same grouping was done for the range of the intensity. As far as the distance is concerned, it was separated into two groups; below and above 10cm. Value of a point is 1, then it is very certain to be a bottom point. On the other hand, when the point has value 8, then it means that it is not enough at all to belong to the bottom surface. The middle numbers ranging from 2 to 7 indicate the increasing confidence in these points.

However, these ranges were neither realistic nor dynamic for the real datasets. Figure 5.11 illustrates the classified point cloud, where the points with confidence value (6) are coloured purple. Few points are red (8) and light blue (5) in the upper part of the dataset, whereas all the blue points have not been classified. Thus, the classification of the bottom points failed. This happened since the range of thresholds' values do not have any practical meaning for the point cloud and needs to be dynamic.

Density (τ_{den})	Distance (τ_{dis})	Intensity (τ_{inten})	Confidence Value
>0.75 - <=1.0	>0.10	>0.75 - <=1.0	1
>0.5 - <=0.75	>0.10	>0.5 - <=0.75	2
>0.25 - <=0.50	>0.10	>0.25 - <=0.50	3
<= 0.25	>0.10	<= 0.25	4
>0.75 - <=1.0	<=0.10	>0.75 - <=1.0	5
>0.5 - <=0.75	<=0.10	>0.5 - <=0.75	6
>0.25 - <=0.50	<=0.10	>0.25 - <=0.50	7
<= 0.25	<=0.10	<= 0.25	8

Table 5.3: Confidence values based on hard coded thresholds

If there is only one peak in the histogram and this value is below the mean_Z of the graph, then the distance value for the extracted point is setted to -1 (see Algorithm 3). The *NormDistance* is certainly negative number and is included in one of the classes 4 to 8. This output is not satisfying, and a more dynamic and robust solution is implemented as described in the methodology (see Section 4.3).

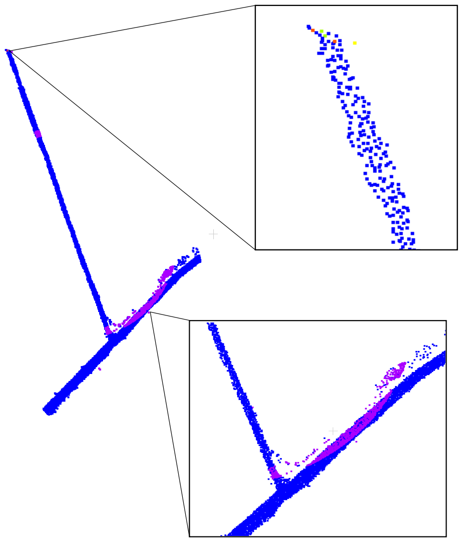


Figure 5.11: Water body (NL1_51): Classified with confidence values. Only points in class (8) and few in (5) and (6)

5.4 COMPARING AND COMBINING PULSE GROUPING AND VOXELIZATION

Both pulse and voxel-based methods have been separately developed in order to enhance the classification procedure of the point cloud. The pulse-based method aims to reconstruct the waveforms of the pulses using the discrete LIDAR data, while the voxel-based method divides the 3D space into a 3D regular grid with voxels.

However, the pulse method is only based on the points' attributes (e.g. RN,NR) in order to group them per pulse in contrast with the voxel method, which examines the spatial distribution of the points in a voxel. Thus, merging the point's information with the geometrical data of a voxel may have the potential to increase the effectiveness; in terms of the accuracy of the classification method.

Experiments with many datasets showed that it is extremely difficult to reconstruct the waveform of pulses containing in voxel. Due to the voxel size and the pre-processing procedures (see 4.1.2), many points have been eliminated. In addition, the direction of a pulse alters, when it hits on the water surface and surely does not follow a vertical route with respect to the water surface. Therefore, it is evident that few returns of the pulses; specifically the last ones, may be cropped out of the voxel due to its geometry. Thus, it is complex to define the right voxel size which will not exclude significant points of a pulse.

A procedure that combines both pulse and voxel-based methods is practically not sufficient. As the pulses that hit on the water surface contained in a voxel, many of their last returns may fall out the voxel. Therefore, only a comparison of the separate classified results can be visually done.

5.5 VALIDATION OF DATA

The simplest way to assess the results of classification procedure is by visual inspection. By colourizing the three different classes: the water surface, underwater and bottom points; the water surface and bottom parts can be roughly shown. In addition comparing the classified dataset and especially the bottom points with ground truth data is more effective.

Since the ground truth data are measured in few profile sections of a water body, the quantitative assessment will be conducted only for those parts. In particular, the data points on that profile sections will be extracted and then compared with the relevant ground truth ones. The comparison is based only on the z differences among them.

6 RESULTS AND DISCUSSION

This chapter presents the classification results and the validation analysis of this research. The classification results of the pulse and neighbourhood (voxel) based methods are discussed. Different datasets are used to test these approaches. The validation is done by making a comparison with ground truth data. The computation time and scalability are also discussed in this chapter.

6.1 CLASSIFICATION RESULTS OF DATASETS

First, the overall performance of the proposed methods is described in detail in Chapter 4. Five different water bodies (see Figure 6.1) from the six regions in the Netherlands, as presented in Section 5.1.1, are chosen to implement the algorithm. As seen in Table 6.1, the selected water bodies vary in area size, number of points and point density.

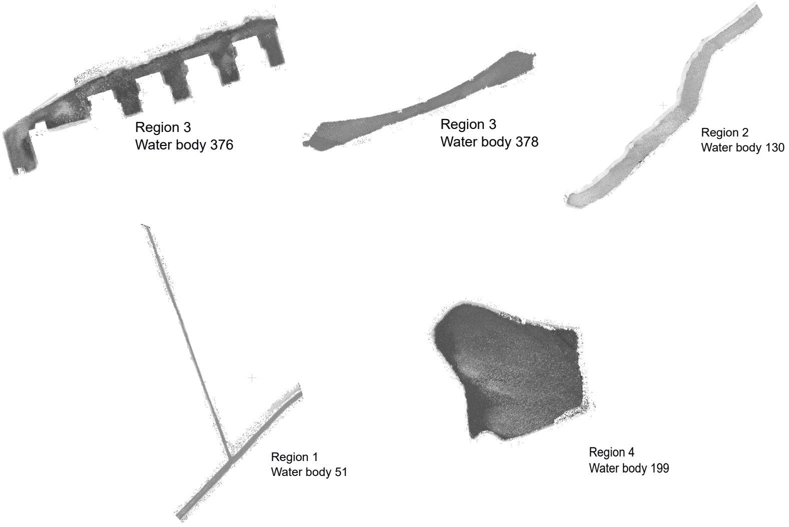


Figure 6.1: Water bodies extracted from different regions

The water bodies are extracted from the 4 out of 6 regions, as these regions showed variation in the size and density of the water areas (see Table 6.1). Also, the range in z level was chosen after a visual inspection of each dataset. Thus, the z boundaries can vary per water body in order to contain only water points (i.e. water surface till bottom part). The area size and number of points differ per water area and have an important role in the neighbourhood-based approach; particularly, the selection of voxel size (see Section 4.3.1). The point density is related with the amount of points per unit area of 1 m^2.

Region	Name	Z Range (m)	Area size (km2)	Number of points	Point Density (points/m2)
1	NL1_51	-3.6 to -2.6	168,03	511.914	∼25 to ∼100
2	NL2_130	-2 to 0	91,31	1.164.170	∼35 to ∼130
3	NL3_376	-4.2 to -3	23,75	2.033.586	∼325 to ∼660
3	NL3_378	-4.5 to -3.3	12,8	607.216	∼250 to ∼430
4	NL4_199	1 to 4	23,9	3.256.278	∼215 to ∼250

Table 6.1: Characteristics of chosen water bodies

As shown in Table 6.1, the datasets NL1_51, NL3_376 and NL3_378 correspond to shallow water areas as their z range is 1m and below. However, these three datasets differ in area size and number of points. The NL1_51 dataset has almost half a million points in an area of 168,03 (km2), whereas the NL3_376 and NL3_378 cover smaller areas of approximately 24 and 13 (km2), respectively. In terms of number of points, the NL3_376 is denser than the NL3_378 with four times as many points.

The NL2_130 and NL4_199 datasets are selected as they are deeper with a depth of 2-3m, compared to the previous datasets. The NL2_130 has just over 1 million points in an area size of 91,31 (km2), whereas the NL4_199 has around 3,3 million points in a small area of 23,9 (km2).

Also, the selected datasets present different point density (points/m2). This factor indicates how dense or sparse is a point cloud based on the amount of points that are concentrated in 1 (m2) area. It has been measured in different parts of each water body in order to obtain its range, as there may be dense and sparse parts. The NL1_51 has the smallest range of (25 - 100) points in 1 (m2), whereas the NL3_376 contains the biggest amount of points from 325 up to 660. The other datasets present different point densities as shown in Table 6.1.

Therefore, all these five datasets are representative to cover various cases of water bodies, which differ in area size, number of points, z range and point density. Pulse and voxel-based approaches will be performed using these datasets to evaluate the classification results and methods' performance.

6.1.1 Pulse based approach

As mentioned in Section 4.2, the LIDAR points have been sorted per GPS time and based on RN, NR and their GPS time, grouped per laser pulse. For this water body, the pulse approach is able to group the points and then assign a classification code for each point. The classification codes are three: water surface, underwater and bottom points. Also, there is an extra class with all the unclassified points. In particular, they are left over from pulses that have been cut in the pre-processing stages and their RN, NR and their GPS time does not correspond to the previous classes.

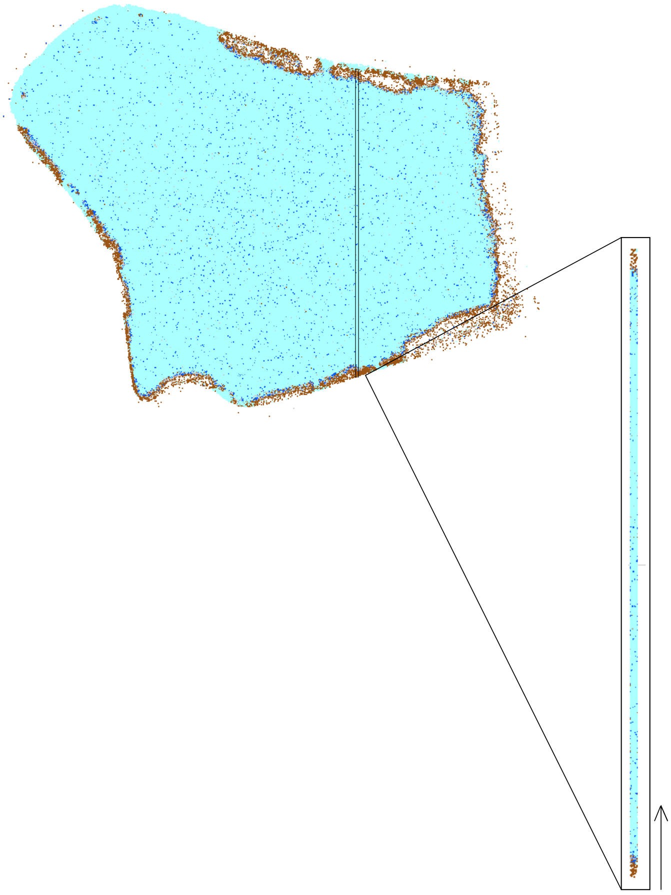


Figure 6.2: Classified water body (NL4_199) into: water surface (light blue), underwater (dark blue) and bottom (brown) points. Vertical section of the dataset (rectangular shape) is selected with 1m width

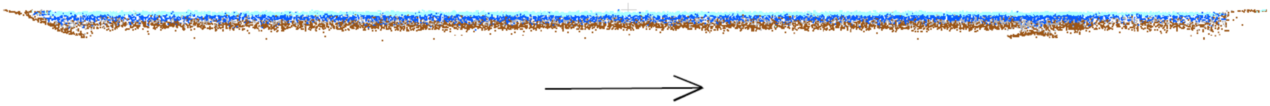


Figure 6.3: Water body (NL4_199): Profile section with the classified points

Figure 6.2 shows the results based on the pulse approach. Light blue points (1.016.207) correspond to the water surface, the dark blue (723.379) to the underwater, while the brown (760.894) ones to the bottom surface. This water body (NL4_199) is a small lake in reality with around 2.800.000 in total. As seen in Table 6.1, its point density varies from 215 to 250 points per square meter, while the point distribution is uniform everywhere. Therefore, the amount of points per unit area is large, and thus no gaps exist in the water area. The laser pulse managed to penetrate the water body and likely reach the bottom part, since based on the pulse approach bottom has been detected. A small part was randomly selected to visualize the points through a section. Its side view shows the horizontal distribution of points in this area (see Figure 6.3). There are a lot of bottom points (brown colour) that can represent the bottom surface, while the left edge can be clearly seen.

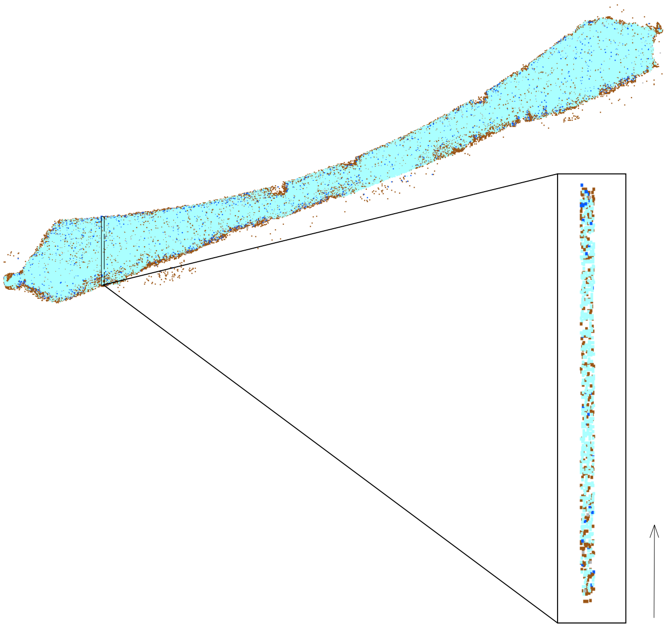


Figure 6.4: Classified water body (NL3_378) into: water surface (light blue), underwater (dark blue) and bottom (brown) points. Vertical section of the dataset (rectangular shape) is selected with 1m width

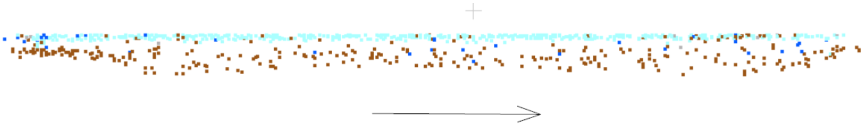


Figure 6.5: Water body (NL3_378): Profile section with the classified points

Figure 6.4 presents a water body (NL3_378) with 351.031 points that belongs to an inland water canal. A small part (rectangular shape) was randomly selected in order to visualize the classified points through a section. Its side view shows the horizontal distribution of points in this small part (see Figure 6.5). The water surface (light blue) points have been correctly classified as they form a horizontal surface with the higher z values. As far as the bottom (brown) points, they do exist in the whole extent of the section and are usually the lower ones. There are more brown than dark blue dots in the water part, as more pulses consist of just two returns. The first is classified as water surface and the last as bottom point.

However, there are many points that are classified as bottom ones, but they probably belong to the middle water (dark blue) part. This happens because many pulses may contain only two returns, as they didn't manage to continue deeper in the water body. Therefore, their last return is near the water surface and is classified as bottom point.

In addition, few brown dots (i.e. bottom points) exist on the water surface. The reason is that these points are the last returns of laser pulses that were eliminated in the pre-processing stage. The same applies for the dark blue dots near or on the same water surface level.

