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

Vasileios Alexandridis

Student ID: 4897803

Supervised by:

Ravi Peters
Jantien Stoter
Maarten Pronk (Deltares)

April 9, 2020

Contents

List of Figures

List of Tables

1 Preface

This document forms a thesis proposal submitted to the Delft University of Technology based on the requirements for the degree of Master of Science (MSc) in Geomatics for the Built Environment.

This proposal will introduce the reader to the defined topic and its relevance with the geomatics field in Section 2. In Section 3, the relevant work is presented and linked with the topic, while the research objectives are declared in Section 4. The methodology and the used tool and datasets are presented in Section 5 and 8 respectively. The time planning of the entire project is given in Section 7.

2 Introduction

The Netherlands, a 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). 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 (Vazquez, 2017), bathymetric information is vital for 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 areas (Vázquez et al., 2017).

However, since the airborne LiDAR bathymetry (ALB) technology has been successfully used in recent decades in deep waters and clear coastal wetted areas, it can be used to improve the bathymetric surveys in the muddy and shallow inland Dutch water bodies. That's why several water boards in collaboration with private companies have already run a pilot project to examine the potential of bathymetric LiDAR, especially green, of monitoring the water depths efficiently, easier and cheaper than manual measurements (i.e. GPS measurements). Especially, green bathymetric LiDAR 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 and 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 be used in order to distinguish the 3D geometry; especially 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). Moreover, 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.

This research aims to implement algorithms that can automatically detect the bottom points of these shallow and muddy water bodies and, then classify them into water surface, underwater and bottom points. The classification process needs to be achieved by performing a high level of automation. The existence of various dense and sparse parts in the point cloud, the muddy and shallow waters, the characteristics and limitations of the data are challenging parameters that certainly affect so the automation process as the accuracy of the results.

Previous studies have been conducted to detect waterbeds in water bodies in the case of Netherlands using bathymetric LiDAR data. The developed methods have not succeeded in detecting bottom points with high certainty and accuracy. Therefore, further research needs to be carried out by applying other techniques and algorithms that could improve the detection process and deal particularly with shallow and muddy water bodies. This research needs to fill this gap by implementing algorithms that enhance the ability to classify the water bodies and then detect the interested bottom points.

The study will be carried out in collaboration with Deltares, a technological institute for applied research in the field of water, soil and infrastructure. Many different companies and organizations (e.g. Water boards, Rijkswaterstaat) benefit from accurate and dense bathymetric LiDAR data, especially with green wavelength for shallow water areas. Deltares in collaboration with other institutes and organizations flew aeroplanes to collect those data from 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.

3 Related work

Several studies have been done in the field of mapping river and shallow water body bathymetry using green LiDAR. Green LiDAR appears as an interesting tool that widely used for coastal surveys. Many studies that focus on water depth measurement quality and limits, in particular for rivers and surface water, 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 such as near-infrared (NIR) and then methods are applied to classify the water points. However, few of them only deal with the detection of the water regions from a point cloud, which 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. Also, other studies point out the various environmental factors such as the water clarity, water turbidity, vegetation and refer to them as important limitations for the accuracy and quality of the LiDAR data.

3.1 Mapping water body 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 combining with additional LiDAR data (e.g. NIR).

3.1.1 Using only green LiDAR

Many studies have been developed which analyse the 3D geometry of water bodies and then detect the water surface and bottom parts using 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, the GLFW modelling, the bathymetry estimation, the determination of limit H_{inf} and 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.

Mandlburger et al. (2015) proposed a method to determine the water surface using only the reflections available from green LiDAR and classify the water echoes. The classification of the water points into (i) surface, (ii) body and (iii) bottom points relied on full waveform features and spatial features based on local neighbourhood. To avoid misclassification of water points, three thresholds were used: water depth, object reflectance and neighbourhood definition. The water depth threshold is related to the maximum penetration depth based on Secchi depth of the water body. The reflectance threshold depends on environmental conditions and the neighbourhood for the dimension of the search (e.g. spheres, cylinders). As seen in Figure 1, 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.
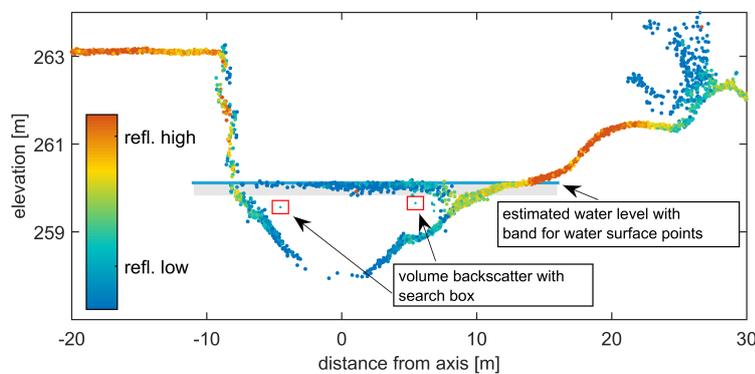
Figure 1: Classification of water echoes based on reflectance, distance form water surface and spatial distribution (Mandlburger et al., 2015)

Andersen et al. (2017) performed a methodology to process green LiDAR data in tidal environments and to improve the classification water surface and bottom points. 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. Then, the refraction correction on the processed points is applied to the detected underwater and bottom points.

Kinzel et al. (2013) considered the importance of green bathymetric LiDAR due to their water-penetrating abilities. The green laser is less attenuated than near-infrared laser, and thus is capable of greater penetration into the water bodies. Specifically, Kinzel et al. (2013) proposed algorithms that used to detect the location of a water surface and bottom return in waveforms for shallow depths (<1 m) and deeper depths. In particular, after correcting the waveform from the refraction factor or water column effects, the algorithm searches for the most significant peak (maximum peak (MP)) which corresponds to the water surface. 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 selects the last peak based on a threshold value. This reduce 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

Using only green LiDAR (532 nm) 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). For depths lower than 2 meters, the difficulty is in discriminating between the two mixed peaks. For that reason, Allouis et al. (2010) proposed the use of near-infrared (NIR) wavelength (1064 nm) and the red wavelength Raman signal (647 nm). The first one is reflected on the water surface, and thus it is easy to distinguish dry land from water surface. The second one 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, NIR and Raman signals help to accurately measure the water surface position and water column, respectively. Also, if depth measurements are missing in muddy shallow waters in green waveforms, Raman signal can be used instead.

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 extras weight to the ALB system. That's why only the green Li-DAR 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 applying statistical analysis. However, the results of this method are affected by the given reference water surface height data (IR data) and water turbidity.

