CZECH UNIVERSITY OF LIFE SCIENCES PRAGUE

FACULTY OF ENVIRONMENTAL SCIENCES

DEPARTMENT OF SPATIAL SCIENCES

Comparison of LiDAR ground filtering algorithms

Diploma Thesis

Supervisor:
Doc. Vítězslav Moudrý

Author:
Mohammad Ebrahimian

2023

CZECH UNIVERSITY OF LIFE SCIENCES PRAGUE

Faculty of Environmental Sciences

DIPLOMA THESIS ASSIGNMENT

Mohammad Ebrahimian, MSc.

Landscape Engineering
Landscape Planning

Thesis title

Comparison of LiDAR ground filtering algorithms

Objectives of thesis

The thesis aims to evaluate the performance of ground-filtering algorithms.

Methodology

The ways of acquisition of accurate information on the 3D structure of the environment have greatly expanded over the past two decades. In particular, laser altimetry, commonly referred to as light detection and ranging (LiDAR) or airborne laser scanning (ALS), has revolutionized the quality of 3D representation of the environment and has become the primary method for the acquisition of accurate terrain information. Regardless of the application and the final product needed, an inevitable and the most critical step of point cloud processing is ground filtering (i.e., the process when points that represent bare ground are separated from non-ground points representing objects above the bare ground, such as trees). The correct filtering (sometimes referred to as classification) of ground points is essential to the subsequent creation of DSMs, DTMs or other derived products. The thesis aims to evaluate the performance of ground filtering algorithms with respect to the land cover of the area under study.

The proposed extent of the thesis

40 pages

Keywords

ALS, Filtering, Ground, Landcover, Lidar, Point cloud

Recommended information sources

Klápště, P., Fogl, M., Barták, V., Gdulová, K., Urban, R., & Moudrý, V. (2020). Sensitivity analysis of parameters and contrasting performance of ground filtering algorithms with UAV photogrammetry-based and LiDAR point clouds. International Journal of Digital Earth, 13(12), 1672-1694.

Meng, X., Currit, N., & Zhao, K. (2010). Ground filtering algorithms for airborne LiDAR data: A review of critical issues. Remote Sensing, 2(3), 833-860.

Moudrý, V., Klápště, P., Fogl, M., Gdulová, K., Barták, V., & Urban, R. (2020). Assessment of LiDAR ground filtering algorithms for determining ground surface of non-natural terrain overgrown with forest and steppe vegetation. Measurement, 150, 107047.

Pingel, T. J., Clarke, K. C., & McBride, W. A. (2013). An improved simple morphological filter for the terrain classification of airborne LIDAR data. ISPRS Journal of Photogrammetry and Remote Sensing, 77, 21-30.

Sithole, G., & Vosselman, G. (2004). Experimental comparison of filter algorithms for bare-Earth extraction from airborne laser scanning point clouds. ISPRS journal of photogrammetry and remote sensing, 59(1-2), 85-101.

Zhang, W., Qi, J., Wan, P., Wang, H., Xie, D., Wang, X., & Yan, G. (2016). An easy-to-use airborne LiDAR data filtering method based on cloth simulation. Remote Sensing, 8(6), 501.

Expected date of thesis defence

2022/23 SS – FES

The Diploma Thesis Supervisor

doc. Ing. Vítězslav Moudrý, Ph.D.

Supervising department

Department of Spatial Sciences

Electronic approval: 16. 2. 2023	Electronic approval: 20. 2. 2023
doc. Ing. Petra Šímová, Ph.D.	**prof. RNDr. Vladimír Bejček, CSc.**
Head of department	Dean

Prague on 29. 03. 2023

Declaration

I hereby declare that I have independently elaborated the diploma/final thesis with the topic of: *Comparison of LiDAR ground filtering algorithms* and that I have cited all the information sources that I used in the thesis and that are also listed at the end of the thesis in the list of used information sources.

I am aware that my diploma thesis is subject to Act No. 121/2000 Coll., on copyright, on rights related to copyright and on amendment of some acts, as amended by later regulations, particularly the provisions of Section 35(3) of the act on the use of the thesis.

I am aware that by submitting the diploma thesis I agree with its publication under Act No. 111/1998 Coll., on universities and on the change and amendments of some acts, as amended, regardless of the result of its defence.

With my own signature, I also declare that the electronic version is identical to the printed version and the data stated in the thesis has been processed in relation to the GDPR.

Prague 30/3/2023
Mohammad Ebrahimian

Acknowledgment

I would like to express my deepest appreciation to my supervisor, Doc. Vítězslav Moudrý, that without his help and motivation, this study would never have come to an end.

Abstrakt

Light detection and ranging (LiDAR) je metoda určování vzdáleností, která po aplikaci filtru povrchu (ground filtering) umožňuje vygenerování digitálního modelu terénu (digital terrain model/DTM) ve vysokém rozlišení. Cílem této práce je prozkoumat chování následujícíh algoritmů pro filtrování povrchu v různých krajinný pokryv a svazích: Multiscale Curvature Classification (MCC), Cloth Simulation Filtering (CSF) a Progressive Morphological Filter (PMF). K tomu je použit kód v jazyce R za použití knihovny lidR, data jsou získána z mračna bodů Dánska. Jak svažitost tak krajinný pokryv měly vliv na přesnost generovaného terénu, a všechny tři algoritmy dávaly ve všech kategoriích obstojné výsledky. Zvýšení svažnosti vedlo k nižší chybovosti a přesnějším modelům, a obecně lze říci, že v kategorii ze všech pokryvů terénu nejlépe vycházely stromy. Tato práce navrhuje vhodné výchozí hodnoty uživatelům, kteří chtějí optimalizovat filtrovaní povrchu v datech získaných metodou LiDAR.

Klíčová slova

LiDAR, lidR, Filtrování povrchu, Klasifikace, Digitální Model Terénu

Abstract

Light detection and ranging (LiDAR) is a method and technology for determining distances that by applying ground filtering on it, high resolution of digital terrain model (DTM) could be generated. The aim of this study was to investigate the behavior of Multiscale Curvature Classification (MCC), Cloth Simulation Filtering (CSF) and Progressive Morphological Filter (PMF) ground filtering algorithms in different landcover and slopes. By lidR package and coding in R, the result obtained from point cloud of Denmark, and among them the best parameters for each category of landcover and slope. Both slope and landcover had an influence on the results and all the three algorithms showed an acceptable performance in all categories. Increasing slope led to decreased errors and increased success rates, and in general the Tree Cover category had the better result than other landcovers. Overall, this study provides default parameters for users who wish to optimize ground filtering algorithms for their LiDAR data.