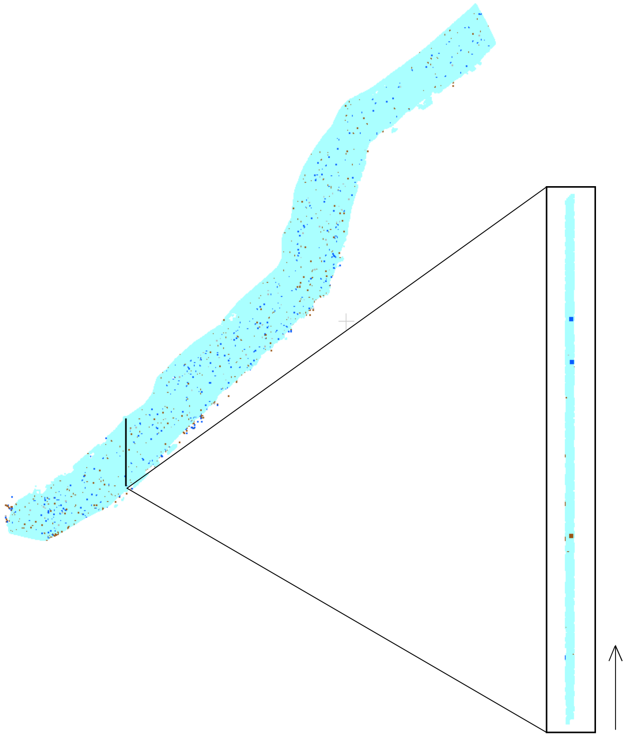


Figure 6.6: Classified water body (NL2_130) into: water surface (light blue), underwater (dark blue) and bottom (brown) points. Vertical section of the dataset (rectangular shape) is selected with 1m width

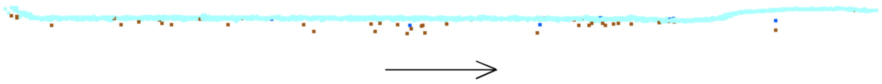


Figure 6.7: Water body (NL2_130): Profile section with the classified points

Another water body (NL2_130) with around 1.164.000 points has been classified, as seen in Figure 6.6. More than two thirds (971.954) of all the points belong to the water surface, while several points (1.456) are only underwater and 42.936 are classified as bottom ones. Similarly, a random section from the initial water body was created (see Figure 6.6). The distribution of points across this section is displayed. The majority of points have been categorized as water surface (light blue), while rarely few are only underwater (dark blue) and just few points correspond to the bottom part. It is obvious that there are not so many below the water surface. The point density is too low (below 200 points) compared to other water bodies, as seen in Table 6.1.

Moreover, there are not many misclassified points on the water surface, but only two or three underwater (dark blue). The bottom points are sparse to each other and in some parts do not exist points.

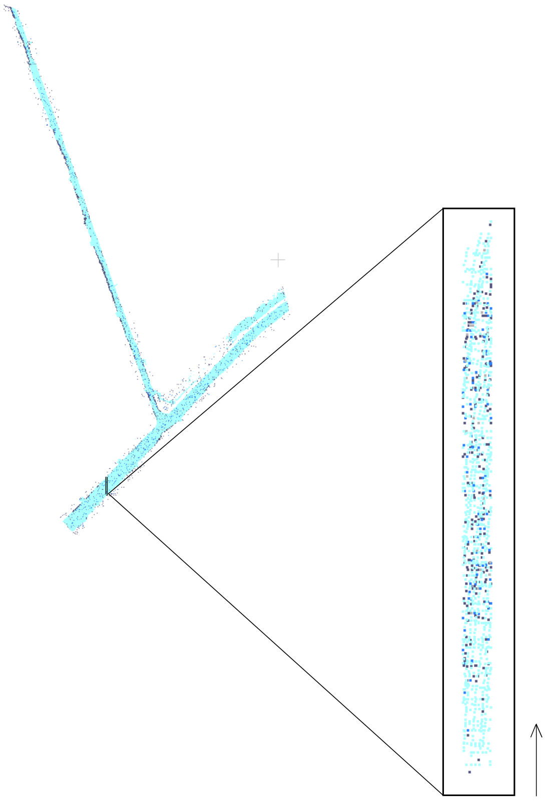


Figure 6.8: Classified water body (NL1_51) into: water surface (light blue), underwater (dark blue) and bottom (brown) points. Vertical section of the dataset (rectangular shape) is selected with 1m width

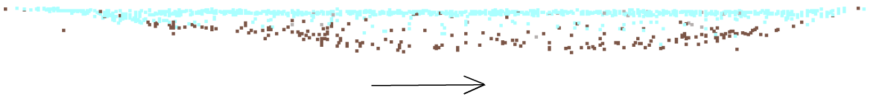


Figure 6.9: Water body (NL1_51): Profile section with the classified points

Figure 6.8 shows the results for the water body (NL1_51) from the region 1. In this case, the majority of the points (282.818) have been selected as water surface, while the bottom points are more than the one third (71.909) of the total (391.309) points. However, the distribution of points is different compared to the other cases. Again a section was randomly selected. Even if the water surface is constructed with a big amount of points, bottom points do exist in the whole extent of the section (see Figure 6.9). However, there are not so many (dark blue) points. This means that there are more pulses with 2 returns. Also, the water surface points do exist a bit lower in the water. This means that individual points, with RN=1 and NR=1, exist and classified in the water surface.

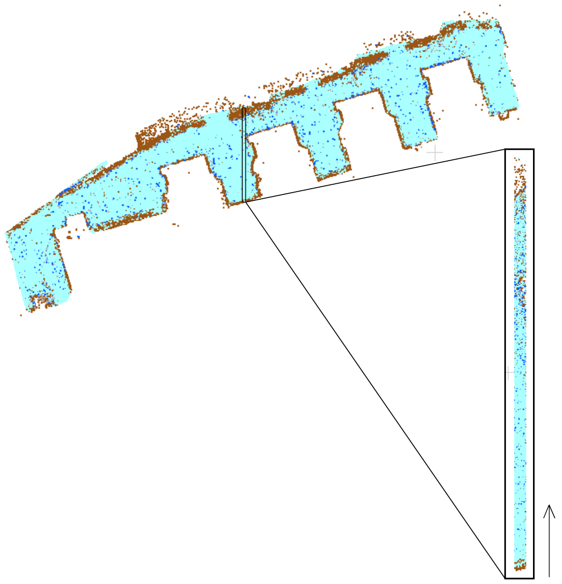


Figure 6.10: Classified water body (NL3_376) into: water surface (light blue), underwater (dark blue) and bottom (brown) points. Vertical section of the dataset (rectangular shape) is selected with 1m width

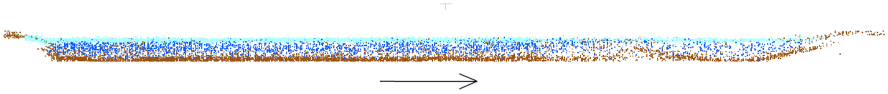


Figure 6.11: Water body (NL3_376): Profile section with the classified points

The last used water body is in the region 3 and contains a bit more than 2 million points (2.033.586). The classified water surface points are just below the 1 million; specifically (900.114). The underwater points are (256.504), while the bottom ones are more than the one quarter (630.772) of the total points. Again a random section was extracted from the initial water body (NL3_376) (see Figure 6.10) and its side view is completely different than the other datasets. In particular, it is evident that the density of points is quite high; between 325 to 660 points per 1 square meter (see also Table 6.1). The water surface (light blue) is displayed without gaps and the bottom (brown) part is also clear and uniform in the whole extent of the section. In addition, many underwater (dark blue) points exist, which means that the laser managed to collect more than two returns per pulse.

Table 6.2 summarizes the number of classified points per class for the five tested datasets. It is important to mention that there are a few unclassified points for all the cases. The percentage of these points in the total amount of points varies per dataset from 1% up 10%. The percentage is calculated by dividing the number of unclassified by the total points for each case. Moreover, the presence of unclassified points does not prevent the detection of bottoms in most cases, while the water surface has been detected in all datasets. The presence of short pulses (up to two returns) affects the quality, in terms of density, of a dataset. According to this pulse approach, all the last returns are classified as bottom surface, but this is not correct in reality. Many pulses are affected by external environmental factors (see Section 2.2.2) and did not manage to penetrate more the water body and reach its water bed. This is obvious in Figure 6.7, where many bottom points are located near the water surface in such a small area of the entire water body.

Region	Name	Water surface	Underwater	Bottom	Unclassified	Total
1	NL1_51	282.818	30.477	71.909	6.105	391.309
2	NL2_130	971.954	7.155	42.936	1.456	1.023.501
3	NL3_376	900.114	256.504	630.772	85.152	1.872.542
3	NL3_378	365.361	19.493	155.780	8.285	548.919
4	NL4_199	1.016.207	723.379	760.894	351.032	2.851.512

Table 6.2: Pulse-based classification results for the five datasets; number of points per class

6.1.2 Neighbourhood based approach

As mentioned in Section 4.3, the 3D space is divided into voxels or water columns. The LIDAR points that fall into every voxel are detected, stored and grouped per voxel. In this section, the voxel-based method was applied to classify a water-body into water surface, underwater and bottom points. Then, the results of various datasets with different voxel sizes are presented and discussed. A workflow of this implementation is visualised in Figure 5.7, which presents the process from an unclassified to a classified water body.

First of all, the median and mean values of the density, distance and intensity are calculated in order to get an insight about their range in the entire dataset. As seen in Table 6.3, the median and mean values of the NormDensity do not differ so much (i.e. almost 2 points), while the values for NormIntensity is around 0.6 and for the NormDistance 13cm.

Parameters	Mean	Median
NormDensity	7.81	**6**
NormDistance	**0.13**	0
NormIntensity	0.52	**0.6**

Table 6.3: Water body (NL1_51): Mean and Median Values of three parameters

Figure 6.12 illustrates the classified bottom of water body (NL1_51) based on confidence values as presented in Table 4.1. It is evident that two green shades show the first two categories (1,2). This means that the points that belong to them are very likely to belong to the bottom surface of the water body. In addition, classes (7,8) have the minimum confidence and presented with blue shades. Their points only exist in a few small parts of the point cloud. Also, points in class (8) are presented with an light red shade colour. In Table 6.4, the amount of points per class is shown. The classes (1,2,8) collect the majority of the points in this dataset.

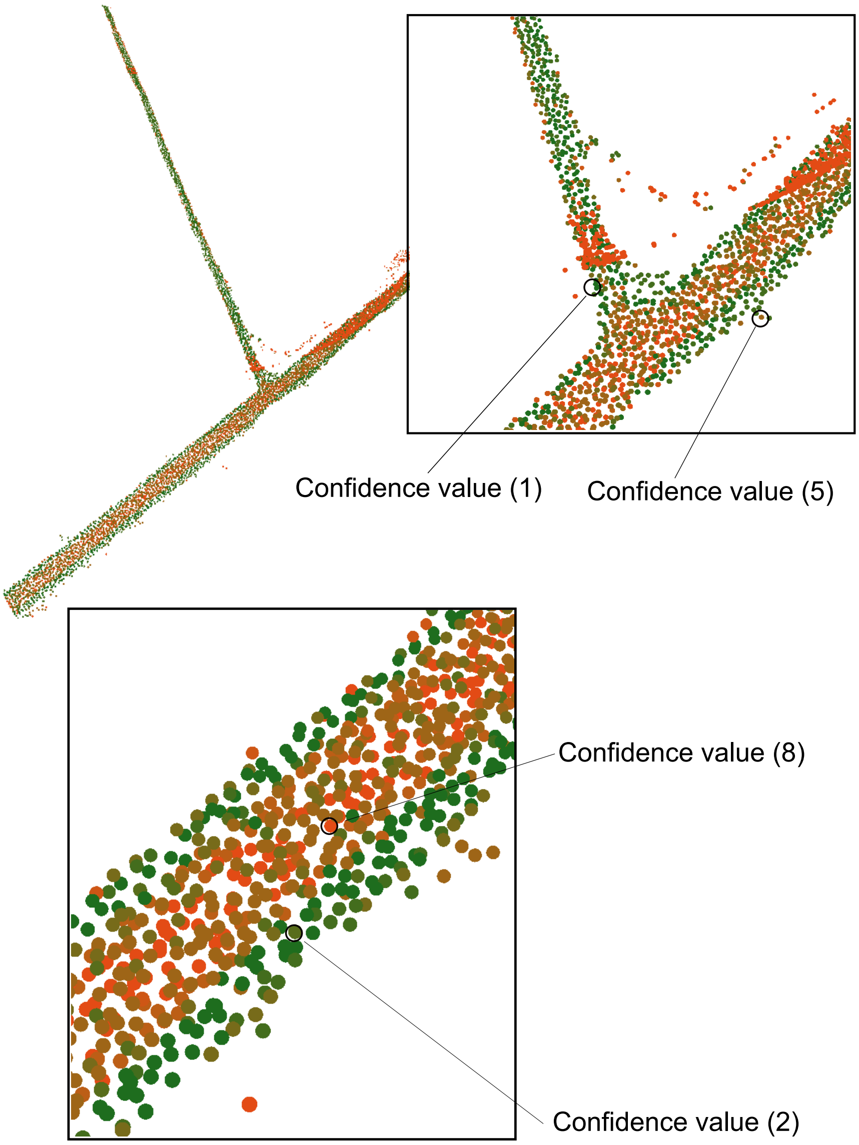


Figure 6.12: Water body (NL1_51): Classified with confidence values

Name	51NL1
Class 1	1396
Class 2	1950
Class 3	293
Class 4	1070
Class 5	1256
Class 6	984
Class 7	391
Class 8	2307
Total	9647

Table 6.4: Water body (NL1_51): Number of points per class

Furthermore, the voxel based approach was tested with another water body (NL4_199) dataset. As seen in Table 6.5, the *median* and *mean* values were computed for the three normalized parameters (NormDensity, NormDistance, NormIntensity).

The median values of both NormDensity and NormIntensity are used as threshold values, while the mean value for the NormDistance. It is important to mention that the mean and median values of the NormIntensity are exactly the same. This means that NormIntensity values probably follow the normal distribution and not a bimodal distribution. The intensity values of the lowest points range in a certain amplitude, as they belong to the deepest part of the voxel and not near to the water surface, where the values are high. Thus, it is unlikely that there is a bimodal distribution.

Parameters	Mean	Median
NormDensity	0.001	**0.004**
NormDistance	**0.25**	0
NormIntensity	0.194	**0.194**

Table 6.5: Water body (NL4_199): Mean and Median Values of three parameters

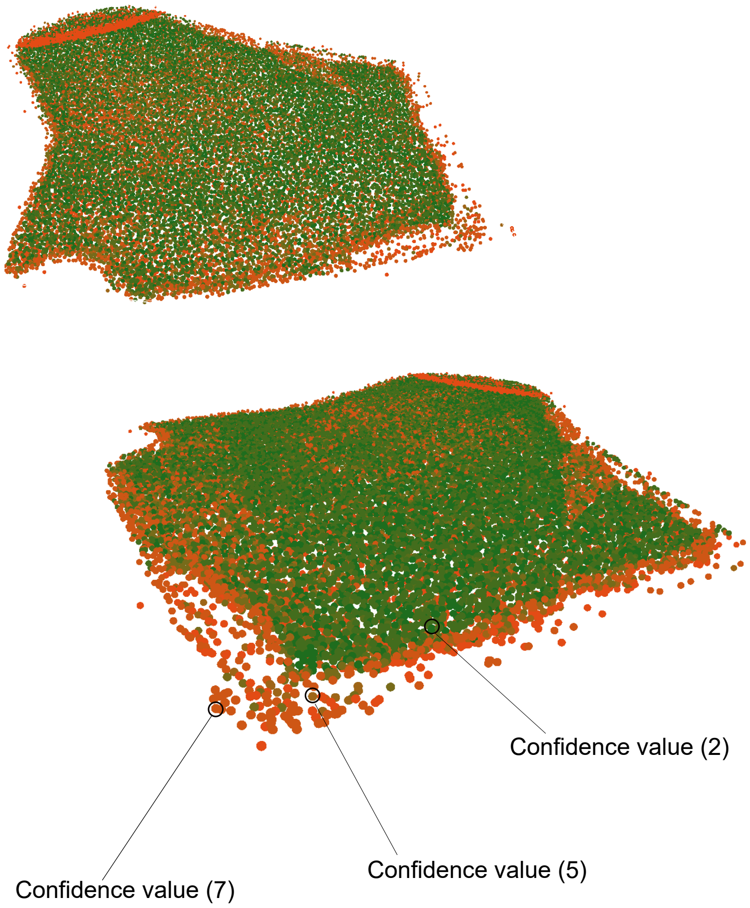


Figure 6.13: Water body (NL4_199): Classified with confidence values

Name	199NL4
Class 1	3882
Class 2	3423
Class 3	1913
Class 4	2050
Class 5	1924
Class 6	181
Class 7	1738
Class 8	7153
Total	22264

Table 6.6: Water body (NL4_199): Number of points per class

Figure 6.13 displays the classified bottom surface of the water body (NL4_199). All the eight classes are presented in this point cloud. Both classes (1,2) contain more than a third (7.305) of the total points. They have NormDensity and NormIntensity above their threshold values, while the distance is above and below its threshold, respectively. This means that the amount of points and the intensity values are quite big in both cases. Also, the fact that many points (7.153) in class (8) have distances below the mean value, it makes clear two things. First, the depth of the corresponding voxels is small as the laser pulse captured points in a low z level. Second, for all the pulses with just one peak below the mean z, their points are near to the bottom part and have distance zero. Thus, all these points are concentrated in class (8). Regarding the class (4), the number of points is quite big (2.050), but it has distances bigger than the mean distance value.