3.2 Detecting water regions from LiDAR data

There are relatively few authors who try to detect and extract water regions from LiDAR data. For instance, (Brzank and Heipke, 2007) study is concentrated on the detection of water regions from LiDAR data using fuzzy logic concept. The raw data are grouped into scan lines. Using training data for the water and muddy areas, the intensity, point density and height of the points were analysed. Individuals weights are calculated using the importance level of these features. Then, the fuzzy logic concept are used to distinguish all the points into water and muddy points. Moreover, even if this study did not aim to distinguish the bottom points of a selected water area, the proposed methodology gave useful insight how fuzzy logic could be used in the classification process.

3.3 Identifying environmental factors that affect laser pulse's transmission

Many ALB systems use green LiDAR as it is suitable for the waterbed detection of water bodies. Thus, 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, 2017).

- **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, 2017).

- **Vegetation** affects negatively the ALB measurements because they can block the laser pulse to reach the bottom sediments.

4 Research objectives

4.1 Research question

As argued in the introduction and the related work sections, the main goal of this thesis is to automatically detect the waterbeds of shallow muddy water-bodies in the Netherlands using airborne green LiDAR datasets. Thus, to classify the water points into three classes: water surface, underwater and bottom points. Therefore, the corresponding main research question is:

- *To what extent can the bottom points of shallow muddy water-bodies in the Netherlands be automatically detected?*

In order to achieve the goal of this thesis, the following sub-questions will be helpful and relevant:

- What is a shallow water-body?

- Which are the characteristics and limitations of a green LiDAR dataset?

- How can the water points of shallow water-bodies into water-surface, underwater and bottom points be classified using bottom point detection (BPD) and voxelization methods?

- How does the various point cloud quality (i.e.density, errors) affect the classification process?

- How can the 3D geometry (DTM surface) of the water-bodies be constructed from the classified point cloud?

4.2 Research scope

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 cropping them in x and y dimensions 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).

Moreover, the main focus of this study is to automatically distinguish the bottoms of the shallow water-bodies by applying different approaches. By detecting those interested bottom points, the original water points can be classified into three different 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 point neighbourhood based methods will be implemented. The later ones take advantage of specific spatial search criteria that tested on the local neighbourhood of each 3D point. Two methods are going to be performed: the boundary point detection (BPD) and the voxelization. All these techniques will be extensively discussed in the methodology section (see Section 5).

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 (see Subsection 5.2.6).

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 (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.
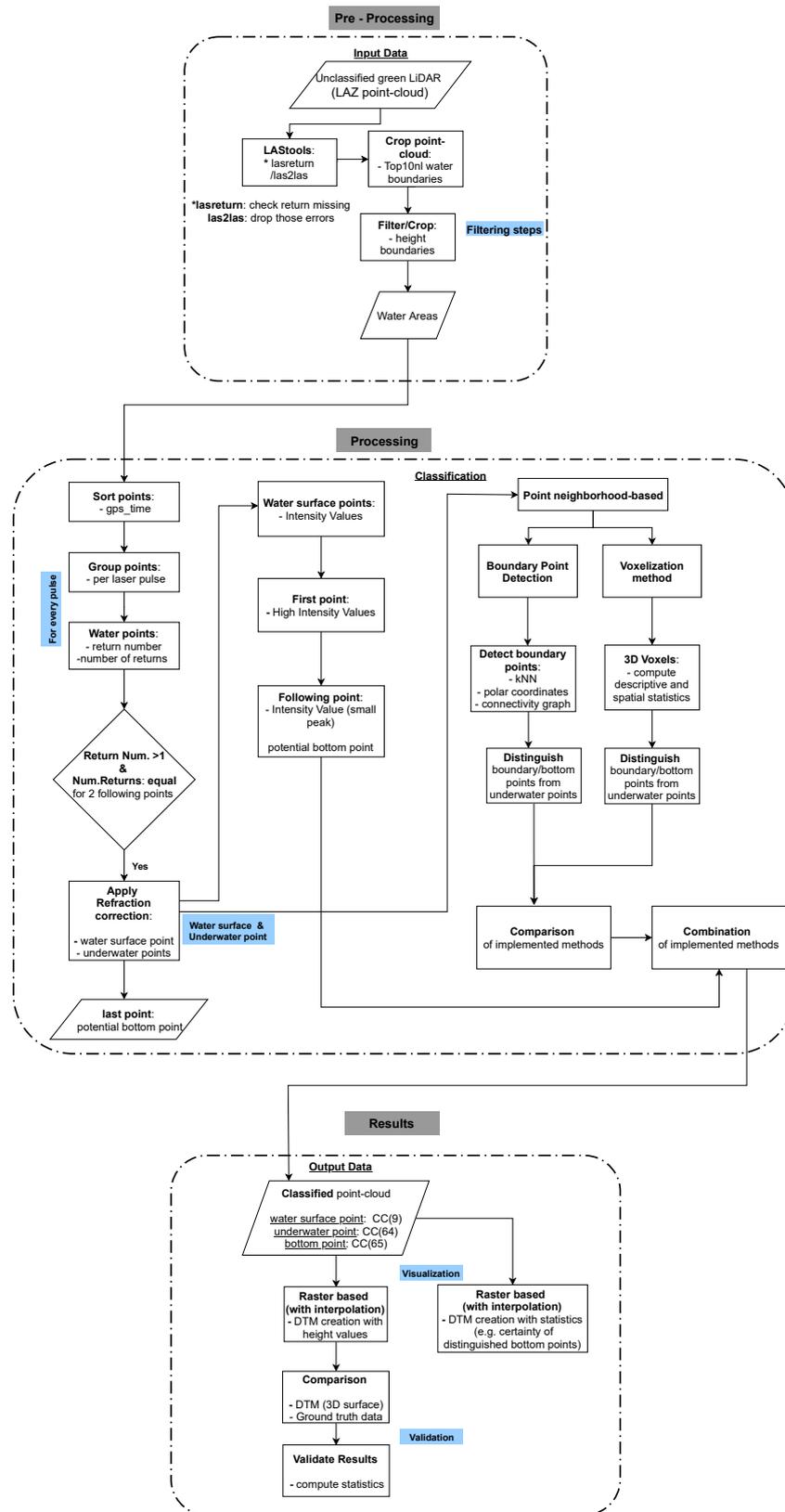
5 Methodology

Figure 2: Workflow chart; the three main steps of the study are the pre-processing,the processing and the results

Figure 2 displays a flowchart of the process of distinguishing potential bottom points in shallow water bodies in the Netherlands. The automatic detection of the waterbeds is carried out in several steps.

First, a pre-processing step is required to filter out errors like outliers due to reflection effect and to restore points from the refraction existence. That filtering pre-processing step is essential as many undesired points, such as urban structures (e.g. buildings, bridges) and vegetation, need to be removed from the initial raw green point cloud. Some LAStools functions (e.g. lasreturn, las2las) could be used to remove errors or duplicates from the dataset. Then, the input point cloud (unclassified green LiDAR data) will be filtered and cropped in the x,y and z dimensions based on top10nl water boundaries and specific height level, respectively. After cropping the input data only to the water areas, the remaining points are sorted by GPS time, and then are grouped per laser pulse based on some features (i.e. return number, number of returns).