Keywords

LiDAR, lidR, Ground filtering, Classification, Digital Terrain Model

Table of contents

1. Introduction

LiDAR is known as an effective means of acquiring high-resolution elevation data. The digital terrain models (DTMs) generated through this technique usually have an obvious advantage of high precision, especially for forested areas where can be detected both the ground surface and the canopy (Chen et al., 2016). LiDAR made a revolution in topographic terrain capturing, especially in the generation of digital terrain models (DTM) (Kraus & Pfeifer, n.d.).

The nature of LiDAR data offers the potential for extracting surface information for many applications (Priestnall et al., n.d.). Laser pulse can reflect from leaves, buildings, ground etc. In order to generate DTM, the ground points in the LiDAR data must be distinguished from the non-ground points (e.g., trees or buildings). This process is called ground filtering. LiDAR intensity data are used to produce the binary classification of ground and vegetation with good accuracy (Wang & Glenn, 2009). By using LiDAR point clouds and color RGB of the points, classification result for both buildings and vegetation in an urban environment (Bandyopadhyay et al., 2013).

But this thesis focuses on the ground filtering by the geometry and elevation of points (xyz). PMF, CSF and MCC are some different algorithms for this purpose with different approaches and methods. All of them have to be parameterized. There are some papers that have already worked on this topic. (Moudrý et al., 2020) worked on the 6 different algorithms in different slopes in shrub, grassland and forest landcover. (Klápště et al., 2020) compared the performance of six algorithms for ground filtering of LiDAR and photogrammetric point clouds in different landcovers.

While these algorithms were presented by their authors, the default parameters were introduced too. But the problem is that these default parameters are just for some specific areas. The question is what if the area has different landcovers than the one introduced as default? What if the slope exists in the area?

In this thesis, by the help of R programming and the lidR package that is open source and free to access, I am going to define some default parameters for PMF, CSF and MCC ground filtering algorithms in different landcovers and different slopes and find the relation of the area characteristics and input parameters. The more specific research questions were as follows:

1. What is the effect of the landcover and slope on the different ground filtering algorithms?

2. How do the parameters of algorithms change among different landcover types and degrees of slope?

3. What are the best parameters settings as input for different algorithms, landcover types and slopes?

2. Literature review

2.1. History of remote sensing

One of the earliest dreams of humankind was observing and watching the Earth from above. This dream came to reality gradually, from the balloons, kites, pigeon and gliders. In 1858, the first aerial photos were taken from a height of 80 m over Bievre, France, by Gaspard Félix Tournachon using cameras mounted on a hot air balloon and the first balloon photography used for urban planning were acquired by James Wallace Black in 1860 over the city of Boston (Chuvieco, n.d.).

The British used kites to reach to aerial photography from 1880s, and in the early 1900s, human used carrier pigeons to carry the lighter cameras. The great San Francisco earthquake of 1906 was captured on film using a panoramic camera mounted 600 m above San Francisco Bay and supported by a string of kites.

In 1908, 5 years after the Wright brothers built the world's first operational aircraft, an aircraft was first used as a platform for aerial photography (Dong & Chen, n.d.). In World War I and World War II aerial photos played a significant role. During 1920s and 1930s aerial photography was used as a source and reference for topography maps.

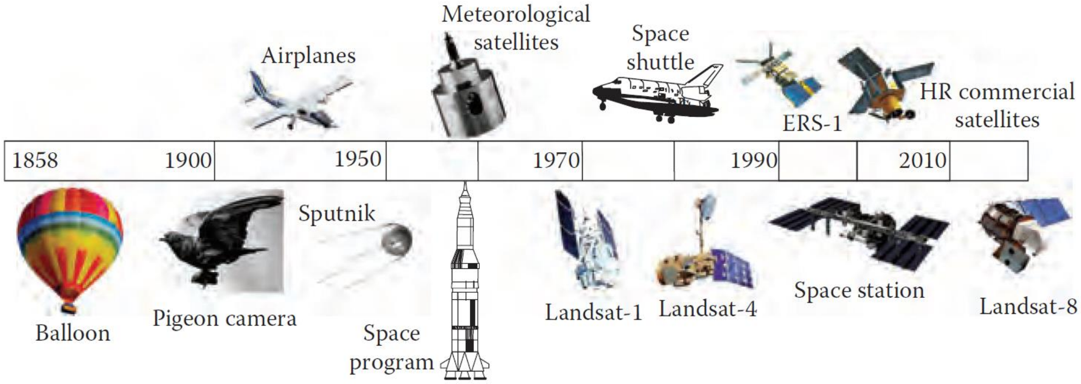


Figure 1. Historical development of remote sensing systems (Chuvieco, n.d.).

The term "remote sensing" was first coined by Evelyn Pruitt of the U.S. Office of Naval Research in the 1950s, and the traditional aerial photography gradually evolved into remote sensing around 1960 (Dong & Chen, n.d.). Photography was the main sensor at that time. Remote sensing means observing Earth from above, in any way and aspect and gathering data and information, about the Earth or a particular case study. The collected data will be interpreted by human or computer and at last these data from remote sensing will be converted to a meaningful maps, graphs or figures for the usage of variety of end-users such as, foresters, farmers, fishers, journalists, ecologists, geographers etc.