Figure 6.14 displays the classified bottom surface of the water body (NL3_376). All the eight classes are presented in this point cloud. Both classes (2,4) contain more than a quarter (3.321) of the total points. These classes have NormIntensity above their threshold values, while the distance is below and above its threshold, respectively. The NormDensity is above and below the threshold values in the two classes, too. Also, the fact that many points (2.103) in class (6) have distances below the mean value, it makes clear that the depth of the corresponding voxels is small as the laser pulse captured points in a low z level. Also, the points (near to the bottom part) of the pulses with just one peak below the mean z have distance zero. As seen in Table 6.7, quite few points are concentrated in class (1) with the highest confidence level. Regarding the class (8), the number of points is quite sizeable (1.369), but it has distances bigger than the mean distance value.

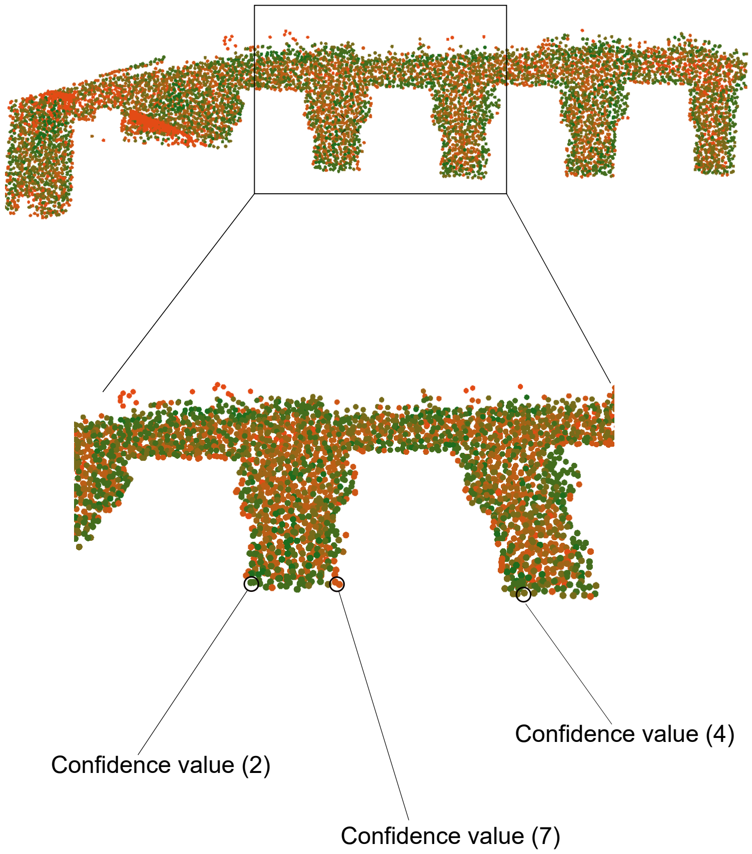


Figure 6.14: Water body (NL3_376): Classified with confidence values

Name	376NL3
Class 1	338
Class 2	1636
Class 3	96
Class 4	1685
Class 5	618
Class 6	2103
Class 7	523
Class 8	1369
Total	8368

Table 6.7: Water body (NL3_376): Number of points per class

Moreover, the voxel-based approach was tested with another two water bodies (NL3_378) and (NL2_130). The Table 6.8 presents the amount of classified points in the eight classes. It is evident that the one third (1.007) of the total points belongs to class (1), while another third (1.069) belongs to the less confident class (8). Figure 6.15 shows the classified bottom surface of the water body (NL3_378). In particular, the dark green points correspond to the high confident bottom points, which are distributed throughout the whole area of the water body. The brown shadow points belong to the less confident classes (7,8) and they also exist in all parts of this water area.

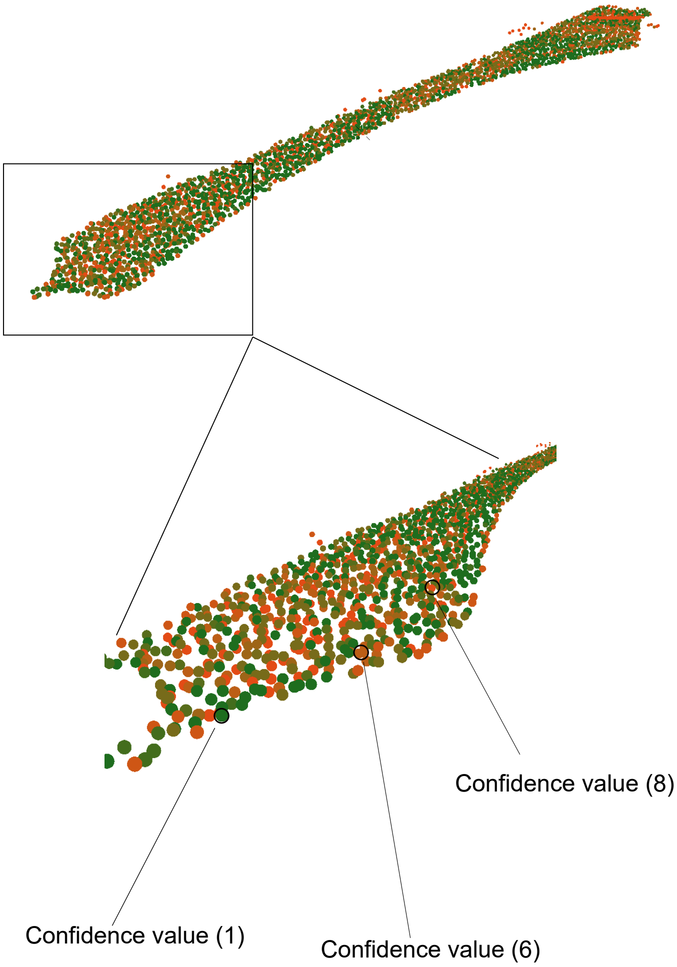


Figure 6.15: Water body (NL3_378): Classified with confidence values

Name	378NL3
Class 1	1007
Class 2	96
Class 3	211
Class 4	577
Class 5	191
Class 6	368
Class 7	149
Class 8	1069
Total	3668

Table 6.8: Water body (NL3_378): Number of points per class

Figure 6.16 displays the classified bottom points of the water body (NL2_130). As shown in Table 6.9, more than the half amount (6.609) of total points (11.636) are concentrated in the classes (1 up to 4). Specifically, the high confident class (1) contains (1.059) points, which are distributed near the shore of the water body.

Indeed, these points belong to the shore part of the water body, as they have been captured by LIDAR and are the lowest points of voxels. In the middle part of the water body, the points belong to the less confident classes (7,8) as they are deeper in the z range and the distance between the first and last peak is below the threshold value.

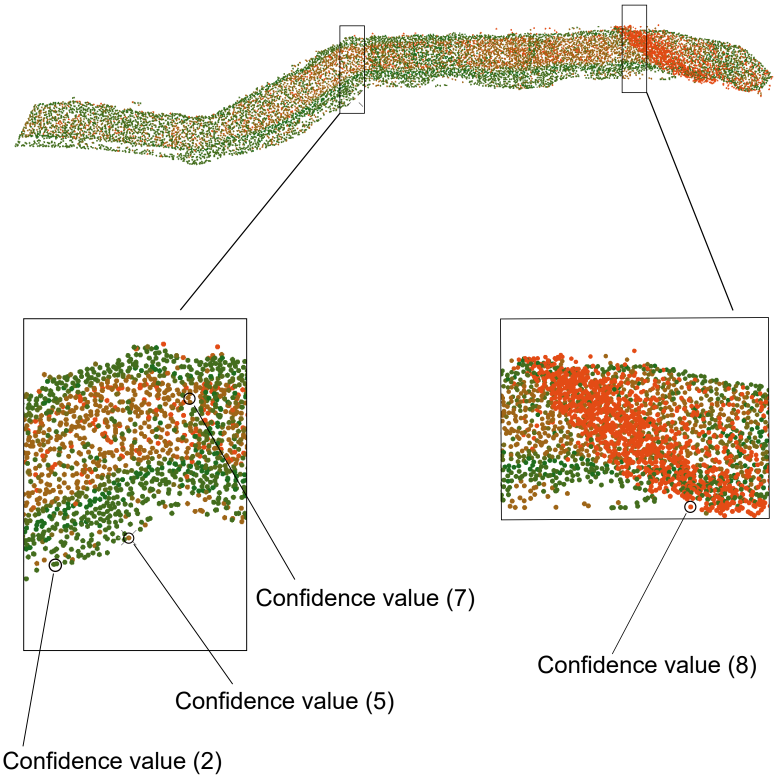


Figure 6.16: Water body (NL2_130): Classified with confidence values

Name	130NL2
Class 1	1059
Class 2	3502
Class 3	987
Class 4	1061
Class 5	2808
Class 6	51
Class 7	189
Class 8	1979
Total	11636

Table 6.9: Water body (NL2_130): Number of points per class

Applying various voxel sizes

As the voxel-based approach was applied to five datasets, different voxel sizes were used to detect how the number of voxels generated and the lowest (bottom) points were affected. As seen in Table 6.10, the number of voxels decreases rapidly as the voxel size increases, while the number of lowest points follows the same trend. It is important to note that the voxelization has not been performed with 0.5m voxel size in most of the used datasets; due to the high and inefficient computation time. Only in the case of NL2_130 dataset was this done, but the runtime was extremely high.

Name	Total	Voxel Size	Voxels	Lowest points	Time (sec)
		0.5m x 0.5m	-	-	
		1m x 1m	13.048	9.647	305
NL1_51	511.914	2m x 2m	4.435	2.933	105
		3m x 3m	2.493	1.605	65
		4m x 4m	1.640	1.082	48
		0.5m x 0.5m	57.124	37.393	1.326
		1m x 1m	17.295	11.636	457
NL2_130	1.164.170	2m x 2m	5.320	3.836	150
		3m x 3m	2.665	1.960	85
		4m x 4m	1.620	1.272	62
		0.5m x 0.5m	-	-	-
		1m x 1m	10.163	8.368	293
NL3_376	2.033.586	2m x 2m	2.868	2.419	107
		3m x 3m	1.416	1.232	49
		4m x 4m	862	766	30
		0.5m x 0.5m	-	-	-
		1m x 1m	4.402	3.668	141
NL3_378	607.216	2m x 2m	1.272	929	44
		3m x 3m	637	439	25
		4m x 4m	397	292	12
		0.5m x 0.5m	-	-	-
		1m x 1m	28.504	22.264	1.214
NL4_199	3.256.278	2m x 2m	7.458	6.152	525
		3m x 3m	3.417	2.902	300
		4m x 4m	1.916	1.705	187

Table 6.10: Voxel based method: general characteristics for the five classified water bodies

Figure 6.17 illustrates the classified point cloud of water body (NL1_51) using different voxel sizes. As shown in Table 6.10, the number of lowest points has been significantly decreased from 9.647 to 1.082 while the voxel size increased from 1m to 4m. Furthermore, a small part (rectangular shape) was randomly selected in order to visualize the classified points through sections (see Figure 6.18). Its side view visualizes the horizontal distribution of points in this small part. Again, the number of points is smaller as the size of voxel increases, while gaps are created in the bottom surface. In particular, the first section (1m) has bottom points in its whole extent, where green points (high confidence) do exist near its borders. There are more red dots in the middle water part, as these points are less confident (class 7,8). This is because the NormDistance of these points is below the corresponding threshold value.

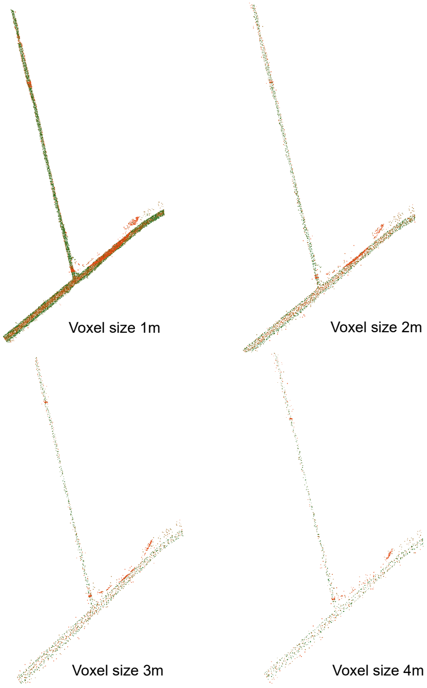


Figure 6.17: Water body (NL1_51): Classified with different voxel sizes: 1m, 2m, 3m, 4m

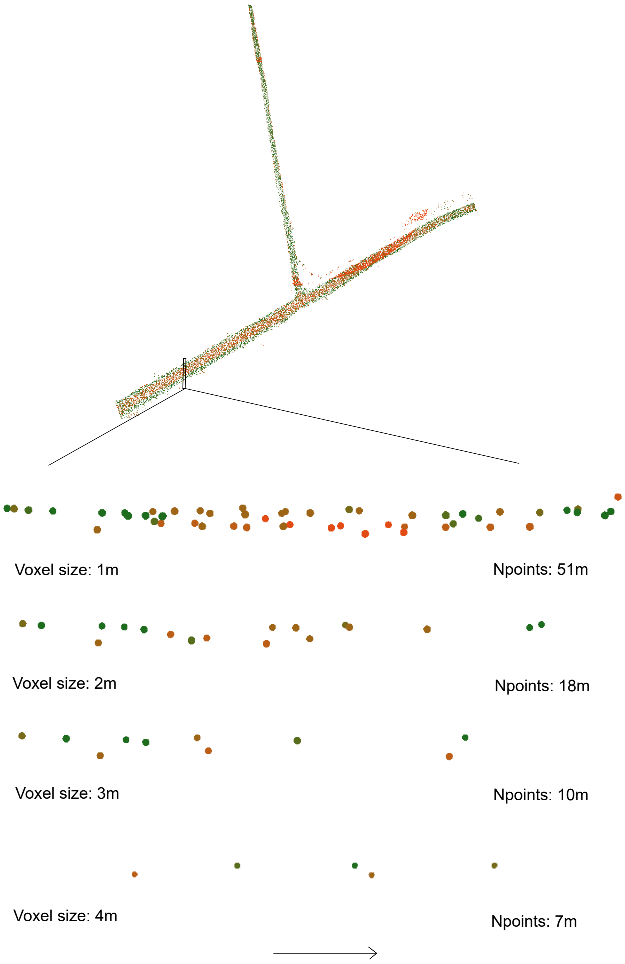
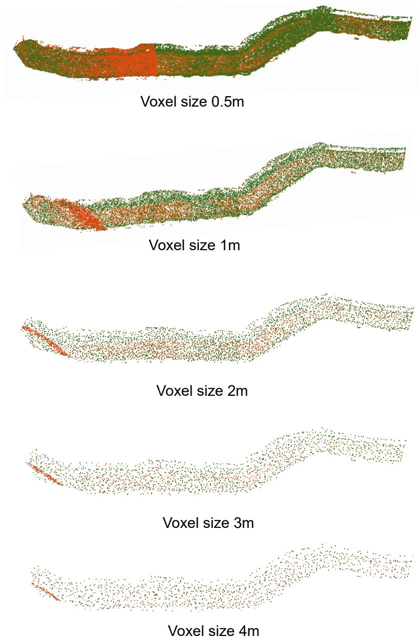


Figure 6.18: Water body (NL1_51): Classified sections with different voxel sizes

Furthermore, Figure 6.19 illustrates the classified point cloud of water body (NL2_130) using different voxel sizes. As shown in Table 6.10, the number of lowest points has been rapidly decreased from 37.393 to 1.272 while the voxel size increased from 0.5m to 4m. Similarly, a small part (rectangular shape) was randomly selected in order to visualize the classified points through sections (see Figure 6.20). The distribution of points across this section is displayed.


Voxel size 0.5m

Voxel size 1m

Voxel size 2m

Voxel size 3m

Voxel size 4m

Figure 6.19: Water body (NL2_130): Classified with different voxel sizes: 0,5m, 1m, 2m, 3m, 4m

The number of points drops as the size of voxel increases, while gaps are created in the bottom surface of sections with voxel size 3m, 4m and 5m. The first section (1m) has bottom points in its whole extent, where few green points (high confidence) do exist near its borders. There are many points (red shades) in the middle water part, as these points are less confident (class 7,8).

Figure 6.21 presents the classified water body (NL3_376) with different voxel sizes. A small part (rectangular shape) was randomly selected in order to visualize the classified points through sections (see Figure 6.22). The number of points decreases as the size of voxels increases, while gaps are created in the bottom surface of the last two sections. It is obvious that the bottom surface can not be reconstructed using so few bottom points.