Second, the processing step relates to the correction of the point cloud due to the refraction and slowdown effects, and then to the main classification procedure. Having each laser pulse with its points, the refraction correction algorithm is applied to correct the underwater points according to the corresponding water surface point. Using the corrected water points for each pulse, the intensity value of each point will be used to give insight about its return strength and its reflectivity. This will be an initial indication of the classification of the points of each laser pulse into water surface, underwater and bottom based only on their intensity. The main classification process is implemented using point neighbourhood based methods: boundary point detection (BPD) and voxelization algorithms. Both of them are going to be applied in parallel in an effort to distinguish the bottom points from the other water points, particularly the underwater ones. In the next step, they will be compared based on their results and then probably combined in order to increase the effectiveness of correctly classify the LiDAR points.

Last, the result of this process will be a classified point cloud with three classes: water surface, underwater and bottom points. After classifying the point cloud, a raster-based approach with interpolation (e.g. TIN, IDW) will be used for creating DTMs (3D surfaces) both with height values and statistics (e.g. certainty of distinguished bottom points). Then, comparison with ground truth data can be done. The last step is to validate the results and calculate statistics.

The following sections will describe the methodology stepwise.

5.1 Pre-Processing

As is known the LiDAR data were collected with a topo-bathymetric airborne laser scanner. The scanner is characterized by emitting green laser pulses with 532 nm wavelength, which can penetrate the water surface and capture the bottom of shallow water bodies. However, the raw point cloud data contain noisy points in the air column as the laser pulses are scattered by clouds, dust, birds or even other particles, while the noisy points can be presented below the water beds (Fig.10) (Brzank et al., 2008). In order to remove this noise before the further processing, the filtering/cropping process is required. To begin with, some LAStools functions such as `lasreturn` and `las2las` are used to select and drop error points by checking missing returns in the point cloud.

The filtering process will be carried out in two steps:

1. crop the input data in x and y dimensions based on Top10nl dataset (Fig. 11a)

2. clip the resulted data in z dimension based on a specific threshold in the height boundary (Fig. 11b)

Then, the desired water areas have been selected and saved as a separate point cloud (Fig. 11).

5.2 Processing

As the shallow water bodies have been selected in the previous step, their water points will be filtered out based on their characteristics, and then two point-neighbourhood based methods will be implemented, compared and possibly combined to receive a desirable classified point cloud.

5.2.1 Filter criteria

Since the remaining water points resulted from cropping/filtering process, their order in the new written point cloud file (LAZ) has been modified. That's the reason why they need to be sorted by their GPS time. Then, the sorted points will be grouped per laser pulse, based on some characteristics (Return Number, Number of Returns), as potential points on the water surface and their corresponding underwater points for the same beam can be pointed out.

In particular, a water surface point should present **Number of Returns** bigger than 1 (i.e. multiple points follow up) 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 written LAZ file with **Return Number** bigger than 1 and to have **Number of Returns** bigger than 1.

Index	Return Number	Number of Returns
6	2	2
7	2	2
9	2	2
11	**1**	**2**
12	**2**	**2**
13	4	4

Table 1: Water-surface point (blue) and underwater point (red)

As seen in Table 1, the 11th point in the dataset has **Return Number** = 1 and **Number of Returns** = 2 indicating that could be a water-surface point, whereas the following 12th point has **Return Number** = 2 and **Number of Returns** = 2 and is a corresponding underwater point.

5.2.2 Refraction correction and Slowdown effect

Moreover, when the green LiDAR data is recorded and stored, the refraction effect and the corresponding change in the speed of light takes place at the air-water interface have not been taken under consideration (Parrish et al., 2019). Both horizontal and vertical errors are introduced in the point cloud, resulting in points that are deeper and further away from the nadir than the true measurement (Parrish et al., 2019).

The light travels in a straight line through transparent media such as air or water. When it encounters surfaces such as the interface between different media (air and water), then one or more of the following things occur. A part of the light ray:

1. reflects off the surface and travels off in a different direction

2. passes from one medium (air) into the other (water) and continues on a new straight path

3. is absorbed

The light ray that hits a surface is the *incident ray* and the angle it hits this surface is called *incidence angle* (ϕ_{air}). The reflected part returns back to the atmosphere with an angle of reflection equal to the incidence angle, while the transmitted light ray bends (refraction). The magnitude and the direction of the refracted light ray depends on the refraction indexes of the two medias (air,water) and the incidence angle (SciencePrimer, 2019).

Except the refraction of the laser beam, the slowdown effect (as the speed of light in the water is smaller) occurs. The beam hits on a point on the water surface and measures how long it takes that beam to bounce back. By knowing the speed of light (c) and the measuring time (t), the distance (d) can be computed by using the formula ($d = c \cdot t/2$). When the beam penetrates the water, the captured points should have smaller distance from the initial water surface point due to the smaller speed of light.
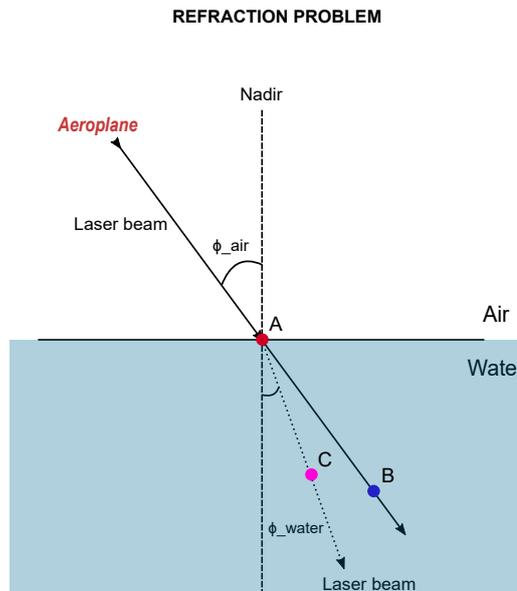

Figure 3: An example cross section of a water area; refraction problem is presented. Point A (red) indicates the point where the laser beam hits the water-surface, whereas Point B (blue) is the underwater point that was recorded from this beam and written in the point cloud document. Point C (magenta) indicates the corrected position of point B (i.e correct coordinates) after applying the refraction factor and considering the slowdown.

As seen in Figure 3, the laser beam (light ray) from the airplane laser scanner transmits through the air and part of it passes from one medium (air) into the other (water), but due to the refraction effect it should follow a new straight path (dashed line segment). However, the

laser scanner interprets that the laser pulse continues on an uninterrupted path (fixed line segment). As a result, the underwater point is written with incorrect coordinates in the LAZ file since the refraction effect has not been taken into account.

5.2.3 Intensity & Depth Information

Since the water points have been grouped per laser pulse and corrected from the refraction and slowdown effects, the characteristics of each point (especially the intensity value and depth) can be used as important factors to classify the points of every pulse into the three classes: water surface, underwater and bottom points.