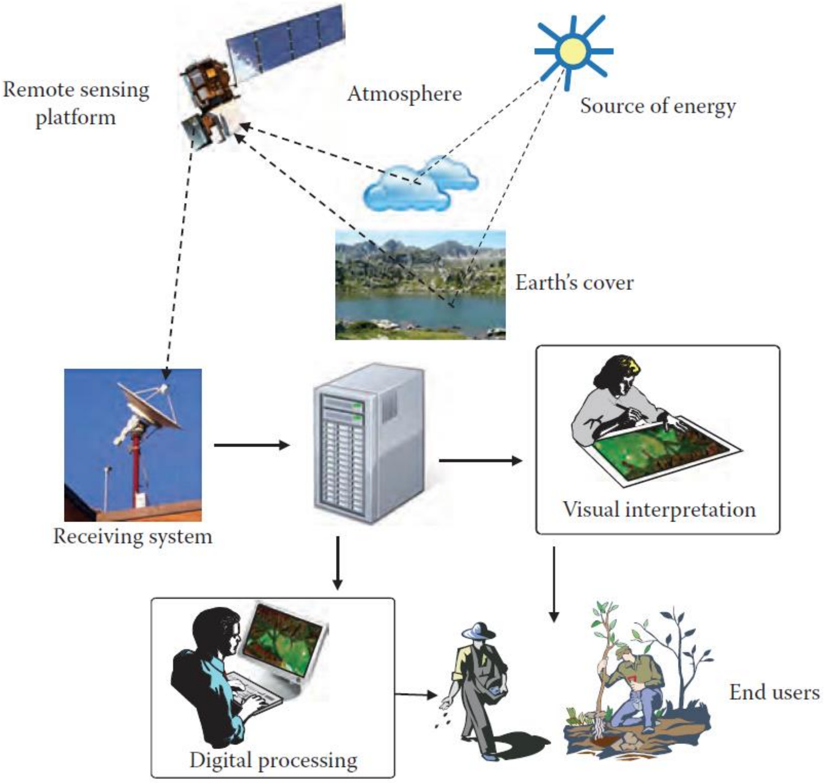


Figure 2. Illustration of the main components associated with remote sensing activities (Chuvieco, n.d.).

Although many other definitions of remote sensing exist in literature it is commonly accepted that the basis for remote sensing is the electromagnetic spectrum (Dong & Chen, n.d.). Typical eye-safe LiDAR have a frequency of 200 THz and a wavelength around 1.5 mm. LiDAR is often used as an imaging sensor. It can be for two-dimensional (2D) imaging, similar to the eye or 3D imaging, where range is measured in each pixel. (McManamon, 2019).

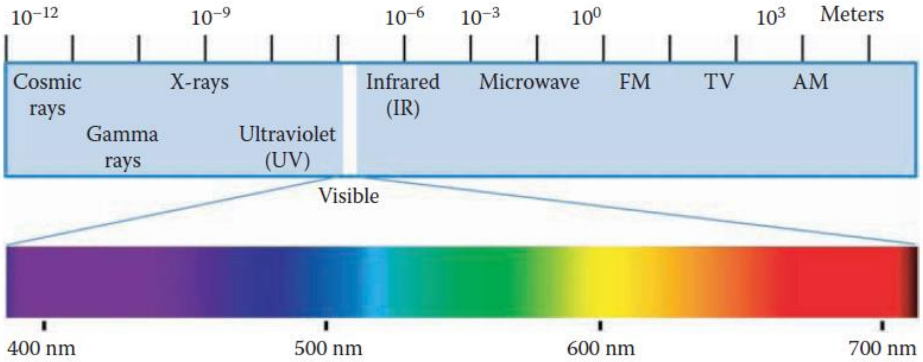


Figure 3. The electromagnetic spectrum. The numbers show wavelengths of spectral regions (Dong & Chen, n.d.)

2.2. Types of remote sensing

Remote sensing has two general types. Passive and Active. Passive monitoring uses ambient light sources, like the ultraviolet–visible–near-infrared radiation from the sun, while for active illumination laser sources are employed (Brydegaard et al., 2016). Active remote sensing systems, on the other hand, emit radiation toward the target using their own energy source and detect the radiation reflected from that target (Dong & Chen, n.d.). The important advantage of active remote sensing is that it is independent from the light of the sun and weather conditions will affect it less than passive one.

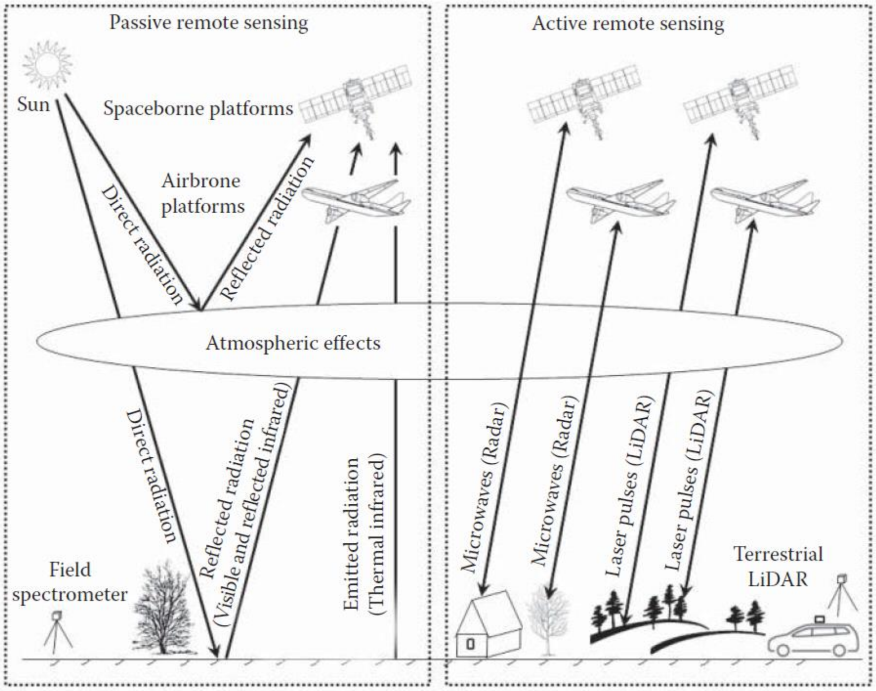


Figure 4. Passive and active remote sensing (Dong & Chen, n.d.).

To make it a little clearer, imagine a camera, that in the daylight it takes photo of a landscape, it uses the beams that coming from sun and reflecting from the objects, it is passive. Now imagine, a camera taking a photo during a night, it uses flash to provide light beams to reflect again to the camera to take a picture, it is active. The advantage of active remote sensing is the ability to obtain measurements anytime, regardless of the time of day or season, but on the other hand, these systems require the generation of a large amount of energy to adequately illuminate targets.