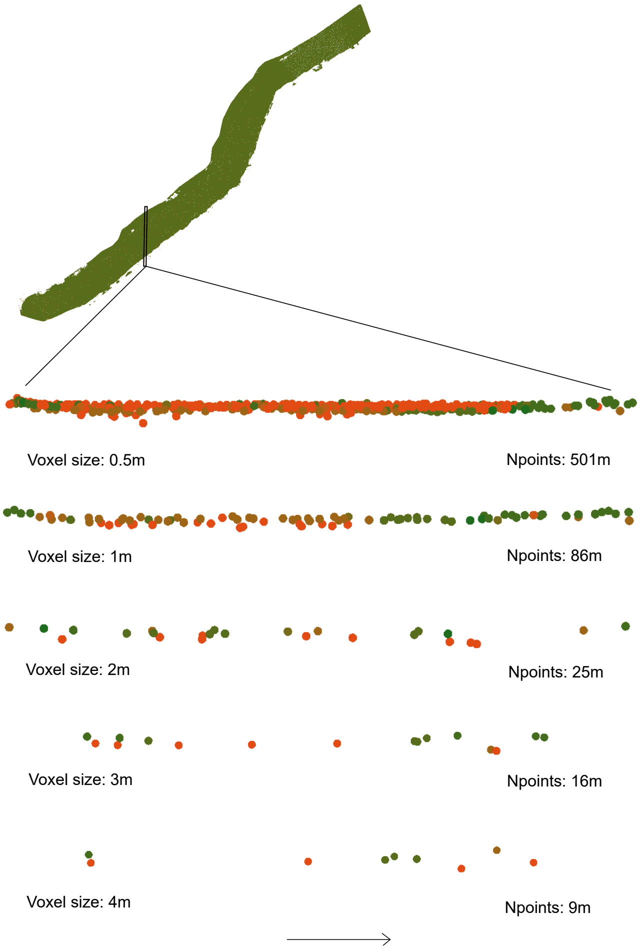


Figure 6.20: Water body (NL2_130): Classified sections with different voxel sizes

Figure 6.23 shows the results of the classified water body (NL3_378) with different voxel sizes. Again, a small part (rectangular shape) was randomly selected in order to visualize the classified points through sections (see Figure 6.24). The number of points decreases as the size of voxels increases. Few small gaps are created in the bottom surface of the third section, specifically in the middle part. As far as the last section, there is only one green point with high confidence and it is near the shore of the water body.

The last used water body is the (NL4_199), as shown in Figure 6.25. It has been classified similarly with four different voxel sizes. In order to study the distribution of bottom points in sections, a rectangular part was randomly chosen (see Figure 6.26). The number of points drops as the size of voxels increases. However, there are not big gaps in the bottom surface of all the sections compared to the previous datasets. Green points do exist in all parts of the sections, while red and light brown points are usually presented in the middle water part.

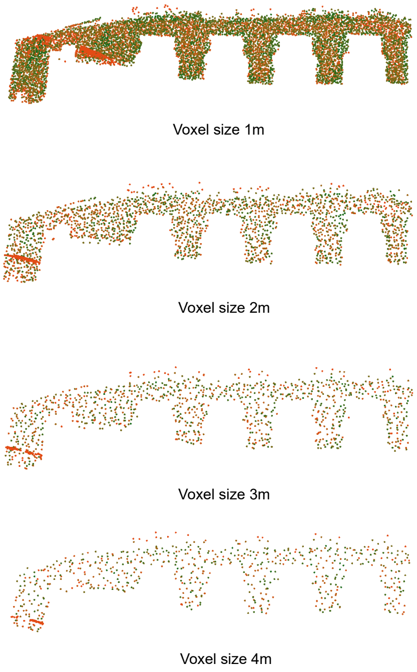


Figure 6.21: Water body (NL3_376): Classified with different voxel sizes: 1m, 2m, 3m, 4m

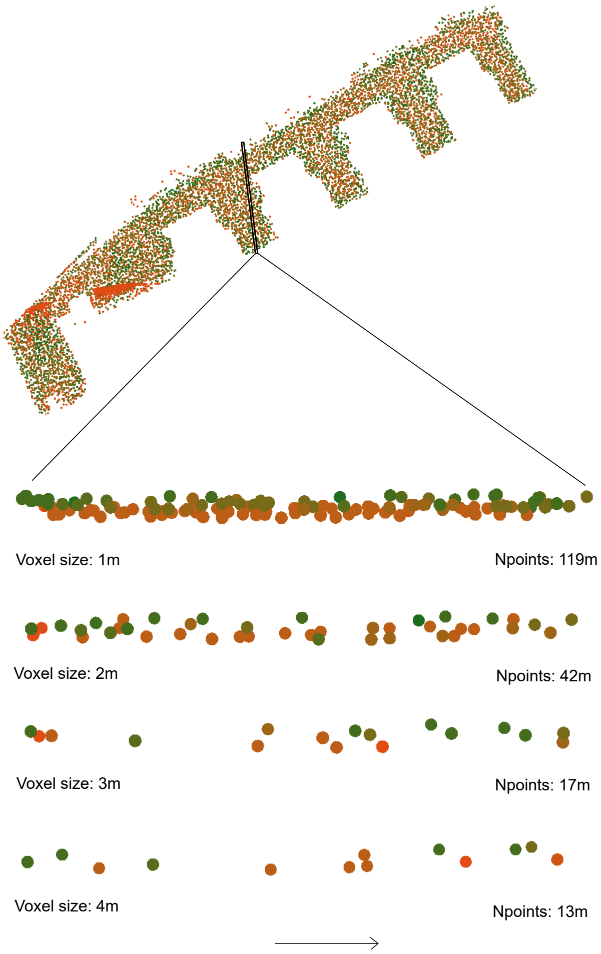


Figure 6.22: Water body (NL3_376): Classified sections with different voxel sizes

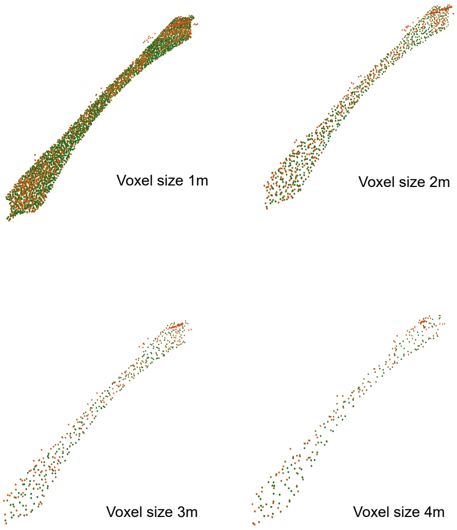


Figure 6.23: Water body (NL3_378): Classified with different voxel sizes: 1m, 2m, 3m, 4m

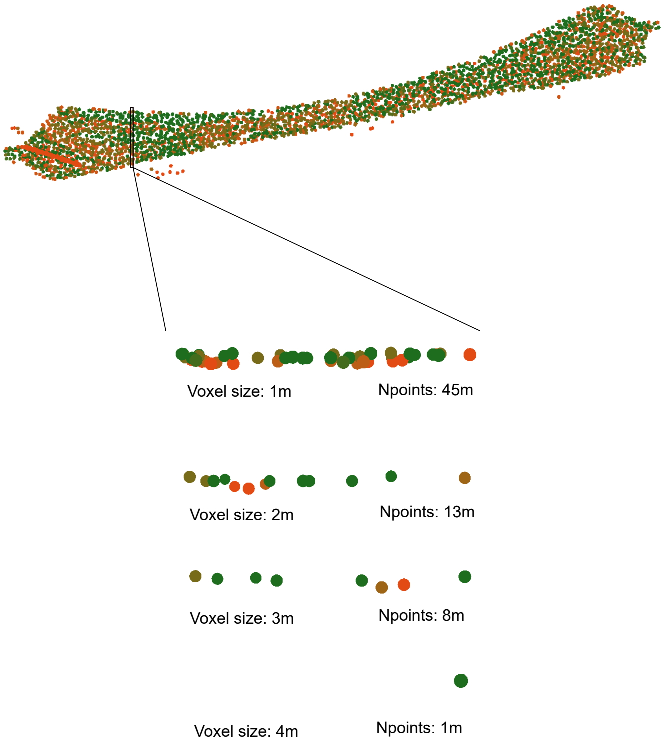


Figure 6.24: Water body (NL3_378): Classified sections with different voxel sizes

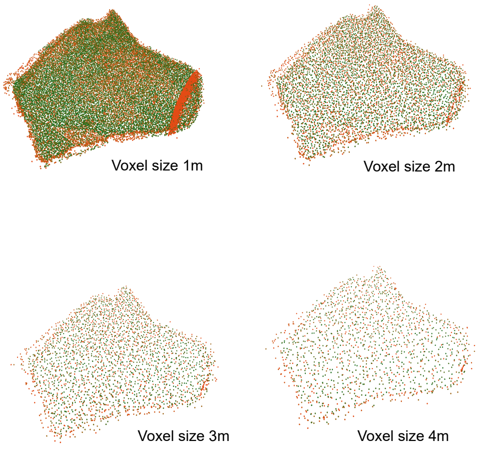


Figure 6.25: Water body (NL4_199): Classified with different voxel sizes: 1m, 2m, 3m, 4m

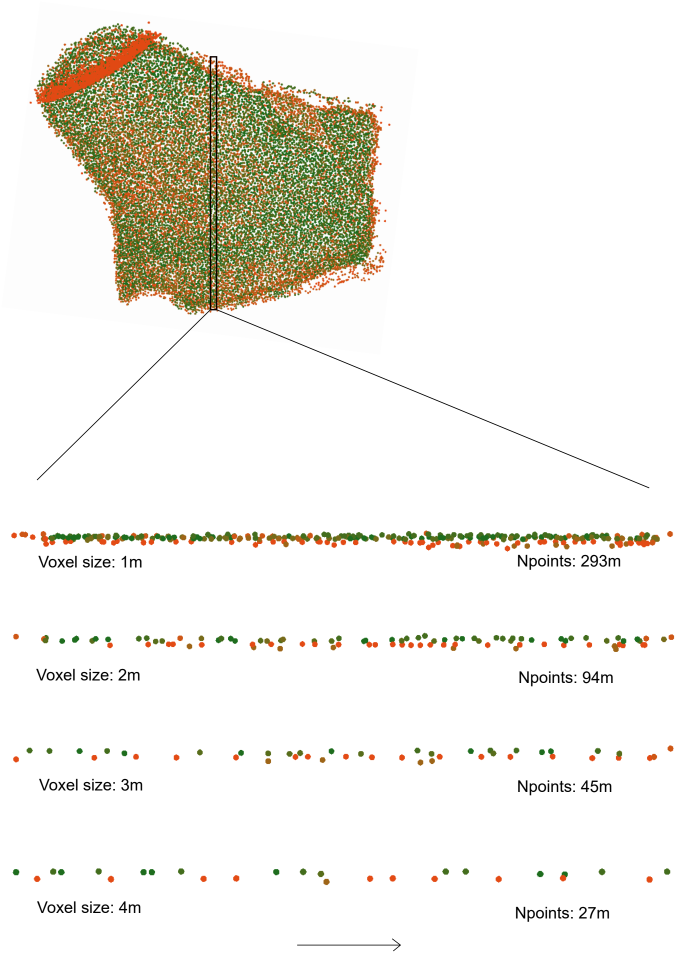


Figure 6.26: Water body (NL4_199): Classified sections with different voxel sizes

After applying the voxelization approach in various datasets with different voxel sizes, the classification results have been affected. In particular, the number of bottom points significantly decreases, as the voxel size increases. Therefore, many gaps do exist in the bottom part of the water body that make difficult to reconstruct the water bed's surface.

6.1.3 Rasterized point cloud

The above two water bodies are classified based on the three parameters (NormDensity, NormDistance, NormIntensity). The confidence value (class) of each point is assigned as a classification code in the original point cloud. Then, the points are colourized and visualized in a 3D viewer software (i.e. Displaz, CloudCompare) as seen in the Section 6.1.2.

However, the classified bottom of the water bodies can be exported into a 2D raster output or a 3D surface by triangulating them as explained in Section 2.4.2 and Section 4.4.

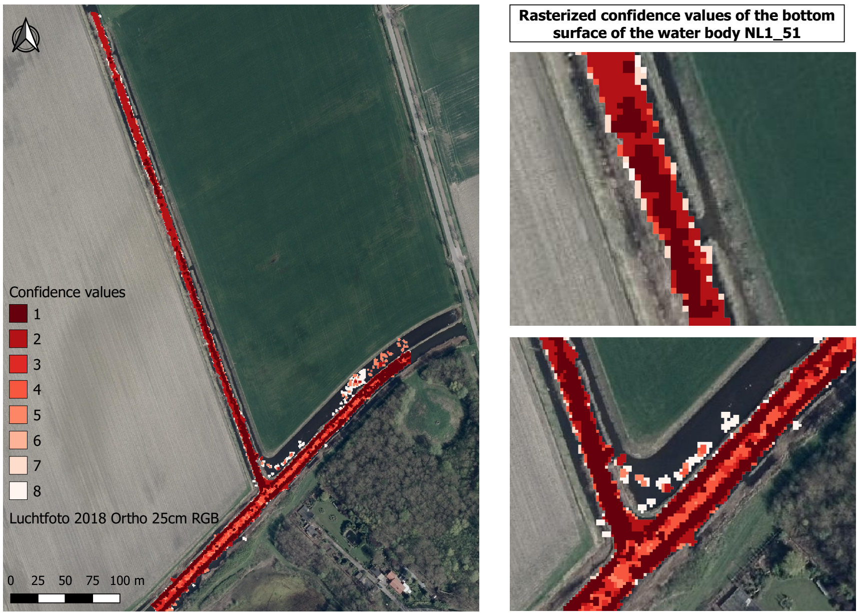

Rasterized confidence values of the bottom surface of the water body NL1_51
Confidence values
1
2
3
4
5
6
7
8
Luchtfoto 2018 Ortho 25cm RGB
0 25 50 75 100 m

Figure 6.27: Water body (NL1_151): Rasterized confidence values of bottom points with resolution 1m

Figure 6.27 displays the rasterized confidence values of the classified water body (NL1_151) on an RGB Ortho-photo of the Netherlands with 25cm resolution. It is evident that many pixels have confidence values one (1) and two (2) visualized with dark brown shades. Therefore, they are highly confident to be bottom points. Moreover, many points have confidence value five (5) (i.e. belong to the class 5) and are presented with dark orange colour. These points are concentrated in the middle part of the water body, as seen in lower right image. Another important remark is the presence of while pixels at the border of the water canals (see upper right image). These points have confidence value eight (8), and they the lowest confident to belong to the bottom surface. The reason is that the water body borders are enlarged with 5m buffer in order to contain surrounding topographic points. In most cases, that points are classified as bottom (class 1) with dark brown colour, whereas there are only a few cases that they wrongly classified.

The same rasterized water body (NL1_151) is displayed also with respect to the AHN3_05_dtm (see Figure 6.28). The AHN3_05_dtm is intended as a ground level file, while all the non-ground objects (e.g. trees, buildings) has been removed from the point cloud. Also, in Figure 6.29 rasterized with 0.5m resolution.

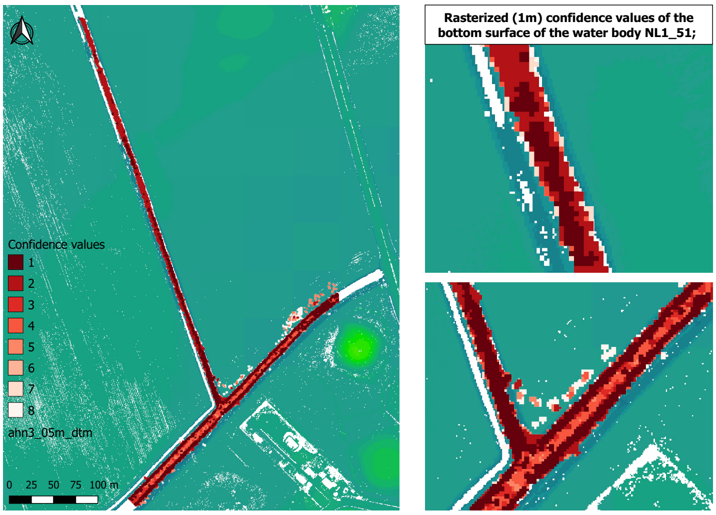

Rasterized (1m) confidence values of the bottom surface of the water body NL1_51;
Confidence values
1
2
3
4
5
6
7
8
ahn3_05m_dtm
0 25 50 75 100 m

Figure 6.28: Water body (NL1_151): Rasterized confidence values of bottom points with resolution 1m compared to the AHN3_05_dtm

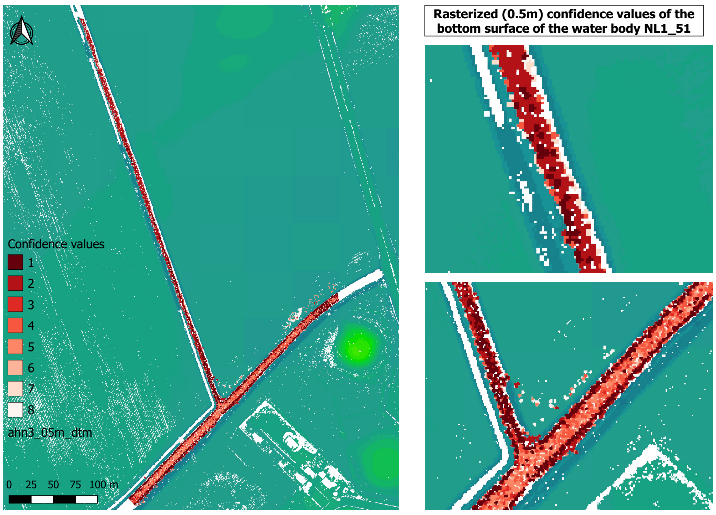

Rasterized (0.5m) confidence values of the bottom surface of the water body NL1_51
Confidence values
1
2
3
4
5
6
7
8
ahn3_05m_dtm
0 25 50 75 100 m

Figure 6.29: Water body (NL1_151): Rasterized confidence values of bottom points with resolution 0.5m compared to the AHN3_05_dtm

Moreover, the classified point cloud is rasterized based on the confidence values, but also multiple bands are created. In particular, the Band1 contains the z information, the Band2 has the density values, the Band3 the distances, the Band4 the intensity values and the Band5 has the final confidence value or class of each point. Many figures are created to visualize the information of each band of the water body (NL1_151).