As the laser beam penetrates the water surface, the intensity value is decreased due to the reflectivity of the water interface. As shown in Figure 4, the water surface point has a high intensity value as it's the first point that laser pulse hits on, while the following underwater points present gradually decreasing values. The last point of the pulse could be a potential bottom point of the water body, if a small peak in the intensity graph could be presented.
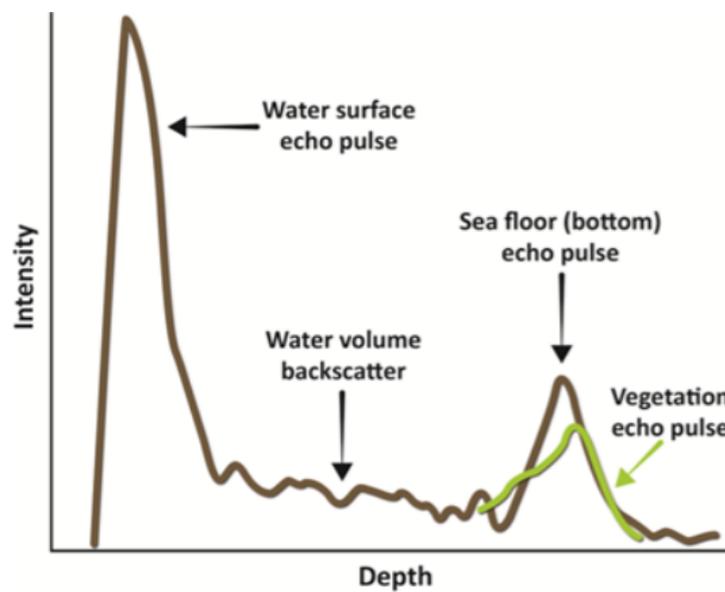

Figure 4: The three components of echoes from water area of green LiDAR consist of water surface return, water volume backscatter and bottom return (IQmulus, 2019).

The laser beams of the given raw point cloud are not stored as a full waveform (Fig. 4), but there are only a few intensity 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, the following values to the underwater points and the last value (last point) indicates the sea floor (bottom) echo pulse.

5.2.4 Detection of boundary points

By acquiring a point cloud, the identification of the boundary points is a challenging task. One approach that works well in 2D domain is to compute the convex hull, which contains all the points of a finite set of point cloud points. The convex hull is uniquely defined and can be easily computed, as it can be extracted from the Delaunay triangulation or computed by another algorithm (e.g. gift wrapping algorithm) (Ledoux et al., 2019). One method that could

work fine is the *quickhull* (Barber et al., 1996), but it is only able to distinguish boundary points that belong to the convex hull and there should be a set of points.

A more generic approach of the *convex hull* is the *alpha-shape* method for 3D point clouds. Every convex hull is alpha-shape, but not every alpha-shape is a convex hull. The limitation of *alpha-shape* approach is that depends on a parameter α. Since it is a parametric method, a specific value α may produce satisfactory result for a given point cloud, but not for another one with dense and sparse parts. Thus, it can give decent results in some regions and poor in others of the same point cloud.

The boundary point detection (BPD) algorithm was chosen to apply to this study as it does not demand any definition of threshold values in order to detect boundary points of an input point cloud (Mineo et al., 2019). The algorithm detects all the potential boundary points based on the local resolution of every region of the point cloud. It uses the k-nearest neighbourhood (k-NN) search method, as it is non-parametric and does not need a threshold value like alpha-shape method. By This technique is non-parametric as no assumptions on the underlying data distribution has been done. In Figure 5, the k-NN method finds the closest k-members of a specific point of point cloud (A to E) where k assumed is 30.

The points belong to each neighbourhood of points are represented through filled circles, while the other points are represented with empty circles.
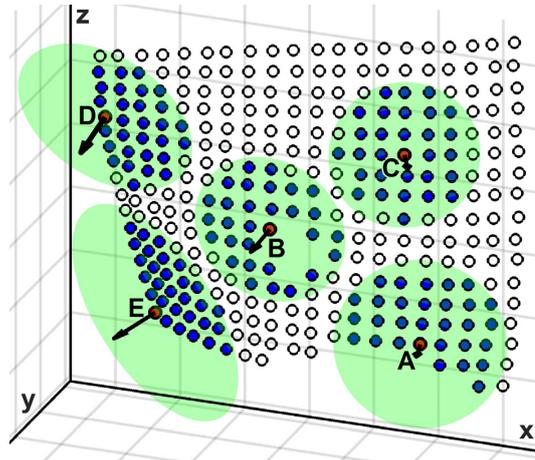

Figure 5: Point cloud with two boundary points on concave regions (A and B), an inner point (C) and two other boundary points on convex regions (D and E) in 3D space. The k value is set to 30 (Mineo et al., 2019).

For every point of the point cloud, the *local cloud resolution* is calculated. Given a point P_i, for every point of its neighbourhood P_{ij}, the minimum distance $d_{j,k}$ between the point P_i and all the neighbours is computed. The local point cloud resolution (β_i) of P_i is calculated by the equation

$$\beta_i = \mu_i + 2 \cdot \sigma_i \qquad (1)$$

where μ_i represents the mean value of minimum distances and σ_i their standard deviation. According to (Mineo et al., 2019), if the distances are Gaussian distributed, then the addition of $2\sigma_i$ ensures that 97,6% of the data are included and main outliers are eliminated.

This method (BPD) takes advantage of the fact that there is only one circle that passes through 3 points in the 3D space. A point P_i is characterized as boundary point if there is at least one circle with radius bigger or equal than β_i and if the sphere with centre the P_i point does not contain any other point of the neighbourhood such as points A and B in Figure 6.
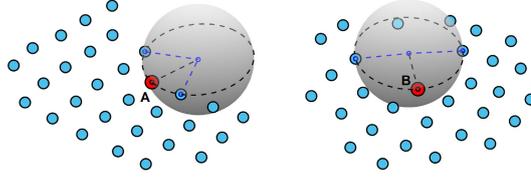

Figure 6: Both points A and B belong to the boundary (Mineo et al., 2019).

However, the above two conditions that categorize a point as boundary point of the dataset do not cover the case of an internal point in the point cloud (Fig. 5). In the case of points C,D and E, even if there is one circle with radius bigger or equal than β_i, there will always exist other points of the neighbourhood inside the circle (Fig. 5).