2.3. What is LiDAR?

Light detection and ranging (LiDAR) mapping is an accepted method of generating precise and directly georeferenced spatial information about the shape and surface characteristics of the Earth (Schmid et al., 2012). LiDAR is an active sensor system that emits pulses to the ground and collects the reflected ones from the surfaces. LiDAR can collect very dense and accurate elevation data. The LiDAR system is like Radar but using laser light pulses instead or radio waves. LiDAR is typically “flown” or collected from airplanes where it can rapidly collect points over large areas and LiDAR is also collected from ground-based stationary and mobile platforms (Schmid et al., 2012).

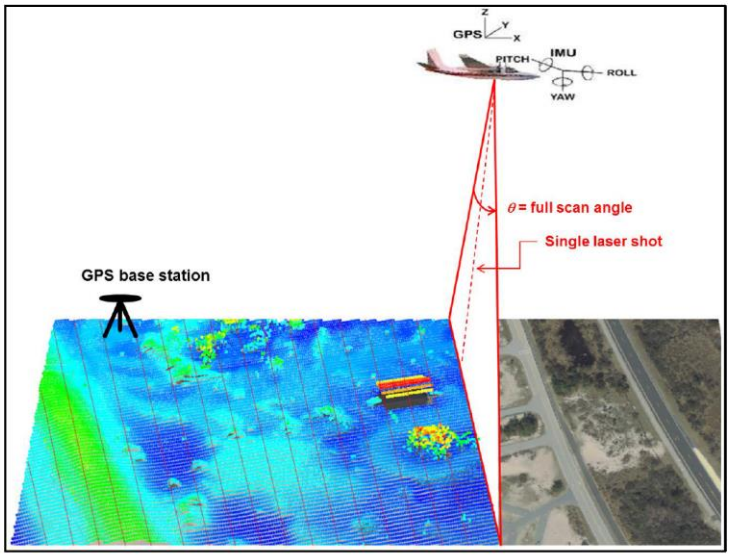


Figure 5. Schematic diagram of airborne LiDAR performing line scanning resulting in parallel lines of measured points (other scan patterns exist, but this one is fairly common) (Schmid et al., 2012).

The general idea of the LiDAR is simple. A pulse travels from Laser to the Earth and back, LiDAR will measure the travel time. The speed of pulse is specified. Then it is easy to find out the distance of the point on the earth.

$$d = \frac{ct}{2}$$

While *d* is distance, *t* is the measured time and *c* is the speed of light (300,000 km/sec).

But the reality is not simple as it seems. Because so many factors should be considered, such as, the speed and movement of plane (airborne). Inertial Measuring Units (IMU) or Inertial Navigation Systems (INS) have been instrumental in making the exact positioning of the plane possible with the help of GPS on the plane, The GPS positions are recorded by the plane and at a ground station with a known position. The ground station provides a “correction” factor to the GPS position recorded by the plane (Schmid et al., 2012).

In general, there are two types of LiDAR, Airborne and Terrestrial. Airborne LiDAR is installed on a helicopter or drone for collecting data and emits light towards the ground surface and recording the returns after hitting the object.

Terrestrial LiDAR, unlike Airborne, are installed on moving vehicles or tripods on the earth surface for collecting accurate data points. They are frequently used to monitor roads, examine infrastructure, or even gather point clouds from both inside and outside of buildings.

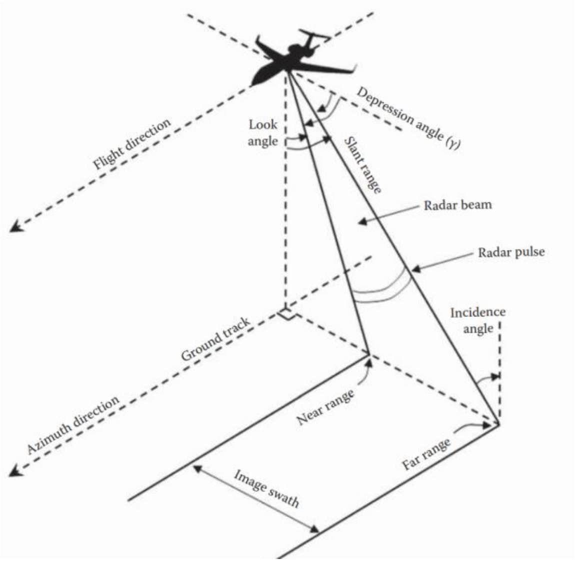


Figure 6. The concept of Airborne LiDAR (Dong & Chen, n.d.)

These are briefly the parameters of Airborne Laser Scanning (ALS):

Repetition rate: This is rate of pulsing, and it'll be measured in kilohertz (KHz). A sensor operating at 200 KHz means the LiDAR will pulse at 200,000 times per second.

Scan angle: The distance that the scanner travels from one end to the other and it is in degrees.

Flying attitude: the elevation that Airborne flies and directly affects the quality of the point (higher elevation, lower dense points).

Swath: This is the length of the LiDAR system's coverage region. It may change based on the flying height and scan angle.

Airborne LiDAR sensors can record discrete return measurements with numerous records for each pulse that is emitted at fixed time intervals such as 1 ns (about 15 cm sampling distance). Full-waveform LiDAR is mainly used in forestry applications, whereas discrete return LiDAR can be used in many fields. (Dong & Chen, n.d.).

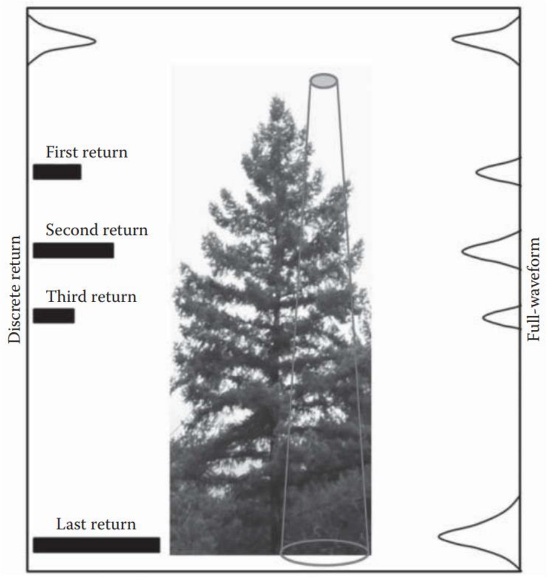
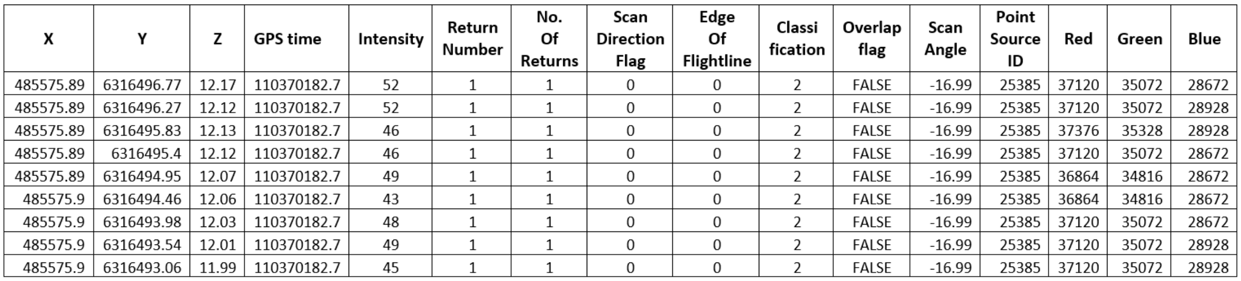