As seen in Figure 6.30, the points have z value below -3 metres in the most parts of the water body.The normalized density is quite low, especially below 0.26 in the entire datasets. Only few more points may exist at the borders of the water area (see Figure 6.31). The normalized distances are usually below 0.26m and it can be seen in the upper left part of the dataset. This means that the laser pulse managed to penetrate the water around 30cm in that part. In addition, the normalized intensity is around 0.50 in the middle part of the dataset. Then, the confidence values resulted based on the threshold values of the Table 6.3 and used in Table 4.1.

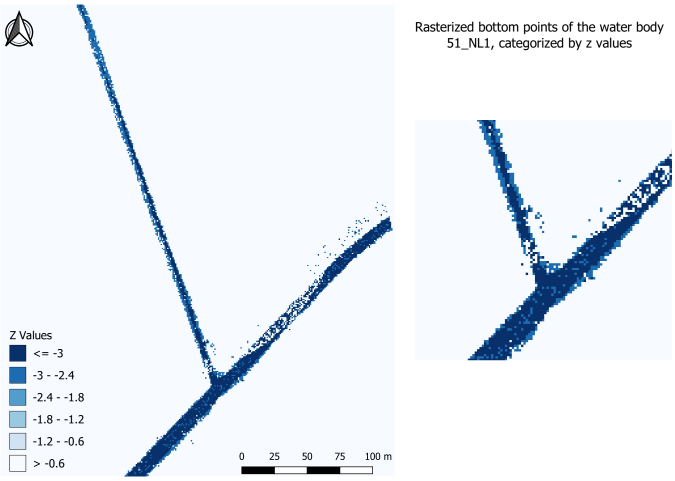

Rasterized bottom points of the water body 51_NL1, categorized by z values
Z Values
<= -3
-3 - -2.4
-2.4 - -1.8
-1.8 - -1.2
-1.2 - -0.6
> -0.6
0
25
50
75
100 m

Figure 6.30: Water body (NL1_151): Rasterized bottom points of the water body 51_NL1, categorized by z values

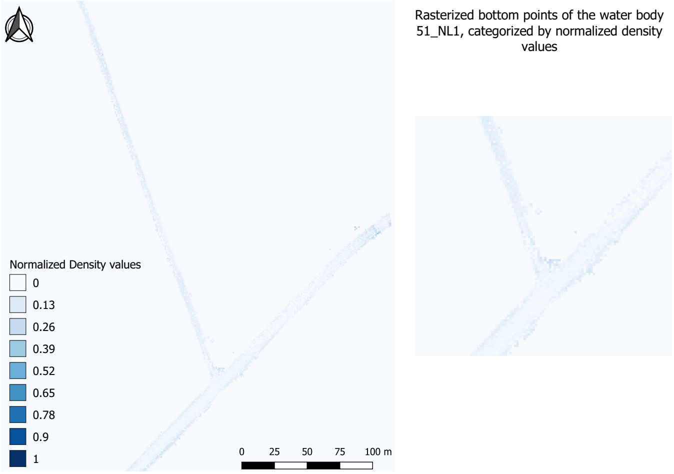

Rasterized bottom points of the water body 51_NL1, categorized by normalized density values
Normalized Density values
0
0.13
0.26
0.39
0.52
0.65
0.78
0.9
1
0
25
50
75
100 m

Figure 6.31: Water body (NL1_151): Rasterized bottom points of the water body 51_NL1, categorized by normalized density values

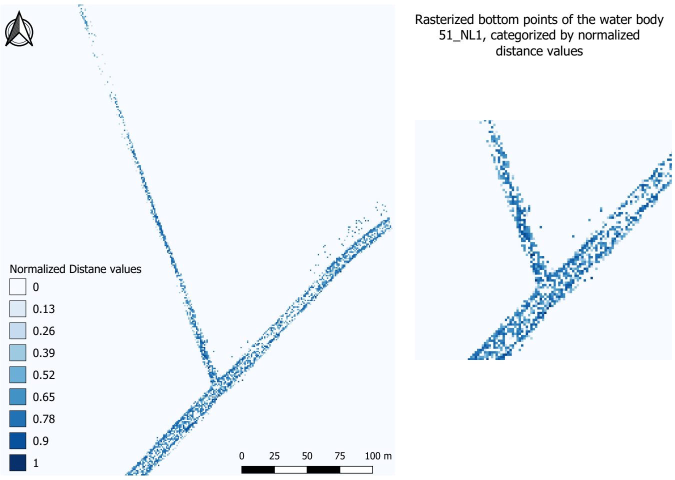

Rasterized bottom points of the water body 51_NL1, categorized by normalized distance values
Normalized Distane values
0
0.13
0.26
0.39
0.52
0.65
0.78
0.9
1
0
25
50
75
100 m

Figure 6.32: Water body (NL1_151): Rasterized bottom points of the water body 51_NL1, categorized by normalized distance values

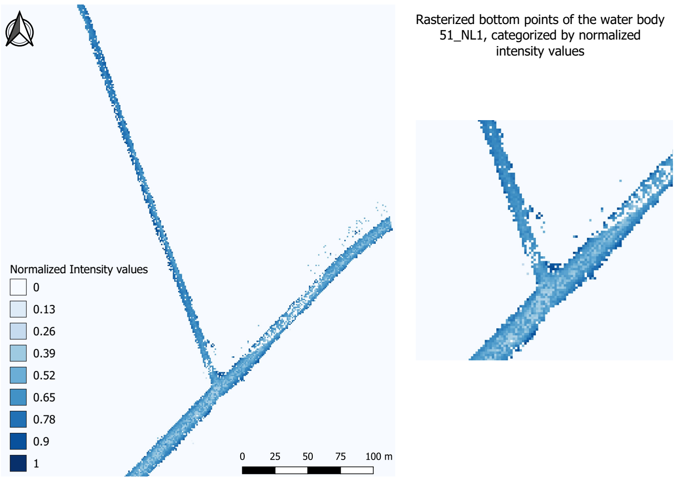

Rasterized bottom points of the water body 51_NL1, categorized by normalized intensity values
Normalized Intensity values
0
0.13
0.26
0.39
0.52
0.65
0.78
0.9
1
0
25
50
75
100 m

Figure 6.33: Water body (NL1_151): Rasterized bottom points of the water body 51_NL1, categorized by normalized intensity values

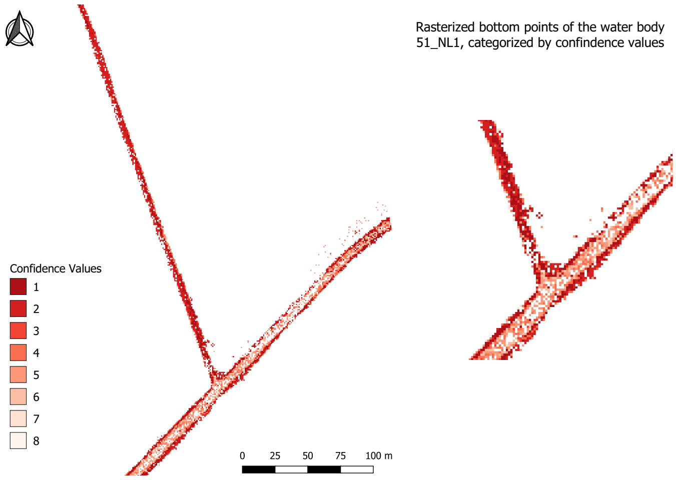

Rasterized bottom points of the water body 51_NL1, categorized by confindence values
Confidence Values
1
2
3
4
5
6
7
8
0
25
50
75
100 m

Figure 6.34: Water body (NL1_151): Rasterized bottom points of the water body 51_NL1, categorized by confidence values

6.2 VALIDATION

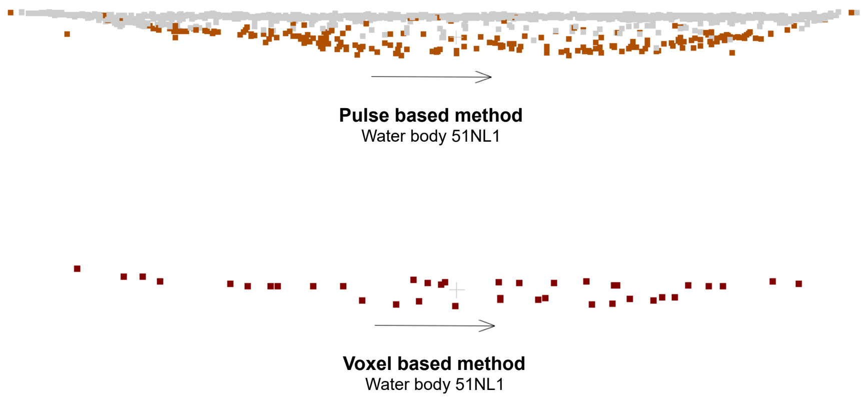
6.2.1 Pulse vs Voxel based method

To understand which method works better and gives more valuable results, each dataset needs to be evaluated. This section gives a detailed look into the performance of each approach for a given dataset (i.e. water body 51NL1). Both methods have been described in Chapter 4 and their results presented in Section 6.1. Therefore, the comparison between them in terms of accuracy and limitations are focused in this section.

As described in Section 4.2, the pulse based approach uses the point's characteristics (e.g. NR, RN, GPS time stamp) in order to group the points into the three classes, without taking into account their spatial information. On the other hand, this is completely different in voxel based approach. The 3D space is divided into a 2D voxel structure and the points grouped into voxels based on their x,y,z information. Thus, all the points that fall into a voxel have been generated from various pulses.

Entire pulse (i.e. all its returns) may belong to one voxel, depending also on the voxel size, while individual returns of other pulse can exist in that voxel, too.

For example, using both methods the detected bottom points are displayed in the Figure 6.35. In particular, the same vertical section is extracted from both classified point clouds (see Figure 6.10). By looking at the Figure 6.35, it is evident that the amount of bottom points (brown colour) in pulse approach is bigger than in the voxel one. This is logical as all the points of the dataset are used during the grouping per pulse procedure, whereas in the voxel approach only one point extracted per voxel. Also, the number of extracted points is directly affected by the selection of a voxel size. If the voxel size is large, then less bottom points are extracted. The opposite happens if it is small. In this case, the voxel size was selected to be 1m x 1m.



Pulse based method
Water body 51NL1

Voxel based method
Water body 51NL1

Figure 6.35: Water body (NL1_151): Pulse vs Voxel based method to detect the bottom points

However, it is easier to create a 3D bottom surface by triangulating the points of the voxel method. The points are not so densely distributed along the water body, and a smoother surface can be easily generated. In the case of the pulse method, the density of the points is higher and more triangles will be generated. Thus, using all the points ,but also the wrong classified ones (i.e. near the water surface), the result will a bumpy 3D surface. This surface will not be so realistic and the computation time will be certainly higher.

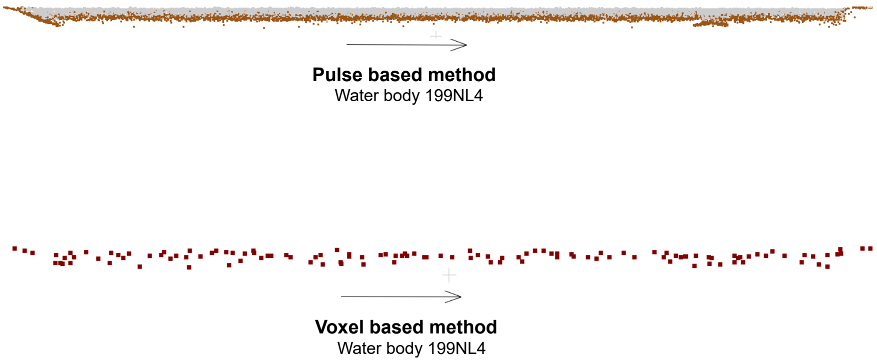


Figure 6.36: Water body (NL4_199): Pulse vs Voxel based method to detect the bottom points

Figure 6.36 shows the bottom points identified by both methods for the water body (NL4_199). The voxel-based result has fewer bottom points than the pulse-based one, as explained previously. The distribution of points in both cases is uniform everywhere and there are no gaps. Therefore, a 3D TIN bottom surface can be created in both cases.

In addition, 2D rasters using the z values of the points can be created to represent the waterbed surface in both cases (see Figure 6.30). It is also simpler to compare the classified output by overlapping the two rasters. Z differences can be calculated by subtracting the values of the overlapping pixels.

6.2.2 Classified point cloud vs ground truth data

As described in Section 5.1.1, the water boards and Deltares that captured and provided me the used green LIDAR dataset, they collected few ground truth data (i.e. GPS measurements) for the specific regions. These data are going to be used in order to validate the classified outputs of both methods.

For the validation process, the water body (NL1_151) is used. There are four measured profile section in this dataset, as seen in Figure 5.5. Many bottom points are measured along each profile line. All the profile sections are displayed as line shapefiles. In order to extract the related points of the point cloud, a buffer of 0.5m was created for each line segment and they are polygons. Then, the water body was cropped based on the new polygons.

The new extracted point cloud parts are going to be validated with the field measurements. In Figure 6.37, the points of the four profile sections are shown and the number of them is presented in Table 6.11. Important to mention that the number of points in this Table 6.11 correspond to all the points, and not to the detected bottom ones from the two methods. As a result, the lowest points of this section are used for the rasterization process and the z differences are computed between them and the ground data.

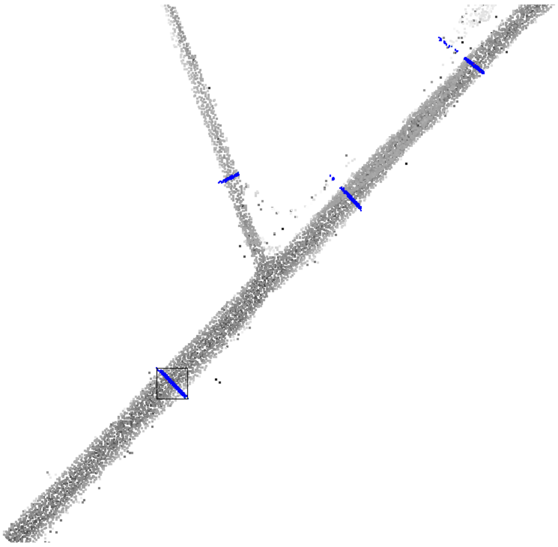


Figure 6.37: Water body (NL4_199): Profile sections with extracted points

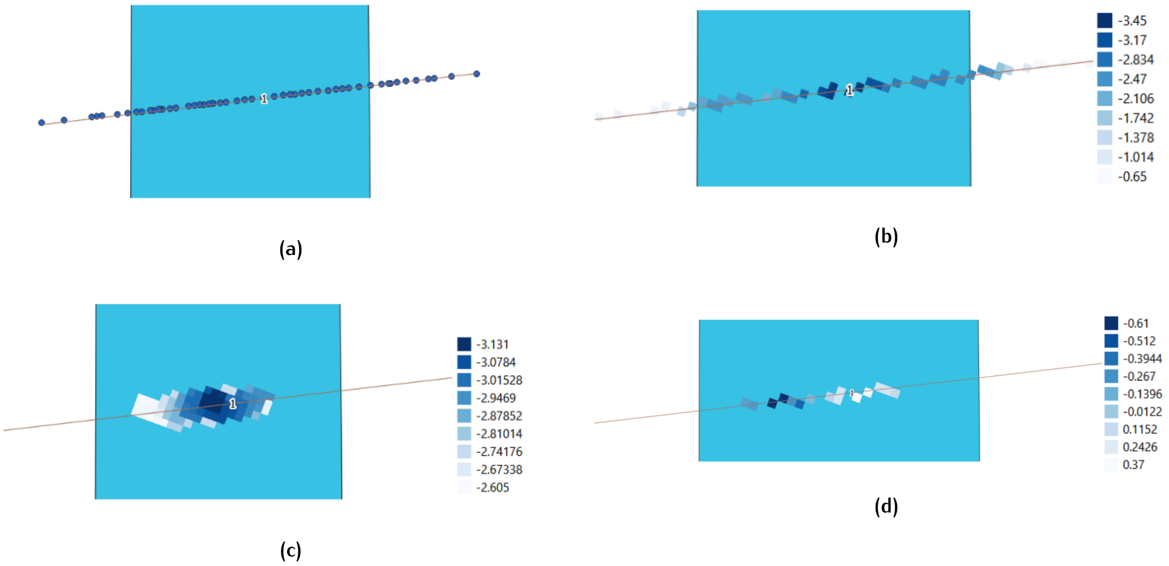
Name	Profile section 1	Profile section 2	Profile section 3	Profile section 4
51NL1	217	683	616	897

Table 6.11: Water body (NL1_151): Number of point cloud points per profile section

Name	Profile section 1	Profile section 2	Profile section 3	Profile section 4
51NL1	53	68	66	47

Table 6.12: Water body (NL1_151): Number of GPS measurements per profile section

In order to compare the two datasets, the ground data are rasterized with pixel size 0.5m and the point cloud section with resolution 0.5m, as well. This procedure has been done for all the profile sections. Following Figures 6.38 present the raster outputs and then their z differences, which were computed by using raster calculator tool.