For that reason, all the distinguished boundary points from the first part of the algorithm and their neighbours will be projected to the best fit plane based on the normal vector of the point. The 2D points can be plotted in polar coordinates, where P_i shown in red and its neighbours in blue (Fig. 7).
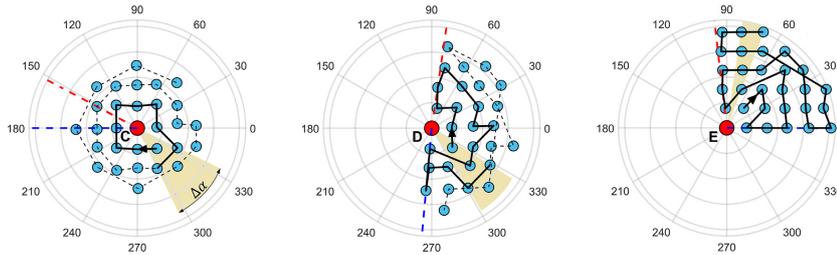

Figure 7: Plot with polar coordinates of points C,D,E and the neighbours (filled circles). The line segments display the application of the algorithm (Mineo et al., 2019).

The main idea of the BPD algorithm is that the point P_i belongs to the boundary area if it is not possible to find any path that can surround it and passes through its neighbours. Every point on the polar plot is determined by distance from the pole (R) and angle (θ). The algorithm tries to create a path that surrounds the parent point (P_i) at the pole. Moreover, the radial (r_j) and angular (θ_j) coordinates are normalized for every neighbour (j) based on the following equations, where R_{min}, R_{max}, θ_{min}, θ_{max} are the minimum and maximum radial and angular coordinates, respectively:

$$r_j = \frac{R_j - R_{min}}{R_{max} - R_{min}} \tag{2}$$

$$\theta_j = \frac{\theta_j - \theta_{min}}{\theta_{max} - \theta_{min}} \tag{3}$$

In order to determine the starting point of the surrounding path (connectivity graph), the parameter γ needs to be calculated for every neighbour and is given by the following equation:

$$\gamma_j = r_j + |\theta_j - \theta_{bis}| \qquad (4)$$

where θ_{bis} is the normalized angle of the direction bisecting the angle between θ_{min} and θ_{max}. Then, the neighbouring point with the smallest γ value is the starting point of the path and the other points of the neighbourhood are linked. Every point can be connected with the path only once.

The characterized parameter is the generic p-th remaining point is computed as:

$$\gamma_p = r_p + [(\theta_p - \theta_{last}) \cdot c] \qquad (5)$$

where θ_{last} is the normalized angular of the last point in the path and c is a factor equal to 1 or -1. The parameter c helps to select point that does not force the change of the path's direction. For instance, if the last point produced a clockwise rotation, any anti-clockwise rotation is penalized. The neighbouring point with the minimum γ value (Eq.5) is the new last point in the path.

Every time we add a point to the surrounding path, the sum of angles ($\alpha = \Sigma \Delta a$) and the sum of absolute values ($\tau = \Sigma |\Delta a|$) are updated. The whole process stops when no more points exist to be added in the path or sum ($\tau > 2\pi$). If $|\alpha| >= 2\pi$, then the investigated point (e.g. point C in Fig.7) is internal, whereas if $|\alpha| < 2\pi$ is a boundary point.

Thus, the condition $\tau > 2\pi$ helps to avoid computational effort, since a point belongs to the boundary without the need to connect all the neighbouring point to the path. For instance, point C in Fig.7 is not a boundary point by just selecting 18 out of 29 neighbours and the $|\alpha| < 2\pi$.

At the end, the potential boundary points of the point cloud can be detected containing the detected bottom points of the water body.

5.2.5 Voxelization

The voxelization of LiDAR data is the 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 too high, the number of voxels that contain no points become larger. But if the resolution is too 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

$$r_i = [\frac{x_i - x_{min}}{\Delta x}], c_i = [\frac{y_i - y_{min}}{\Delta y}], l_i = [\frac{z_i - z_{min}}{\Delta z}] \tag{6}$$

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. 8). 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.

Figure 8: LiDAR point cloud is divided into 3D voxels
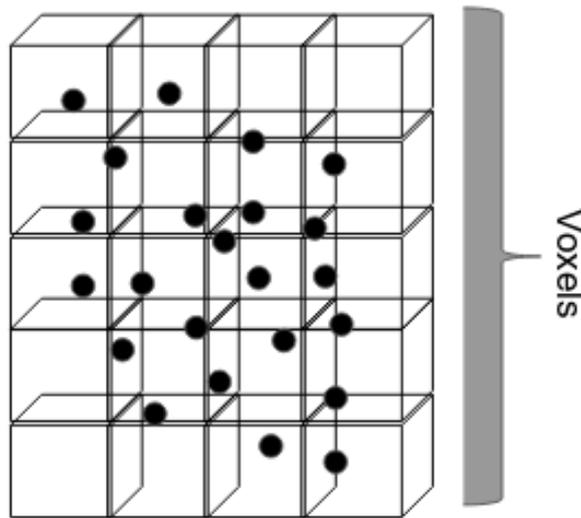
By calculating various descriptive and spatial statistics such as minimum z, maximum z, mean z, number of points, standard deviation for each 3D voxel, bottom points can be distinguished from the other underwater surface points. As seen in Fig. (9), the red points could correspond to the potential bottom points, while the blue ones can be classified as inner water points.
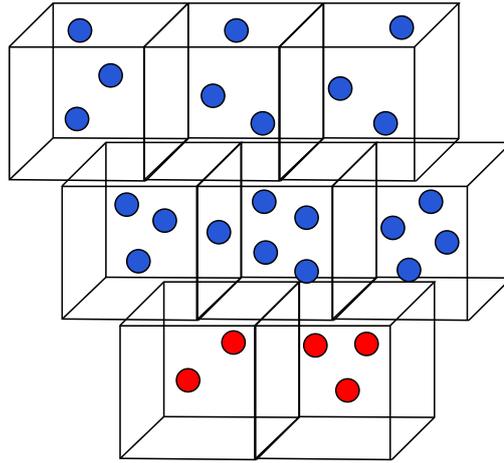
Figure 9: LiDAR points divided into 3D voxels; blue colour indicates water points while red one display potential bottom points

5.2.6 Comparison and Combination of methods

The next step is to compare the two aforementioned point neighbourhood based techniques, the boundary point detection (BPD) and the voxelization. The comparison can be done by easily checking the classified points from the two different techniques. Then, statistics for the two classified point clouds can be calculated by using accuracy indexes such as true positives, false negatives and false positives and kappa coefficient. However, in order to compute this statistical analysis, the most promising results from one of the two techniques can be used as reference data. These statistical results can indicate the effectiveness of each technique and its sufficiency to obtain the desired result; a classified point cloud.

Moreover, the prospect of combining the two methods in one to increase the potential of detecting bottom points, and to further classify the point cloud into the three classes: water surface, underwater and bottom points will be checked. As mentioned in Subsection , the intensity value and depth information of each point in a laser pulse will give insight about its reflectivity and return strength. As this will be an initial indication for the classification process, this information is going to be used in the potential combination of the two methods.