Figure 7. Discrete return and full-waveform measurement using airborne LiDAR (Dong & Chen, n.d.).

LAS is the format of point cloud produced by LiDAR and LAZ is the compresses format of the LAS. These files contain some data for every point: X, Y and Z as the geometry of point that is the most important data, and the rest are Intensity, Return number, Classification, Scan angle etc. (*LAS Specification 1.4-R15 Release Information*, 2002).

Table 1. A part of a LAS file and the data that contains.



X	Y	Z	GPS time	Intensity	Return Number	No. Of Returns	Scan Direction Flag	Edge Of Flightline	Classi fication	Overlap flag	Scan Angle	Point Source ID	Red	Green	Blue
485575.89	6316496.77	12.17	110370182.7	52	1	1	0	0	2	FALSE	-16.99	25385	37120	35072	28672
485575.89	6316496.27	12.12	110370182.7	52	1	1	0	0	2	FALSE	-16.99	25385	37120	35072	28928
485575.89	6316495.83	12.13	110370182.7	46	1	1	0	0	2	FALSE	-16.99	25385	37376	35328	28928
485575.89	6316495.4	12.12	110370182.7	46	1	1	0	0	2	FALSE	-16.99	25385	37120	35072	28672
485575.89	6316494.95	12.07	110370182.7	49	1	1	0	0	2	FALSE	-16.99	25385	36864	34816	28672
485575.9	6316494.46	12.06	110370182.7	43	1	1	0	0	2	FALSE	-16.99	25385	36864	34816	28672
485575.9	6316493.98	12.03	110370182.7	48	1	1	0	0	2	FALSE	-16.99	25385	37120	35072	28672
485575.9	6316493.54	12.01	110370182.7	49	1	1	0	0	2	FALSE	-16.99	25385	37120	35072	28928
485575.9	6316493.06	11.99	110370182.7	45	1	1	0	0	2	FALSE	-16.99	25385	37120	35072	28928

2.4. Ground filtering

The process of separating non-ground points from terrain measurements is referred to as “terrain filtering” or “ground filtering” and is a challenging research task in LiDAR mapping applications (K. Zhang & Whitman, 2005). In recent years, several filter algorithms have been created for automatically extracting ground points from airborne

LiDAR point clouds that *morphological*, *slope-based*, and *interpolation-based* approaches being the most widely used (Liu, 2008).

The *interpolation-based* filter method, also called linear prediction, was first proposed by Kraus and Pfeifer (1998) (Kraus Ł & Pfeifer, 1998). It iteratively approximates the terrain surface using weighted linear least squares interpolation. This filter method was originally developed for filtering LiDAR data and terrain modelling in forested areas, and later extended to use in urban areas.

The *slope-based* filter developed by Vosselman (2000) (Vosselman, 2000) assumes that the gradient of the natural slope of the terrain is distinctly different from the slope of non-terrain objects such as buildings and trees. Slope-based filters work well in flat terrain but become more inaccurate as the slope of the terrain increases.

The morphological filter is based on the idea of mathematical morphology, which has been used to identify objects in a greyscale image by using morphological (Liu, 2008). The elevations of non-ground objects such as trees and buildings are usually higher than ground points. If points are converted to a greyscale image in terms of elevation, the non-ground objects can be identified by the difference of grey tone (K. Zhang et al., 2003).

Ground filtering is important because it eliminates non-ground objects such as buildings, trees, and cars from the point cloud data set and helps in creating accurate digital terrain models that is useful in various industries such as civil engineering, urban planning, and environmental science.

2.5. DTM (Digital Terrain Model) and DSM (Digital Surface Model)

The Digital Terrain Model (DTM) was introduced by Charles L. Miller in 1958 and widely used in many fields such as cartography, land utilization, urban planning etc. DTM data may be produced using a variety of methods, mostly using airborne light detection and ranging (LiDAR), interferometric synthetic aperture radar (InSAR), and digital photogrammetry based on satellite and aerial photography.

A digital surface model, or DSM, is a representation of a surface, including both natural and artificial structures like trees and buildings. They represent reflected surfaces of all features elevated above the bare ground. In short, DSM represents the Earth's surface and all objects on it, but DTMs depict the bare ground without any trees and building on it.