(a)

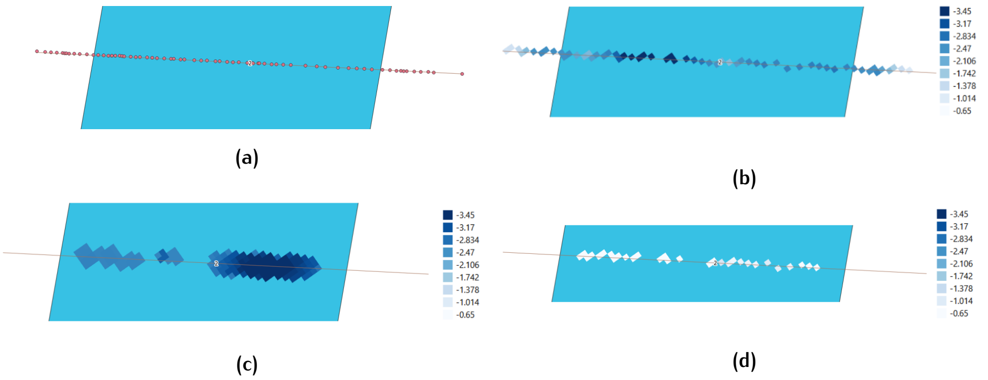
(b)

(c)

(d)

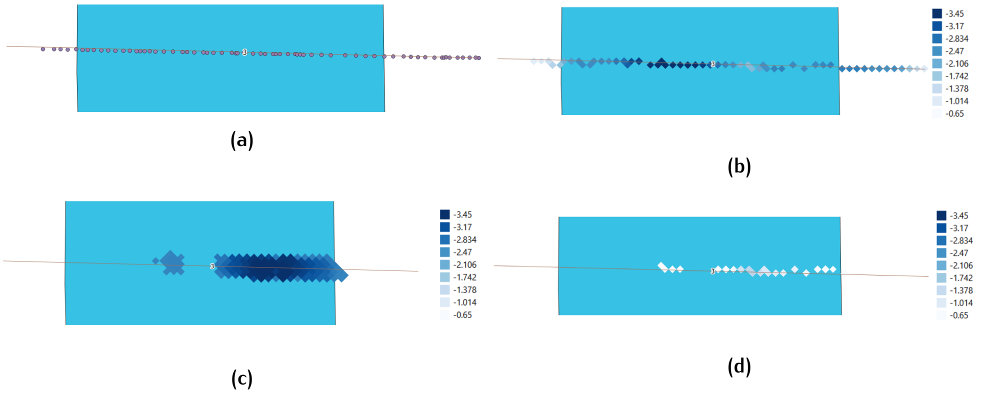
Figure 6.38: Profile section 1 of water body (NL1_151): 6.38a Ground truth data. 6.38b Rasterized ground truth data with 0.5m pixel size. 6.38c Rasterized bottom points 6.38d Computed z differences between the two rasters.

Similarly, the comparison was done for the other three profile sections.



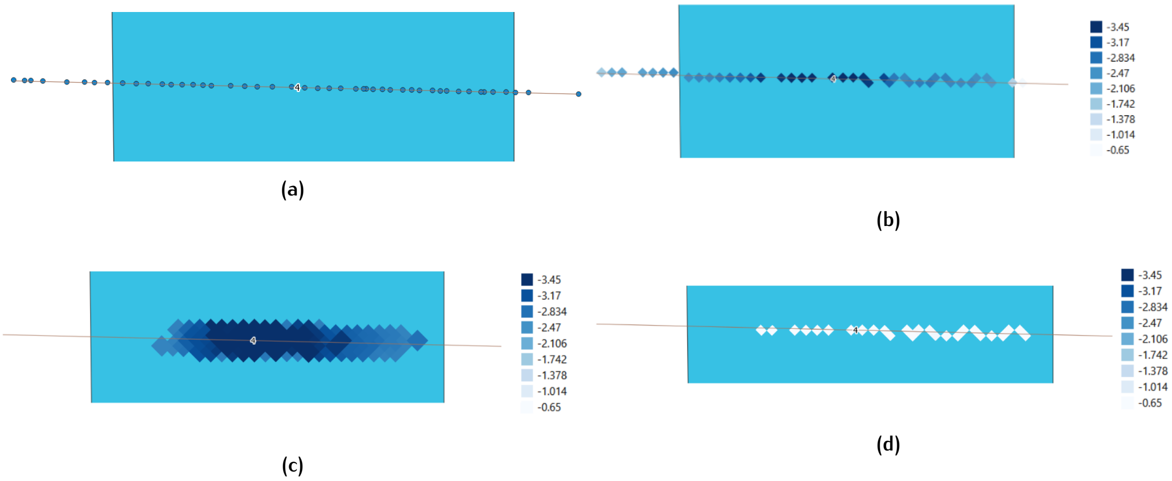
(a) (b) (c) (d)

Figure 6.39: Profile section 2 of water body (NL1_151): 6.39a Ground truth data. 6.39b Rasterized ground truth data with 0.5m pixel size. 6.39c Rasterized bottom points 6.39d Computed z differences between the two rasters.



(a) (b) (c) (d)

Figure 6.40: Profile section 3 of water body (NL1_151): 6.40a Ground truth data. 6.40b Rasterized ground truth data with 0.5m pixel size. 6.40c Rasterized bottom points 6.40d Computed z differences between the two rasters.

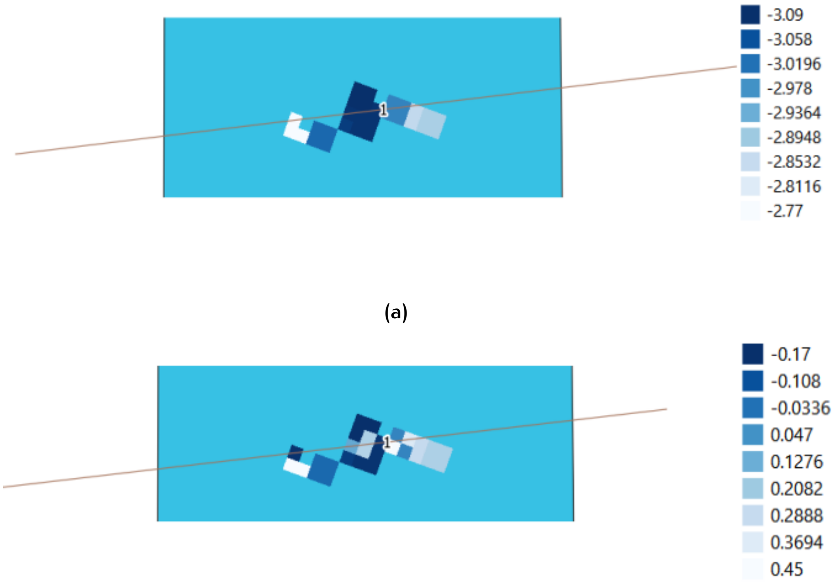


(a) (b) (c) (d)

Figure 6.41: Profile section 4 of water body (NL1_151): 6.41a Ground truth data. 6.41b Rasterized ground truth data with 0.5m pixel size. 6.41c Rasterized bottom points 6.41d Computed z differences between the two rasters.

As seen in Figures 6.39, 6.40 and 6.41, the z differences are almost always below the 0,65m (white pixels), and only a few pixels (light grey) show a difference bigger than 1m.

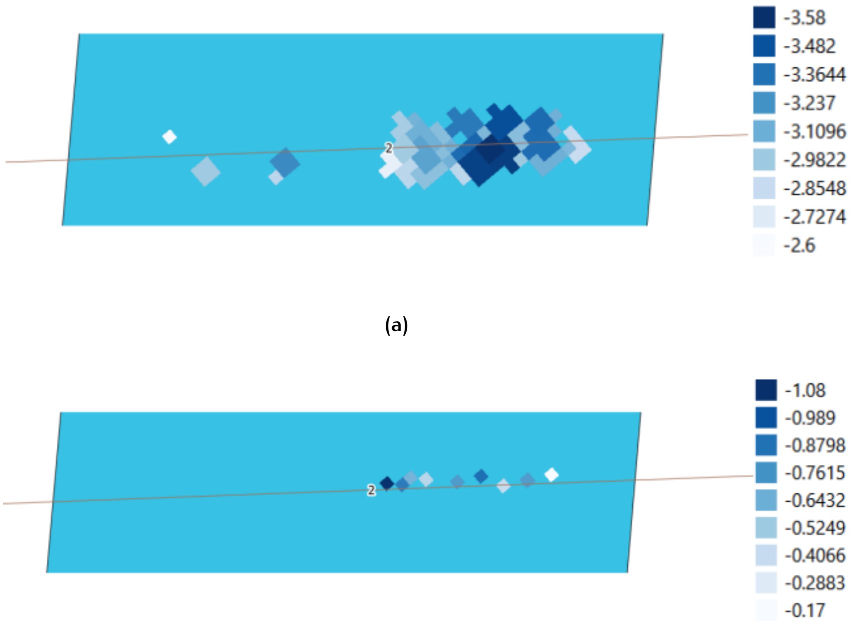
The next step is to apply the same procedure, but using now the detected bottom points of both pulse and voxel based methods. In addition, since the point cloud contains now only the detected bottom points, the profile sections were extracted using 1m buffer zone in order to have bigger number of surrounding points.



(a)

(b)

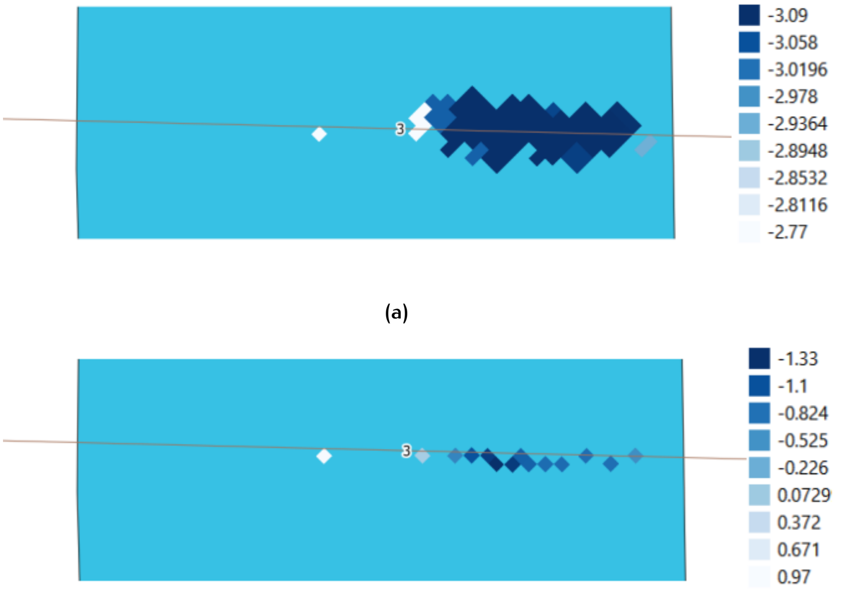
Figure 6.42: Profile section 1 of water body (NL1_151): 6.42a Rasterized detected bottom points of voxel-based method. 6.42b Computed z differences (m) between rasterized ground truth data and Figure 6.42a.



(a)

(b)

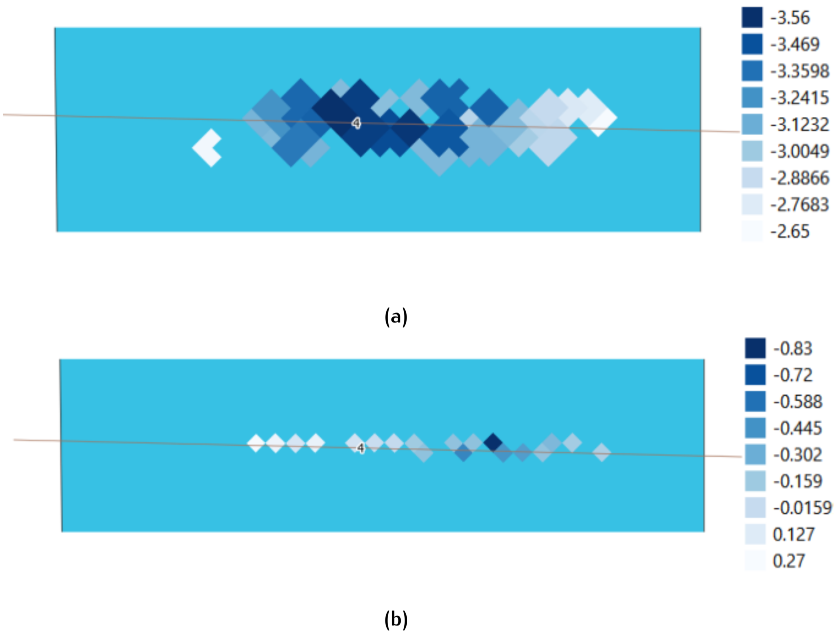
Figure 6.43: Profile section 2 of water body (NL1_151): 6.43a Rasterized detected bottom points of voxel-based method. 6.43b Computed z differences Computed z differences (m) between rasterized ground truth data and Figure 6.43a.



(a)

(b)

Figure 6.44: Profile section 3 of water body (NL1_151): 6.44a Rasterized detected bottom points of voxel-based method. 6.44b Computed z differences (m) between rasterized ground truth data and Figure 6.44a



(a)

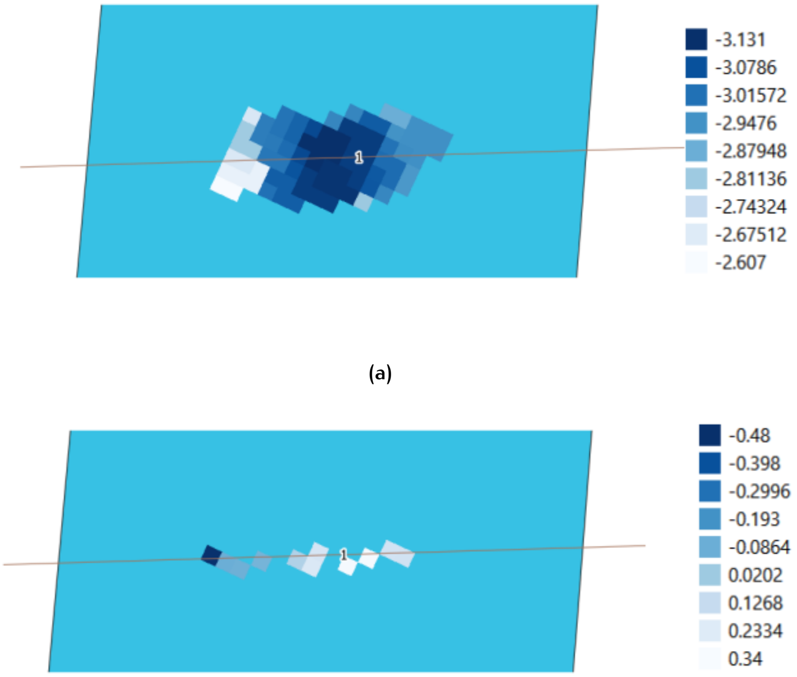
(b)

Figure 6.45: Profile section 4 of water body (NL1_151): 6.45a Rasterized detected bottom points of voxel-based method. 6.45b Computed z differences (m) between rasterized ground truth data and Figure 6.45a

As seen in Figures 6.42, 6.43, 6.44 and 6.45, the detected bottom points using the voxel-based approach rasterized with pixel size (0.5m), and then the z differences with the ground truth data were computed. The detected points do not cover the entire extent of the profile section, and therefore the raster output presents gaps (i.e. missing pixels). Thus, the z differences were computed only for the overlapping pixels of the ground data and bottom points. It is evident that these differences vary from few centimetres (10cm) to maximum 1m. For instance, in case of Figure 6.44b most of the pixels have values (light blue) around 50cm, whereas in Figure 6.45b the majority of them (white shades) ranges from 15cm to 27cm.