5.3 Results

5.3.1 Classification and Visualization of the water areas

The output data would be a classified point cloud with three classes: water surface, underwater surface and bottom points. By acquiring this information, then a raster-based approach with interpolation will be implemented for creating DTMs (3D surfaces) both with height values and statistics (e.g. certainty value of distinguished bottom points). Then, comparison with ground truth data (see Section 8.2) can be done by calculating height differences between the 3D surface and the measured ground truth points. The last step is to validate the results and calculate statistics that indicate the certainty of correctly classified points and especially the identified bottom points throughout this methodology.

6 Preliminary Results

In this section, some preliminary results will be presented. As it is already known that the input data are unclassified green LiDAR point cloud, some pre-processing steps are needed to filter out both noisy (e.g. outliers) and non-useful points (e.g. buildings, vegetation).

In order to apply the proposed pre-processing steps, a green LiDAR subset that covers part of the south area (water canals) of Pijnacker (east to Delft) was chosen (Fig. 10).
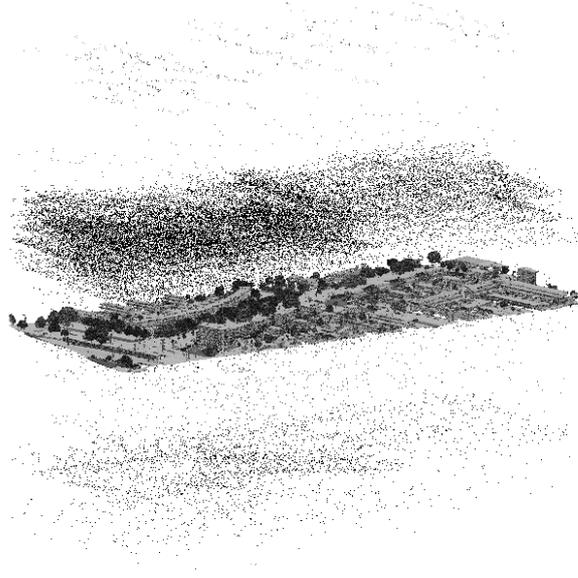

Figure 10: Green LiDAR subset with outliers; provided by Deltares

By using the *lasmerge.exe* tool, different LAZ files from laser scanner flights covered the same area (Pijnacker) were merged into one LAZ file. The new merged LAZ dataset consists of various dense and sparse parts as points from overlapping areas existed in the different LAZ files were added to it.

Moreover, the *lasinfo.exe* tool was used to get more detailed information about the LAZ header and point record entries were retrieved. In header part, the version number is 1.4 with point data format 6, scale factor 0.01 in all three dimensions (x,y,z) and offset values in x = 89876, y = 447436 and z = -105. Also, the 3D extent of the dataset (min x = 90250.000, min y = 447100.000, min z = -324.924, max x = 90553.103, max y = 447299.999, max z = 436.921) and the number of point records (15.008.072 points) are clearly presented. The Coordinate Reference System (CRS) of the data is the Amersfoort/RD New (EPSG:28992). As far the point record part, the attributes of all the LAS points such as the X,Y,Z coordinates, the intensity, the return number, the number of returns, the edge of flight line, the scan direction flag and the GPS-time are given.

The next step was to acquire only the water areas from the subset point cloud. Specifically, the input dataset (Fig. 10) was cropped (in x and y dimensions) based on the top10nl shapefiles, which provide the exact boundaries of the desired water areas (i.e. water canals) (Fig. 11a). Then, it clipped on specific height level (i.e. between 0m to -3m) (Fig. 11b). As a result, a new

point cloud dataset with only the water points and without outliers was created (Fig. 11b).

(a) Cropping in x and y dimensions based on Top10nl shapefile

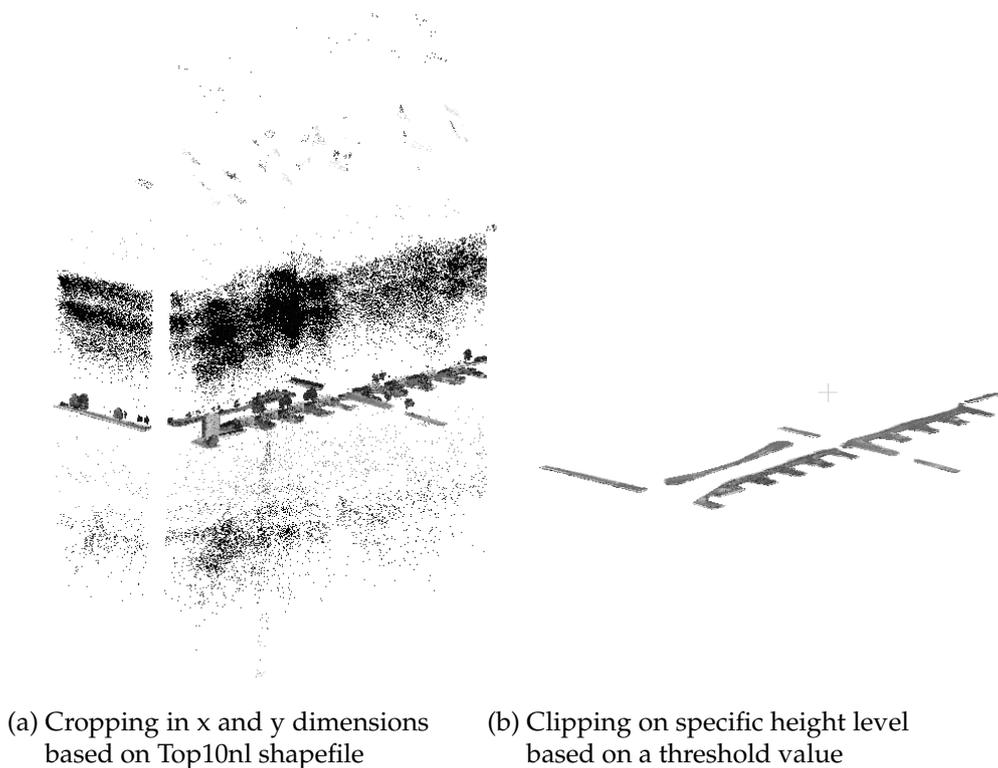(b) Clipping on specific height level based on a threshold value

Figure 11: Cropped water areas of the subset point cloud; provided by Deltares

At this point, it is important to mention that the appropriate height values of the clipping part were chosen after numerous tests of those that best eliminate the outliers in z dimension. This is a hard-coded approach that demands tests and manual processing, but it was a quick and efficient brute force way to filter this particular subset. A more generic filtering procedure in z dimension will be experimented in the further steps of the thesis.

Also, the point data format were modified from number 6 to 3 in order to allow the compatibility with point cloud libraries in Julia. This modification in the header part will not affect that much the further steps of the proposed workflow, as the difference between 6 and 3 is related to the amount of used core bytes. To be more specific, the difference is there are more bits for return numbers in order to support up to 15 returns, more bits for point classifications to support up to 256 classes, higher precision scan angle (16 bits instead of 8), and the GPS time is mandatory (ASPRS, 2013).