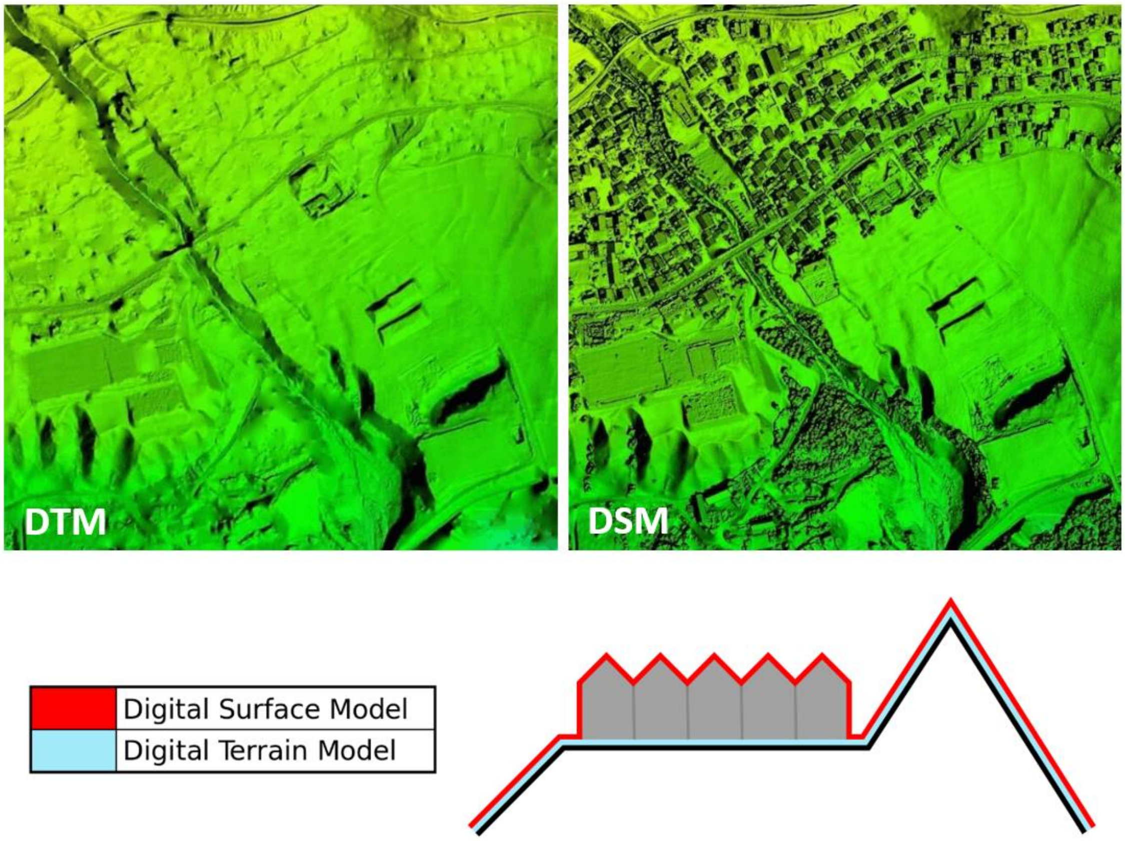


Figure 8. Difference between DTM and DSM

Several terms have been used since the original word was created by Miller in 1958. These include digital elevation model (DEM), digital height model (DHM), digital ground model (DGM) and digital terrain elevation model (DTEM). These terms (DTM, DEM, DHM, and DTEM) are often assumed to be, but sometimes they refer to different products (Li et al., 2005).

3. Methodology

3.1. Study area

For the case study, I decided to use the Denmark airborne laser scanning data. There were several reasons for choosing this data: First, the access[1] to airborne laser scanning data of Denmark is easy and free. Second, it has the points cloud for the whole country, which allows a selection of several areas with various environmental conditions that plays an important role for point cloud classification. Third, the quality and classification accuracy of the LAS files is good enough and reasonable to have them as reference.

3.2. General Workflow

I selected several areas that differed according to the terrain slope and landcover, which are important parameters affecting filtering accuracy and parameters setting. Then, I filtered the ground points using three different algorithms and compared it to the original classification of the point cloud. The last step was to select some of the best results with different parameters and settings and compare the filtering performance of the three selected algorithms. Below is the overall flowchart of the approach used in this thesis (Figure 8).

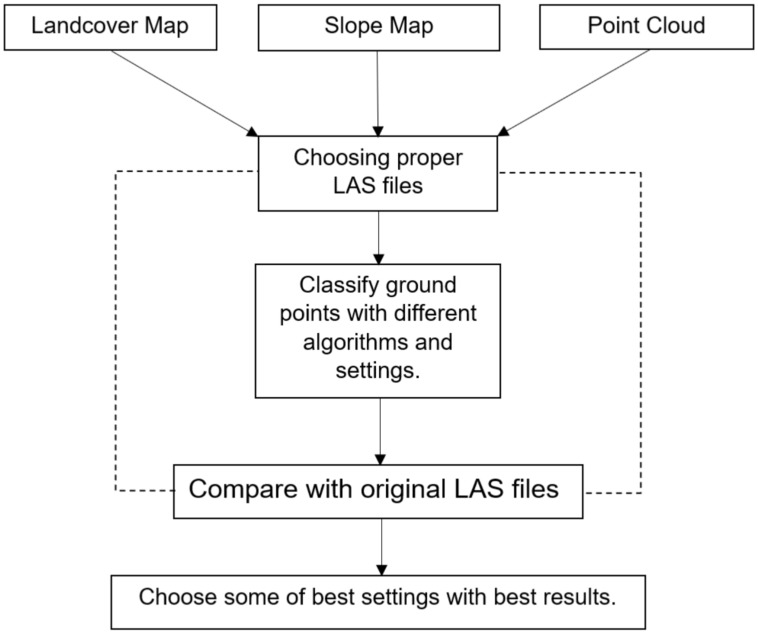

Landcover Map
Slope Map
Point Cloud
Choosing proper LAS files
Classify ground points with different algorithms and settings.
Compare with original LAS files
Choose some of best settings with best results.