Furthermore, the detected bottom points of the pulse-based method compared with the ground truth data. The approach was the same as both data points were rasterized. The Figures 6.46, 6.47, 6.48 and 6.49 show the rasterized last returns (i.e. bottom points) of the pulses. The number of pixels is bigger in the rasterized pulse points as opposed to the voxel points. This makes sense as the original point cloud was grouped per pulses without removing any points. In the voxel approach, only one point per voxel was stored. Thus, the number of points changes when the number of voxels changes, as long as it is related to the voxel size.

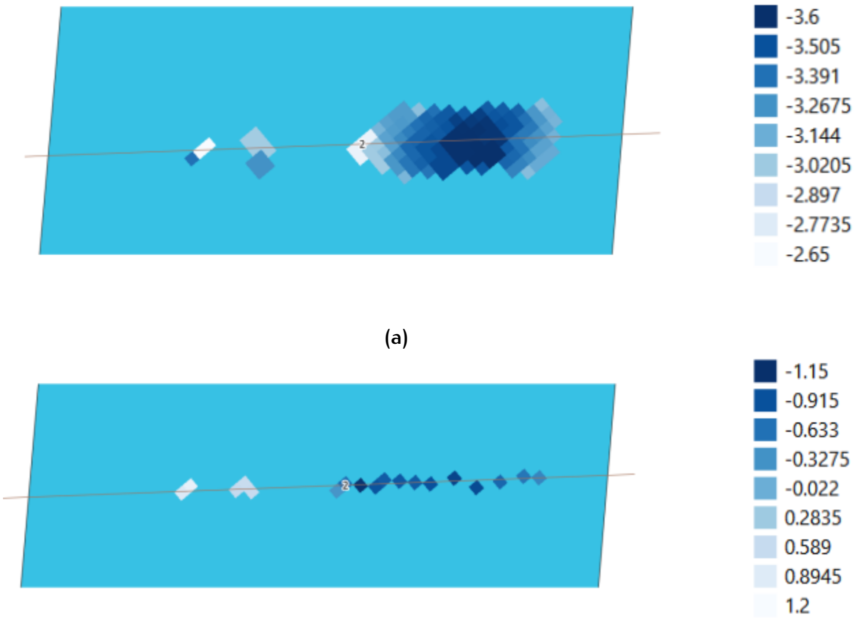
The z differences range from 1.5m to up a few centimetres (0.25m). The number of pixels also differs per profile section. For example, the Figure 6.49b has points almost all across the section line, whereas the Figure 6.46b only in the middled part of the profile. In case of section 3, the results are quite promising as the differences are small (20cm - 40cm) in the majority of pixels. Whereas most pixels have high values (around 60cm) in section 2.



(a)

(b)

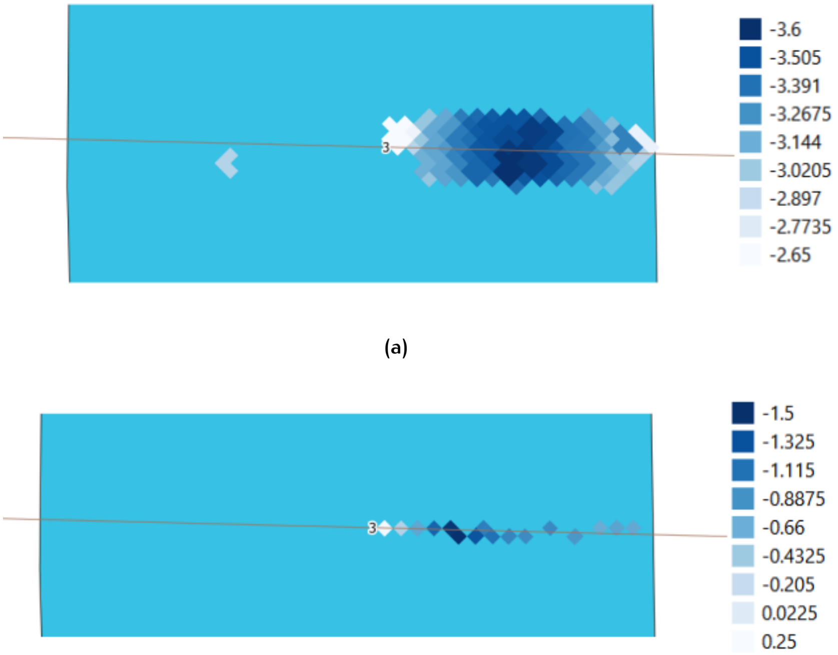
Figure 6.46: Profile section 1 of water body (NL1_151): 6.46a Rasterized detected bottom points of pulse-based method. 6.46b Computed z differences (m) between rasterized ground truth data and Figure 6.46a.



(a)

(b)

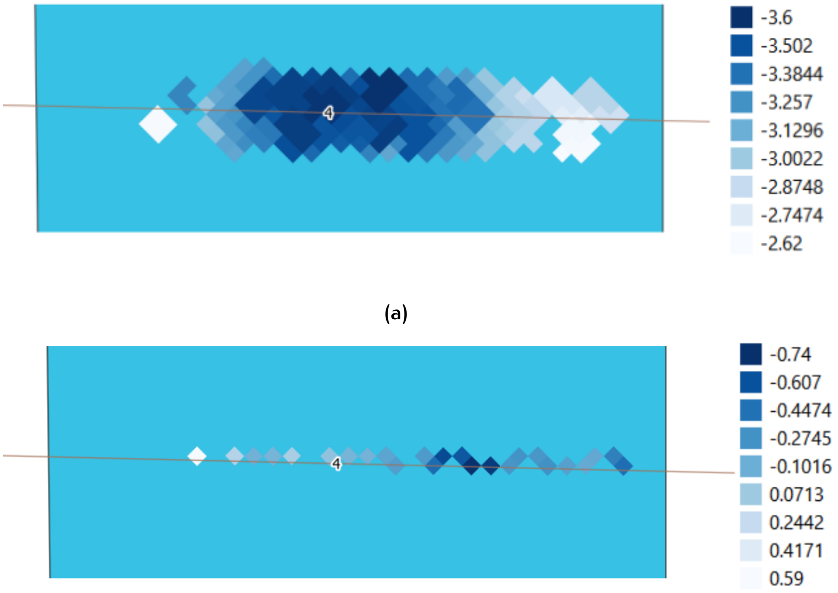
Figure 6.47: Profile section 2 of water body (NL1_151): 6.47a Rasterized detected bottom points of pulse-based method. 6.47b Computed z differences (m) between rasterized ground truth data and Figure 6.47a.



(a)

(b)

Figure 6.48: Profile section 3 of water body (NL1_151): 6.48a Rasterized detected bottom points of pulse-based method. 6.48b Computed z differences (m) between rasterized ground truth data and Figure 6.48a.



(a)

(b)

Figure 6.49: Profile section 4 of water body (NL1_151): 6.49a Rasterized detected bottom points of pulse-based method. 6.49b Computed z differences (m) between rasterized ground truth data and Figure 6.49a.

To sum up, the ground truth data is essential to validate the results of the two proposed methods. This section presents the raster outputs of pulse and voxel-based approach and the comparison with the field measurements. Three comparisons were applied:

1. Lowest points of section vs ground truth data
2. Bottom points (i.e. last returns) of pulse-based method vs ground truth data
3. Bottom points of voxel-based method vs ground truth data

The Table 6.13 presents the amount of data points that used during the rasterization procedure. It is evident that the number of bottom points between pulse and voxel approach differs a lot. This affects both the number and the values of pixels. Points may have lower than 0.5m (pixel size) distance in the pulse approach, and therefore the pixel value will take into account more than one point. In the voxel approach, it is rare as the points extracted with voxel size 1m.

Points	Profile section 1	Profile section 2	Profile section 3	Profile section 4
Section	217	683	616	897
Bottom (pulse)	82	266	234	358
Bottom (voxel)	10	44	55	35

Table 6.13: Water body (NL1_151): amount of points used for the rasterization process.
* For the section points, the lowest points used based on the resolution (0.5m)

Next the z differences of the profile section 3 are presented for all the comparisons. By using a point sampling tool and the exact location of the field points (x,y,z), the z difference values of the three rasters were extracted and presented in Table 6.13. All the points (p4 - p9) in the voxel based method have smaller differences with the ground truth data than with the pulse ones. This is just a small validation example for few points of this profile section. This can be created for the other profile sections, too.

Dataset (51NL1)	Profile section 3					
Points	p4	p5	p6	p7	p8	p9
Section	-	-	-1.09	-	-1.198	-1.478
Pulse - based method	-0.716	-0.941	-1.112	-1.021	-1.198	-1.485
Voxel - based method	-0.637	-0.843	-0.958	-0.958	-1.086	-1.226

Table 6.14: Z Differences between the ground truth data and the pulse and voxel-based methods. The lowest points of section had only three differences

6.3 COMPUTATION TIME AND SCALABILITY

Factors that can affect the effectiveness of the method in terms of computation time and memory allocations are: the *voxel size* and the *dataset size*. This section presents the evaluation of these parameters for different datasets. The experiments have been implemented in a computer with specific characteristics as seen in Section 5.1.2.

6.3.1 Voxel size

The voxel size obviously is a difficult selection and has a major influence in the computation time and the accuracy of the results. The number of voxels increases as the voxel size decreases and the computation time may be unpractical for this case. The high density of points in a voxel; the points inside a voxel, influence the complexity of the process. Thus, a point cloud with sparse density may be processed faster during the voxelization procedure.

Table 6.15 presents the computation time with respect to the different voxel sizes of the water body (NL1_151).

Input point size	Voxel size (m)	Voxels	Time (sec)
511.914	0.5	43.806	1.015
511.914	1	13.048	326
511.914	2	4.435	132
511.914	3	2.493	90
511.914	4	1.640	75

Table 6.15: Water body (NL1_151): Various voxel sizes with the corresponding number of voxels and computation time

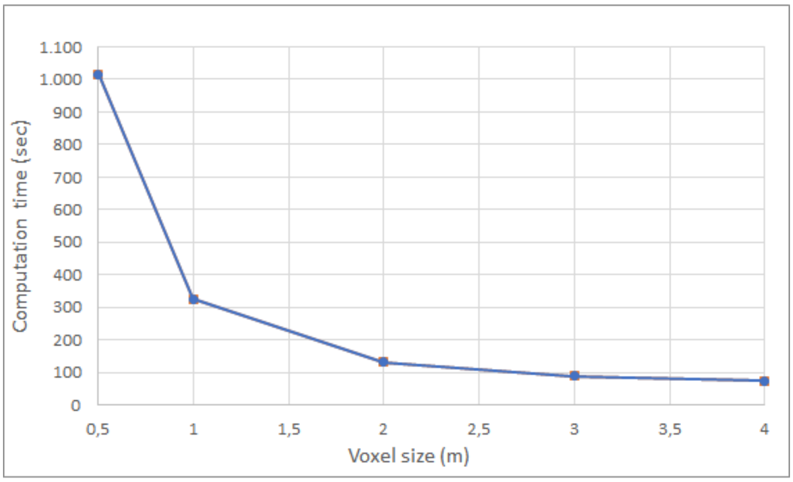


Figure 6.50: Water body (NL1_151): Computation time per spatial resolution

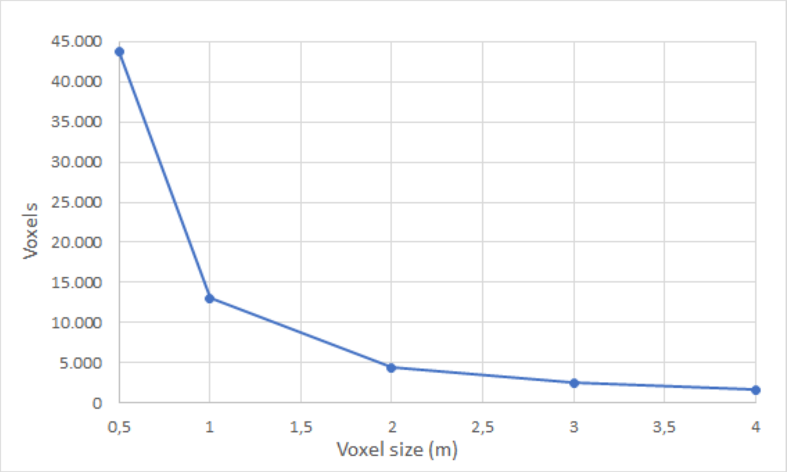


Figure 6.51: Water body (NL1_151): Number of voxels with respect to various voxel sizes

The computation time is directly affected by the spatial resolution (voxel size) during the voxelization procedure. When the voxel is less than 1m (0.5m), the computation time increases rapidly and is tripled (1015 sec) compared to voxel size 1m (326 sec) (see Figure 6.50). The time is also related to the amount of voxels in the dataset. When the voxel size decreased to 0.5m, the number of voxels increased to almost 44.000. However, it was around 13.000 when the voxel size was 1m.

6.3.2 Dataset size

The size of a point cloud affects the computation time, too. A large point cloud definitely needs more processing during the voxelization and pulse-based procedure. Computation time mainly in voxelization grows rapidly as the number of voxels increases. Table 6.15 presents how the computation time for a dataset with around 511.000 points changes based on the voxel size. For instance, the shape of bounding box is rectangular and all the points are allocated in its diagonal. Then, the voxels at the corners of the bounding box will be empty.

7 CONCLUSION AND FUTURE WORK

This chapter presents the conclusion of this study. First the research questions are answered here in Section 7.2. Secondly, the research contribution are given in Section 7.3. Lastly, the future work are summarized in Section 7.4.

7.1 CONCLUSION

This study described a proposed workflow to classify the shallow and muddy water bodies in the Netherlands using green LIDAR data. Two methods, pulse and voxel-based, are implemented to classify the water bodies and especially distinguish their water beds. Also, ground truth data (i.e. GPS measurements) of their bottom surface have been collected and they have been used to validate the classified outputs of the two methods. It showed that the voxel-based approach resulted to less bottom points than the pulse one as it is influenced by the selection of a voxel size. However, the validation results were better using the voxel approach. The z differences between the ground truth data and the detected bottom points of the voxel method were smaller (20-40cm) than of the pulse method (>60cm). Furthermore, the voxelization of the source point cloud requires a careful selection of the voxel size to achieved satisfying results.

It is important to mention that even if the waterbed detection achieved with promising results compared to the ground truth data, some issues need attention. The major issue on the voxel-based method as implemented in this study, is the selection of the voxel size. It is a trade-off between the computation time and the number of voxels. Small voxel size (e.g. 0,5m) may result to more points per voxel, if the point cloud is dense, and bigger number of voxels. Definitely, the computation time sharply increases (see Section 6.3.1) and may be unpractical for the entire process. A bigger voxel size (e.g. 1m) results in less voxels and might less points per voxel. Therefore, the amount of detected bottom points will be less since one points extracts per voxel.

In addition, using the pulse-based method, more potential bottom points were detected as the procedure was based solely on their characteristics and not on their spatial distribution. More points may wrongly be classified as waterbed due to the presence of errors in point's attributes such as RN and NR. These errors can affect the grouping per pulse procedure. This can be the reason why the z differences between the ground truth data and pulse-based detected bottoms are larger in most cases.

Potential bottom areas can be detected using both methods. Nonetheless, this thesis showed the benefits of the voxel-based method compared to the pulse one in terms of accuracy in the detection.

7.2 RESEARCH QUESTIONS

This section answers the research questions stated in Section 1.2, starting with the sub-questions, and ending with the main research question.

Question 2. *Can pulse and/or neighbourhood based methods - in a green airborne LIDAR - be used to classify and detect the bottom points?*

Both the pulse and voxel-based methods manage to classify green LIDAR point clouds and especially detect the bottom parts.

For the pulse-based method, the point cloud is classified into three classes: water surface, underwater and bottom points, unclassified. The points are grouped per pulses using their NR, RN and the GPS time attributes. The first return of a pulse is classified as a water surface point, as it is the first point that the laser beam hit during its transmission from the air to water interface. The middle returns correspond to the under water surface points, while the last returns considered as bottom point. Thus, a classification code is assigned for each point based on its order in a pulse. In addition, the class with all the unclassified points contains the left overs from pulses that have been cut in the pre-processing procedure and may not be classified in the other classes.

For the voxel-based method, the point cloud is divided into 3D voxels (i.e. water columns). The x,y dimensions of a water column are the same (e.g. 1m x 1m) and the z dimension matches with the z extent of the point cloud. For each voxel, a histogram is created with fixed size bins. The highest bin corresponds to the water surface while the second highest to the bottom surface. Then, the lowest point of all the points of the second bin is extracted and assumed as potential bottom point. Based on three parameters: density, distance and intensity the confidence value for each bottom point are calculated. Thus, all the bottom points can have a value that estimates how confident they are to be at the bottom. This method takes account the spatial distribution of points and not their descriptive characteristics.