Obtaining the desired water areas from the original point cloud, the remaining points were sorted based on their GPS time. This was essential in order to ensure that the remaining points follow the right order in the file (i.e. 1st collected point corresponds to the 1st written point in the file), since many points were eliminated from the file during previous cropping steps. Then, the remaining points were grouped into separate laser pulses according to their return number and number of returns (see Subsection 5.2.1).

Considering that even if light travels in a straight line through transparent media such as air or water, when it encounters surfaces such as the interface between different media (air and water) two things occur. Specifically, a part of the light array is reflected on the surface and returns back to the atmosphere while transmitted light ray bends (refraction). Except the

refraction of the laser beam, the slowdown effect (as the speed of light in the water is smaller) occurred. As a result, the underwater point is written with incorrect coordinates in the LAZ file and the points need to be corrected by applying the refraction correction as described in Section 5.
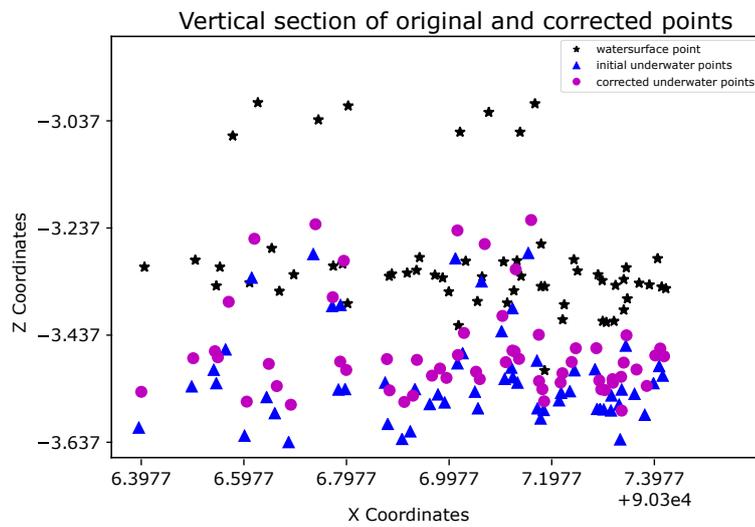

Figure 12: Vertical section of original and corrected water points

As seen in Figure 12, the vertical section of the original and corrected points in the Z and X axes is presented. Each water surface point (black) corresponds to an initial underwater point (blue) with its refracted point (magenta). Looking closely at this figure, a few water surface points differ a bit more in the height level than the other ones. However, this difference is ranged around 20cm indicating possibly the existence of small objects-structures or plants on the water surface.
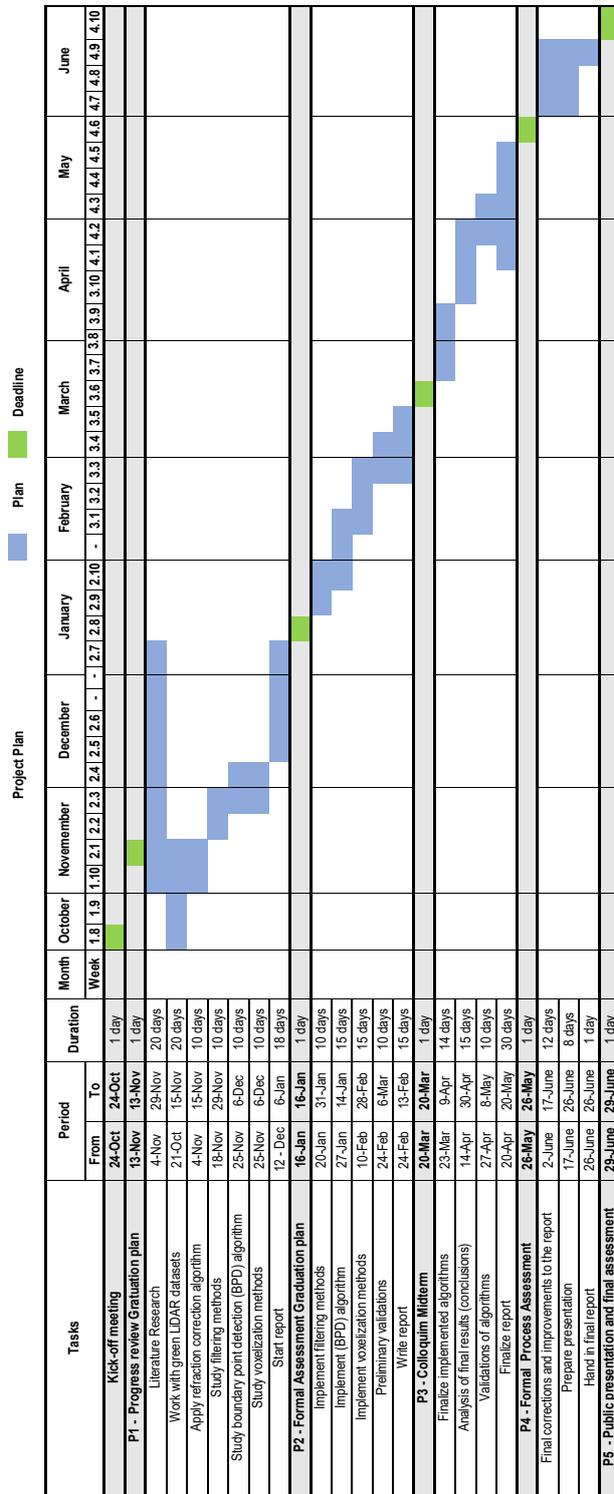
7 Time Planning

7.1 GANTT chart

Figure 13: GANTT chart indicating the planned project phases and deadlines

7.2 Meetings

Another important part of the time planning is the arrangement of meetings. Weekly meetings will be held with the daily supervisor Maarten Pronk to evaluate and discuss the improvements of the project. Additional guidance and helpful feedback will be provided by the graduation tutors Ravi Peters and Jantien Stoter every 15 days.

8 Tools and Data

8.1 Tools

In order to read, process and manipulate green LiDAR point cloud several tools need to be used.

Specifically, the Julia [1]programming language will mainly be used for data processing. 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. Due to the highly demanding computational analysis of a point cloud dataset, all the algorithms will be implemented in Julia [1] as it is suitable for high-performance numerical analysis and computational science.

Existing packages such as LazIO [2], LasIO [3], FileIO [4] will be used to traverse the point cloud dataset, read, store the points and the header file.

Also, by using plotting packages like Plots [5] and Plotly [6] in Julia and Matplotlib [7] in Python [8], several 2D and 3D plots will be produced.

For point cloud viewing, an open-source LiDAR viewer called Displaz [9] is going to be used. Using the Shader Parameters dialogue box, parameters (e.g..point radius, colour mode, selection etc.) change the way the points displayed. Displaz allows full control over how points are displayed in the 3D window by allowing the user to edit,customize and recompile these parameters.