Figure 9. Flowchart of the overall approach

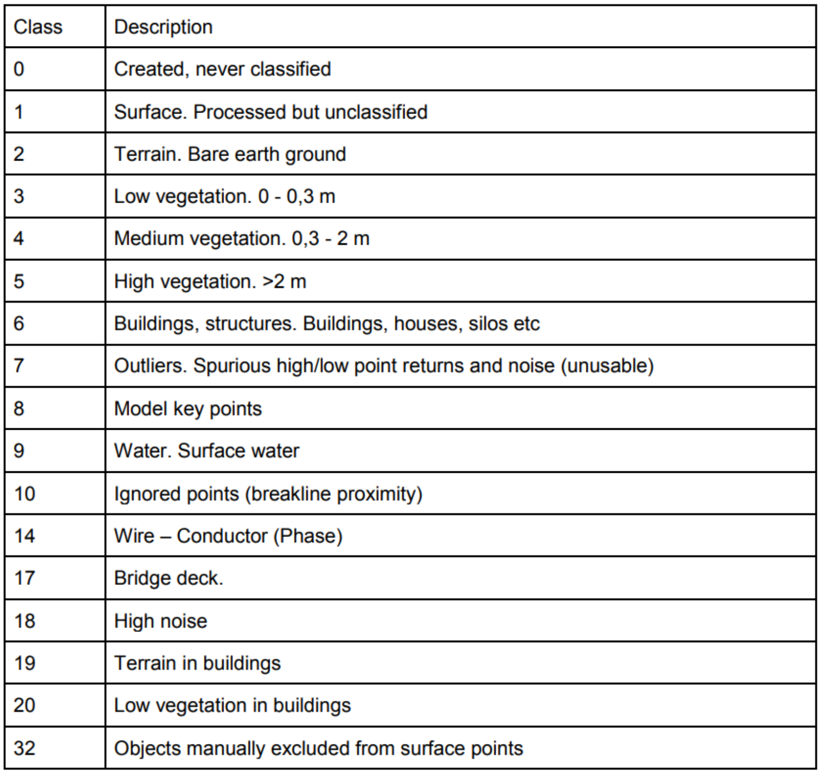
[1] https://dataforsyningen.dk/

3.3. Airborne laser scanning reference data

The point clouds in LAS files have a density of at least 4 points per sq meter. They are classified into the ground (as number 2) and some other non-ground points (e.g., Water, vegetation, buildings). The map of whole country is divided into 1 km by 1 km tiles, that it is the dimension of the LAS files.

The point clouds have been collected from 2018 in 5 years, to 2022 and classified in 11 categories and 17 subcategories (as shown in Table 2). Class 2 as Terrain and Ground is the most important class I used. The points have some other information like Intensity, RGB etc. But as the algorithms that I used in this thesis, just using the geometry of the points (XYZ) this information is not necessary. The coordinate system of the points from the source is UTM32_EUREF89.

Table 2. The classes of Point Cloud from Danish Elevation Model



Class	Description
0	Created, never classified
1	Surface. Processed but unclassified
2	Terrain. Bare earth ground
3	Low vegetation. 0 - 0,3 m
4	Medium vegetation. 0,3 - 2 m
5	High vegetation. >2 m
6	Buildings, structures. Buildings, houses, silos etc
7	Outliers. Spurious high/low point returns and noise (unusable)
8	Model key points
9	Water. Surface water
10	Ignored points (breakline proximity)
14	Wire – Conductor (Phase)
17	Bridge deck.
18	High noise
19	Terrain in buildings
20	Low vegetation in buildings
32	Objects manually excluded from surface points

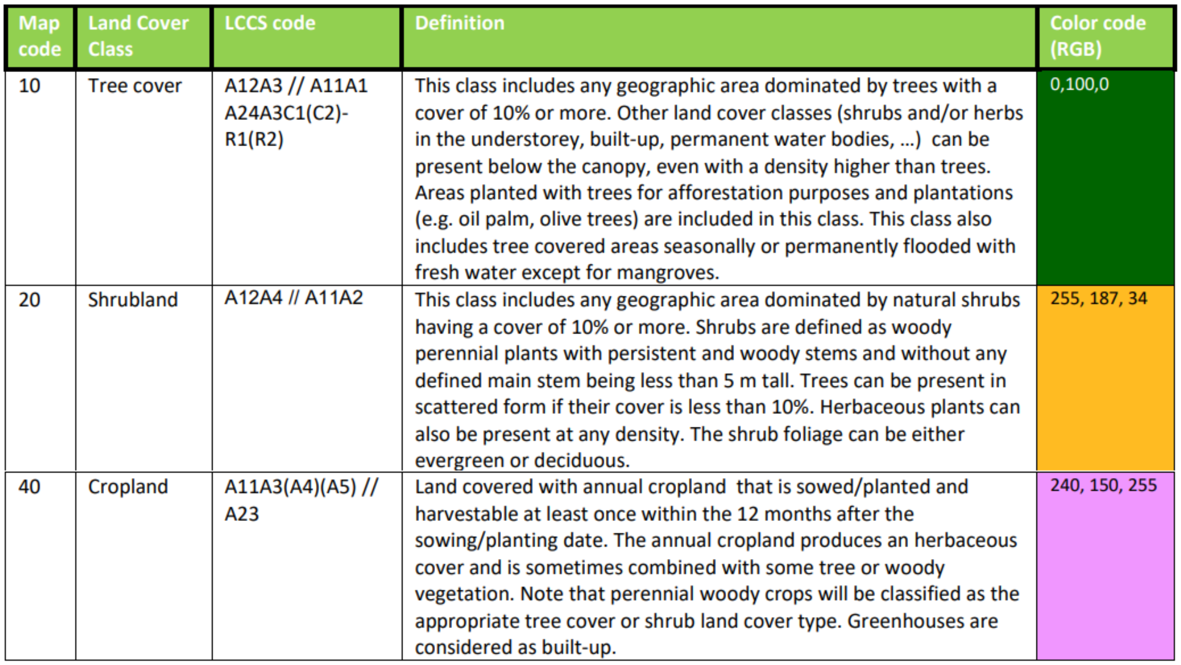
3.4. Landcover and Slope data

For this thesis, I used the ESA Worldwide Landcover [2] map. This map is a freely accessible global land cover product at 10 m resolution and containing 11 land cover classes. This WorldCover map for 2021 was released on 28 October 2022 and resulted in a global overall accuracy of 76.7%.

The discrete classification map provides 11 classes and is defined using the Land Cover Classification System (LCCS) developed by the United Nations (UN) Food and Agriculture Organization (FAO). The UN-LCCS system was designed as a hierarchical classification, which allows adjusting the thematic detail of the legend to the amount of information available.

Below is the description of the three categories I used in this thesis from the Product User Manual of WorldCover.

Table 3. Coding of the Map layer and definition of the classes



Map code	Land Cover Class	LCCS code	Definition	Color code (RGB)
10	Tree cover	A12A3 // A11A1 A24A3C1(C2)-R1(R2)	This class includes any geographic area dominated by trees with a cover of 10% or more. Other land cover classes (shrubs and/or herbs in the understorey, built-up, permanent water bodies, …) can be present below the canopy, even with a density higher than trees. Areas planted with trees for afforestation purposes and plantations (e.g. oil palm, olive trees) are included in this class. This class also includes tree covered areas seasonally or permanently flooded with fresh water except for mangroves.	0,100,0
20	Shrubland	A12A4 // A11A2	This class includes any geographic area dominated by natural shrubs having a cover of 10% or more. Shrubs are defined as woody perennial plants with persistent and woody stems and without any defined main stem being less than 5 m tall. Trees can be present in scattered form if their cover is less than 10%. Herbaceous plants can also be present at any density. The shrub foliage can be either evergreen or deciduous.	255, 187, 34
40	Cropland	A11A3(A4)(A5) // A23	Land covered with annual cropland that is sowed/planted and harvestable at least once within the 12 months after the sowing/planting date. The annual cropland produces an herbaceous cover and is sometimes combined with some tree or woody vegetation. Note that perennial woody crops will be classified as the appropriate tree cover or shrub land cover type. Greenhouses are considered as built-up.	240, 150, 255

I cropped the GIS (.shp) file of the Worldcover by the Denmark map. Denmark map has just 8 categories as can be seen in Figure 10. The colors of the layers are from the standard ESA Worldwide Landcover.