Question 3. *What is the influence of different voxel resolutions for classification, in terms of accuracy and computation load?*

The voxel size has a major influence on the accuracy of the proposed voxel-based method. A size larger than 1m results to a small number of voxels to capture the details in the bottom surface (see Figure 6.51). Even though more points may fall into a voxel, its coverage area (x,y) increases, but again only one point is extracted - as bottom point - from its entire area. Thus, the detected bottom points will certainly be less in number and more sparse.

On the other hand, a resolution smaller than 1m (e.g. 0.5m) results to a big number of voxels as seen in Figure 6.51. The number of voxels has been sharply increased. Every voxel might contain less points than before, but more empty voxels will exist as its extent shrinks.

Moreover, the voxel size directly affects the computation time. A too fine resolution (e.g. < 0.5m) increases a lot the running time as seen in Figure 6.50. This means that using a small voxel size and dealing with a large dataset (e.g. 500.000 points), the processing time reaches unpractical levels even if the number of detected bottom points can increase. Also, the existence of sparse and dense parts in a point cloud directly influences the running times. The use of 1m voxel size seems satisfied for the tested dataset to represent the bottom surface.

Question 4. *How does the various point cloud quality (i.e. density, outliers) affect the classification process?*

The point cloud quality in terms of density and outliers play a vital role in the classification process of both proposed methods. During the pulse based-method, especially in the grouping procedure, many points remained ”unclassified”. This is related to the existence of outliers in the water body even if few pre-processing steps are applied. In particular, the original green LIDAR clipped in x,y dimensions using an external dataset (Top10NL) with the water boundaries. In the z level, it has been cropped using hard-coded threshold values, after a visual inspection of the dataset. This is a certain reason why outliers exist, especially near the water surface, in a water body.

However, as stated in Section 1.3, the scope of this research is to use water body point clouds and not to extract them automatically from an unclassified green dataset. Therefore, there are expected to be some outliers that may influence the classification process. This is a limitation of the process that needs to be taken into account.

During the voxel-based method, the potential presence of outliers below the bottom of a water body can not affect the detection of bottom points. Even if there are a few outliers a bit lower (e.g. 10cm away) from the potential water bed of the water body, they will not affect the procedure. The created histogram of a voxel contains bins with certain width size (5cm). Therefore, the outliers will not be so many as to create a peak in the histogram and be interpreted as a bottom area.

The classification procedure in both methods can be influenced by the point cloud density. A denser point cloud affects the computation time of the grouping per pulse process and the voxelization (see Table 6.15). A sparse point cloud does not cost computation time, but it could result into less accurate output. In this study, the processed water body datasets presented an acceptable number of points per square meter (see Table 6.1) and did not influence negatively the classification process.

Question 5. *Can a confidence value of water points be calculated?If it is possible, how?*

Yes, a confidence value was calculated and assigned for every detected bottom point during the voxelization procedure. The confidence value is defined based on three parameters: the density, the distance and the intensity (see also Section 6.1.2). Since a histogram per voxel has been created and the second highest bin (i.e. bin of interest) has been also detected, the lowest point from all the points of that bin is extracted. In order to be known how confident or not is a bottom point, the three parameters are calculated.

The density of points in the bin of interest are measured and normalized with respect to the maximum density of the corresponding bin in all the other voxels. The normalized density is value then assigned to the point (*NormDensity*). Similarly, the intensity value of that point normalized (*NormIntensity*). Also, the distance is the vertical distance between the highest point of the bin of interest (water surface) and the lowest point of the second highest bin. The distance per voxel has been normalized (*NormDistance*) with the maximum one.

In order to define the various confidence classes, the median of the NormDensity, the NormIntensity and the mean of the NormDistance were used for each water body dataset. The median can give a more generic overview of these two values in the entire extent of a dataset. Instead of using hard-coded values that may be useful only for a certain dataset, the median is used for the definition of confidence classes for all the datasets.

And finally, the main research question is answered:

Question 1. *Can the bottom points of shallow and muddy water-bodies in the Netherlands be automatically detected using ALB?*

Yes, the two proposed methods managed to automatically classify the water beds of the Dutch shallow and muddy water-bodies. However, based on the quality metrics presented in Table 6.14, the voxel-based method performs well in terms of accuracy with the ground truth data. The comparisons between the pulse-based/ground truth data and the voxel-based/ground truth data showed that the voxel-based is more accurate to the original GPS measurements. However, the amount of detected bottom points is directly affected by the selection of the voxel size and the density of points in a voxel. Thus, less but more accurate points represent the bottom surface than in the pulse-based, where more bottom points exist.

7.3 RESEARCH CONTRIBUTION

The main contributions of this thesis are:

- The design of a detailed workflow from the pre-processing until the final classification results of an unclassified ALB dataset.
- The implementation of two methods to classify a green LIDAR dataset using either the LIDAR points' characteristics or their spatial distribution. This study explores the performance of both methods.
- The comparison of the pulse and voxel-based methods in terms of accuracy. Quality metrics are presented with the provided ground truth measurements.
- This study shows that the voxel-based method can provide more accurate bottom points, while the pulse-based approach contains bigger number of bottom points.

7.4 FUTURE WORK

7.4.1 Point cloud classification with machine learning

This section gives a few recommendations that may be helpful for this research topic.

1. **Deep Learning on point cloud** The PointNet++ algorithm uses a deep net architecture that manipulates a set of points without applying voxelization. It learns from both global and local point features, and they can be used for object classification and segmentation purposes. Starting from the entire point cloud, the points are grouped into some clusters and then compressed into a single point that carries this information. This procedure continues by taking the new points and grouping them into new clusters. Then, the process reverses and tries to build the original point cloud (Singer, 2019). The possibilities of this algorithm for the classification of water body datasets can be explored. Especially, for the detection of the bottom points based on their distances from the water surface.
2. **Different point cloud densities** To evaluate the limits of the pulse and voxel based methods, the implementations should be tested with other green LIDAR datasets of different densities. I assume that both methods could also work for less denser datasets, even though I tested them with quite dense ones. Also, using a denser dataset, more bottom points may be detected using the voxel-based method. For example, datasets from other regions that have shallower and more muddy waters can be a big challenge for the algorithms.
3. **Pre-processing automation** The unclassified green LIDAR contains also topographic features (e.g. buildings) from which only the water bodies must be extracted. This is an essential pre-processing step that relies on the use of procedures in LAStools and QGIS. This was an easy and required step as it saved a lot of development time to choose the water bodies. A further detailed description of the step-by-step pre-processing procedures may help the optimization.

4. **Ground filtering** Instead of cropping the point cloud in the z level using some hard coded range values, the ground filtering approach can be used. Thus, during the voxelization procedure, the points of every voxel could be classified into ground and non-ground points. Ground points are those points that are part of the water body, while non-ground points are vegetation (e.g. bushes) or structures (e.g. buildings) near the water body.

BIBLIOGRAPHY

AeroData (2015). Pilotproject: Meten ondiepe sloten in de polder groot wilnis vinkeveen met laser bathymetry. Technical report, Waternet, Utrecht.

Allouis, T., Bailly, J.-S., and Feurer, D. (2015). Assessing water surface effects on LiDAR bathymetry measurements in very shallow rivers: A theoretical study. page 9.

Allouis, T., Bailly, J.-S., Pastol, Y., and Le Roux, C. (2010). Comparison of LiDAR waveform processing methods for very shallow water bathymetry using Raman, near-infrared and green signals. *Earth Surface Processes and Landforms*, pages n/a–n/a.

Andersen, M. S., Gergely, Á., Al-Hamdani, Z., Steinbacher, F., Larsen, L. R., and Ernstsen, V. B. (2017). Processing and performance of topobathymetric lidar data for geomorphometric and morphological classification in a high-energy tidal environment. *Hydrology and Earth System Sciences*, 21(1):43–63. ZSCC: 0000007.

ArcMap (2016). What is a tin surface?

ASPRS (2013). LAS SPECIFICATION VERSION 1.4 - R13. Technical report, ASPRS, Maryland.

Boerner, R., Hoegner, L., and Stilla, U. (2017). Voxel based segmentation of large airborne topobathymetric lidar data. page 8.

Corti Meneses, N., Baier, S., Geist, J., and Schneider, T. (2017). Evaluation of green-lidar data for mapping extent, density and height of aquatic reed beds at lake chiemsee, bavaria—germany. page 17.

Ferraz, A., Bretar, F., Jacquemoud, S., and Gonçalves, G. R. (2009). The role of lidar systems in fuel mapping. page 43.

Guenther, G., Cunningham, G., LaRocqu, P., and Reid, D. (2000). Meeting the accuracy challenge in airborne bathymetry. page 29.

Guiotte, F., Lefevre, S., and Corpetti, T. (2019). Rasterization strategies for airborne lidar classification using attribute profiles. page 4.

Habel, H., Balazs, A., and Myllymaki, M. (2018). Spatial analysis of airborne laser scanning point clouds for predicting forest variables. page 23.

Hilldale, R. C. and Raff, D. (2008). Assessing the ability of airborne LiDAR to map river bathymetry. *Earth Surface Processes and Landforms*, 33(5):773–783.

IQmulus (2019). Lidar full waveform coastal feature extraction.

Isenburg, M. (2019). Completeness and correctness of discrete lidar returns per laser pulse fired.

J. Sumnall, M., A. Hill, R., and A. Hinsley, S. (2015). Comparison of small-footprint discrete return and full waveform airborne lidar data for estimating multiple forest variables. page 10.

Kinzel, P. J., Legleiter, C. J., and Nelson, J. M. (2013). Mapping River Bathymetry With a Small Footprint Green LiDAR: Applications and Challenges. *JAWRA Journal of the American Water Resources Association*, 49(1):183–204. ZSCC: NoCitationData[s0].

Koma, Z. (2017). Lidar data handling techniques.

Ledoux, H., Ohori, K. A., and Peters, R. (2019). Computational modelling of terrains. page 144.

LiDARNews (2018). Airborne lidar – discrete return or full waveform?

Lohani, B. (2010). Lidar error sources.

Mahphood, A. and Arefi, H. (2019). Virtual first and last pulse method for building detection from dense lidar point clouds. page 27.

Mandlburger, G., Hauer, C., Wieser, M., and Pfeifer, N. (2015). Topo-Bathymetric LiDAR for Monitoring River Morphodynamics and Instream Habitats—A Case Study at the Pielach River. *Remote Sensing*, 7(5):6160–6195.

Meng, X., Nate, C., and Zhao, K. (2010). Ground filtering algorithms for airborne lidar data: A review of critical issues. page 28.

Nourian, P., Gonc¸alves, R., Zlatanova, S., Arroyo Ohori, K., and Vu Vo, A. (2016). Voxelization algorithms for geospatial applications computational methods for voxelating spatial datasets of 3d city models containing 3d surface, curve and point data models. page 18.

Parrish, C. E., Magruder, L. A., Neuenschwander, A. L., Forfinski-Sarkozi, N., Alonzo, M., and Jasinski, M. (2019). Validation of ICESat-2 ATLAS Bathymetry and Analysis of ATLAS's Bathymetric Mapping Performance. *Remote Sensing*, 11(14):1634.

Plaza-Leiva, V., Antonio Gomez-Ruiz, J., Mandow, A., and García-Cerezo, A. (2017). Voxel-based neighborhood for spatial shape pattern classification of lidar point clouds with supervised learning. page 17.

RIEGL (2016). RIEGL_VQ880G.

Sharma, B. (2019). What is lidar technology and how does it work?

Sharma, B. (2020). What is lidar?

Singer, Z. (2019). Understanding machine learning on point clouds through pointnet++.

Ussyshkin, V. and Theriault, L. (2011). Airborne lidar: Advances in discrete return technology for 3d vegetation mapping. page 19.

van Tol, L. (2019). *Groene LiDAR Voor de Ondiepe Nederlandse Binnenwateren*. PhD thesis, Maritiem Instituut Willem Barentsz.

van Tol, L. (2019). Groene lidar voor de ondiepe nederlandse binnenwateren.

Vazquez, D. B. (2017a). Bathymetric profiling using a yellow wavelength and a time-of-flight approach based on Single-Photon Counting. A laboratory study for Dutch shallow inland waters. page 1.

Vazquez, D. B. (2017b). Better Report-Bathymetric profiling using a yellow wavelength and a time-of-flight approach based on Single-Photon Counting. Technical report, University of Amsterdam, Amsterdam.

Vázquez, D. B., Smits, F. J., and ter Hennepe, E. (2017). *Multi-Wavelength Bathymetric Profiling of Shallow Inland Water Bodies by Using Supercontinuum Lasers, a Wavelength of 590 Nm, Extremely Short Pulses and Single-Photon Counting*. PhD thesis.

Wang, L., Xu, Y., Li, Y., and Zhao, Y. (2018). Voxel segmentation-based 3D building detection algorithm for airborne LIDAR data. *PLOS ONE*, 13(12):e0208996.

Wasser, L. A. (2020). The basics of lidar - light detection and ranging - remote sensing.

Zhao, J., Zhao, X., Zhang, H., and Zhou, F. (2017). Shallow Water Measurements Using a Single Green Laser Corrected by Building a Near Water Surface Penetration Model. *Remote Sensing*, 9(5):426.

A REPRODUCIBILITY SELF-ASSESSMENT

A.1 MARKS FOR EACH OF THE CRITERIA

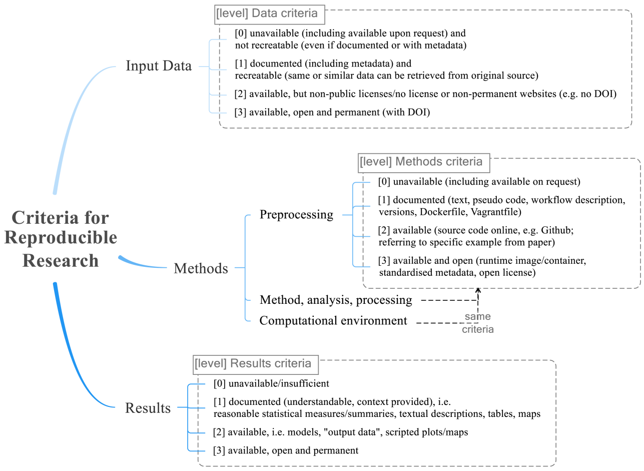

Criteria for Reproducible Research
Input Data
[level] Data criteria
[0] unavailable (including available upon request) and not recreatable (even if documented or with metadata)
[1] documented (including metadata) and recreatable (same or similar data can be retrieved from original source)
[2] available, but non-public licenses/no license or non-permanent websites (e.g. no DOI)
[3] available, open and permanent (with DOI)
Methods
Preprocessing
[level] Methods criteria
[0] unavailable (including available on request)
[1] documented (text, pseudo code, workflow description, versions, Dockerfile, Vagrantfile)
[2] available (source code online, e.g. Github; referring to specific example from paper)
[3] available and open (runtime image/container, standardised metadata, open license)
Method, analysis, processing
Computational environment
same criteria
Results
[level] Results criteria
[0] unavailable/insufficient
[1] documented (understandable, context provided), i.e. reasonable statistical measures/summaries, textual descriptions, tables, maps
[2] available, i.e. models, "output data", scripted plots/maps
[3] available, open and permanent

Figure A.1: Reproducibility criteria to be assessed.

Criteria for reproducible research:

- Input Data: 2

- Pre-processing: 1

- Methods: 2

- Computational environment: 3

- Results: 2

A.2 SELF-REFLECTION

The processing and experiments of this study were performed in collaboration with Deltares; an independent institute for applied research in the field of water and subsurface. All the information related to the input data can not be publicly available. In particular,

- **Input Data**: During this thesis, airborne bathymetric LIDAR datasets. The datasets are provided by Deltares and correspond to six different regions in the Netherlands. Several water boards had organized a pilot project in cooperation with Deltares [1], Stowa [2] and Waternet [3]. These datasets are an internal part of Deltares and are not publicly available.

[1] https://www.deltares.nl/en/
[2] https://www.stowa.nl/english
[3] https://www.waternet.nl

- **Pre-processing**: datasets and all the pre-processing procedures are documented with text and workflow descriptions. Most of the procedures were performed through open source toolboxes and software such as LAStools, CloudCompare, Displaz.

- **Methods**: the source code of the classification methods (i.e. pulse and voxel-based) used in this research is available online in GitHub. The algorithms are documented with text, pseudo codes and workflow descriptions.

- **Computational environment**: this study is conducted by using publicly available Julia and Python libraries to process, manipulate and analyse the provided LIDAR datasets.

- **Results**: are documented through text, graphs, plots, but they may also be available by running the freely available source code.

COLOPHON

This document was typeset using LaTeX. The document layout was generated using the arsclassica package by Lorenzo Pantieri, which is an adaption of the original classicthesis package from André Miede.