Another software for rapid LiDAR (LAZ/LAS files) processing is the LAStools [10], which provides useful tools for getting point cloud info, clipping, correcting etc. Some tools such as las2las, lassort, lasclip, lasinfo are going to be used in the pre-processing step to deal with the initial point cloud dataset.

8.2 Data

As for the datasets to be used for this study, Deltares has already received green LiDAR data from six different regions in the Netherlands. Several water boards had organized a pilot project in cooperation with Deltares, Stowa and Waternet where tried to examine the potential

---

[1]https://julialang.org

[2]https://github.com/evetion/LazIO.jl

[3]https://github.com/visr/LasIO.jl

[4]https://github.com/JuliaIO/FileIO.jl

[5]http://docs.juliaplots.org/latest/

[6]https://plot.ly/julia/

[7]https://matplotlib.org

[8]https://www.python.org

[9]https://github.com/c42f/displaz

[10]http://lastools.org

of green LiDAR for the muddy shallow Dutch water-bodies. This 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.

As seen in Table 2, 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 (km^2)	Average Point Density (points per m^2)	Maximum soil depth (m)
Westerschelde	14,263	37,33	-5.04
Gevelingenmeer	12,304	58,89	-5,24
Regio Rotterdam	13,458	17,63	-2.46
Oss	18,263	15,70	-2,30
Vechtgebied	14,041	23,167	-1,08
Dinkelgebied	7,527	52,63	-1,12

Table 2: Characteristics of green LiDAR datasets from six different regions in the Netherlands (van Tol, 2019)

(a) Westerchelde   (b) Gevelingenmeer   (c) Regio Rotterdam

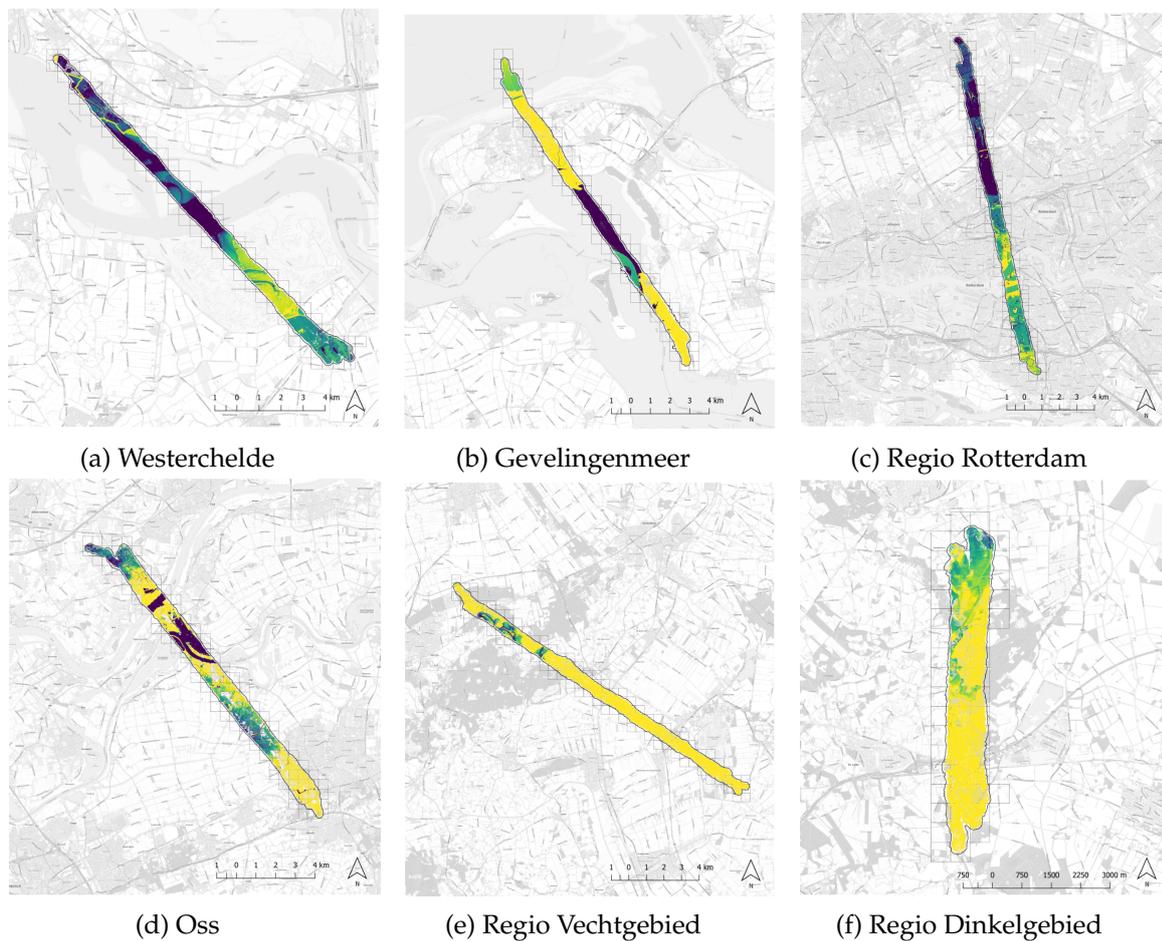(d) Oss   (e) Regio Vechtgebied   (f) Regio Dinkelgebied

Figure 14: Datasets of six different regions in the Netherlands (van Tol, 2019)

In Figure 14, the six different point cloud datasets are displayed on a background map. 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.

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 where will be tested with the results of the implemented algorithms.

Another useful data source is the new delivered green LiDAR data from the Wadden Sea area, in the north-east part of the Netherlands. That area forms a shallow body of water with tidal flats and wetlands with high biological diversity. The next coming weeks Deltares will receive the processed and un-processed raw point cloud that will be used in this study as well.

References

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.

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

Barber, C. B., Dobkin, D. P., and Huhdanpaa, H. (1996). The quickhull algorithm for convex hulls. *ACM Transactions on Mathematical Software*, 22:469–483.

Brzank, A. and Heipke, C. (2007). Supervised classification of water regions from lidar data in the Wadden Sea using a fuzzy logic concept. page 6.

Brzank, A., Heipke, C., Goepfert, J., and Soergel, U. (2008). Aspects of generating precise digital terrain models in the Wadden Sea from lidar–water classification and structure line extraction. *ISPRS Journal of Photogrammetry and Remote Sensing*, 63(5):510–528.

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

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.

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].

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

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.

Mineo, C., Pierce, S. G., and Summan, R. (2019). Novel algorithms for 3D surface point cloud boundary detection and edge reconstruction. *Journal of Computational Design and Engineering*, 6(1):81–91.

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.

SciencePrimer (2019). Light reflection and refraction.

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

Vazquez, D. B. (2017). 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.

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.

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.