[2] https://esa-worldcover.org/

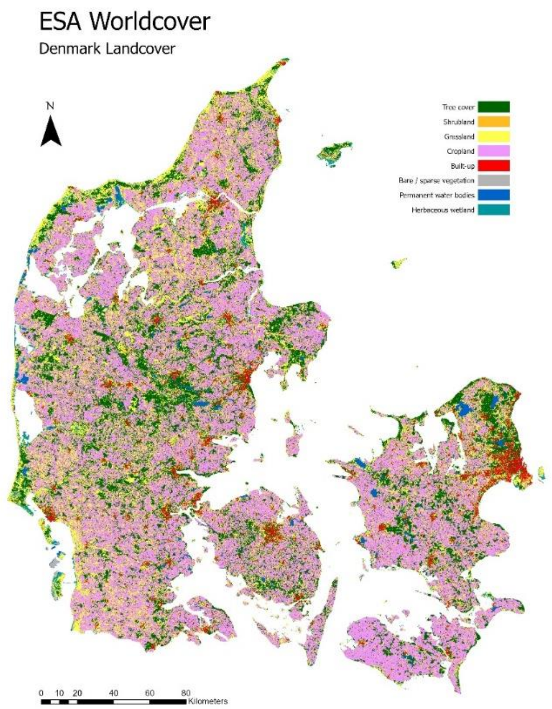


Figure 10. Landcover of Denmark (from ESA Worldcover)

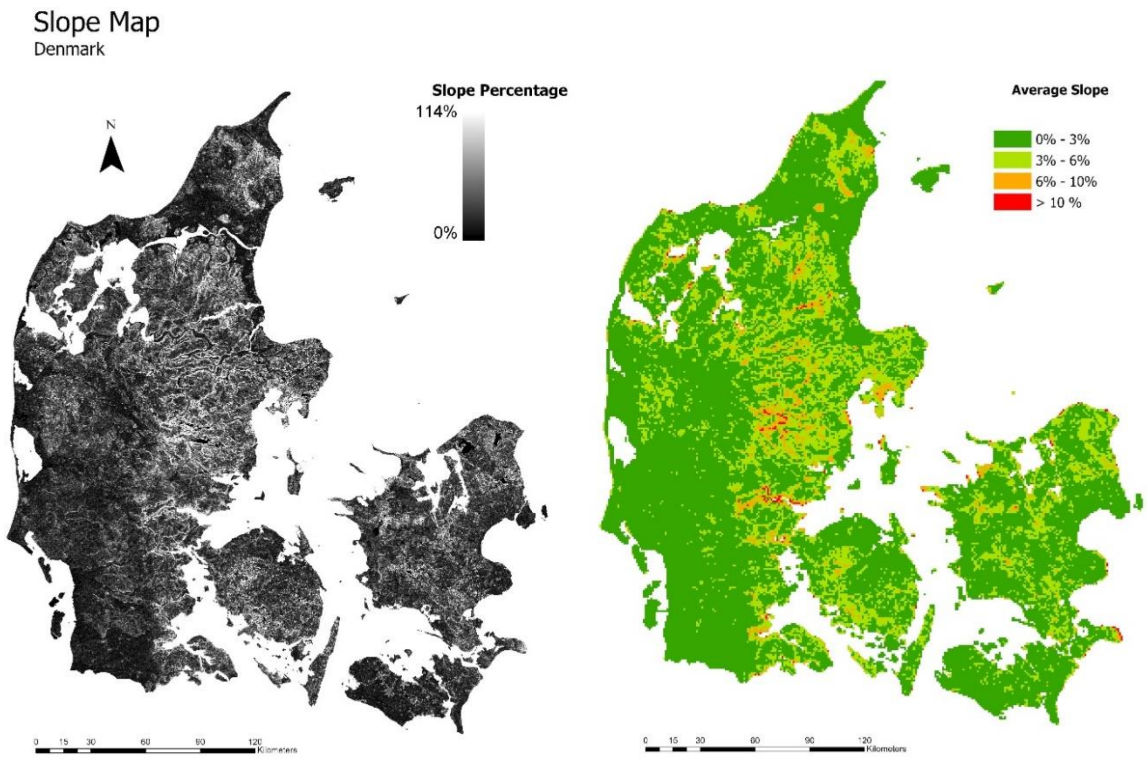


Figure 11. Slope of Denmark (a) General Slope (b) Average in every 1km by 1km tile

Figure 11 shows the map of slope in Denmark. (a) is the general map of slope generated from DTM at resolution of 25 m that downloaded from the same source[3], and the (b) is the average slope that calculated by ArcGIS pro from the DTM, based on the tiles of the point clouds. (1 km by 1 km tiles).

3.5. Selection of LAS tiles

It is known that the algorithms behave differently in different situations and different characteristics of the landcover, and slope will influence directly the results and choosing the proper parameters (Moudrý et al., 2020). Therefore, to choose the point clouds under different environmental conditions, I selected three different landcovers that have different properties in vegetation structure to investigate the behavior of the algorithms in different situations. I used three land cover classes: Tree Cover, Grassland and Cropland. The second category was terrain slope. Since the Denmark is almost a flat country and there are not so many lands on slopes, I decided to categorize the slope in relatively small ranges. I used three categories for slope in this thesis: 0% to 3%, 3% to 6% and 6% to 10%.

The LAS files from Denmark, are distributed in form of 1km x 1km tiles (see Figure 12). From the Figure 12 map(d) I chose the blue tile that it covers mostly with the specifications that required.

[3] https://dataforsyningen.dk/

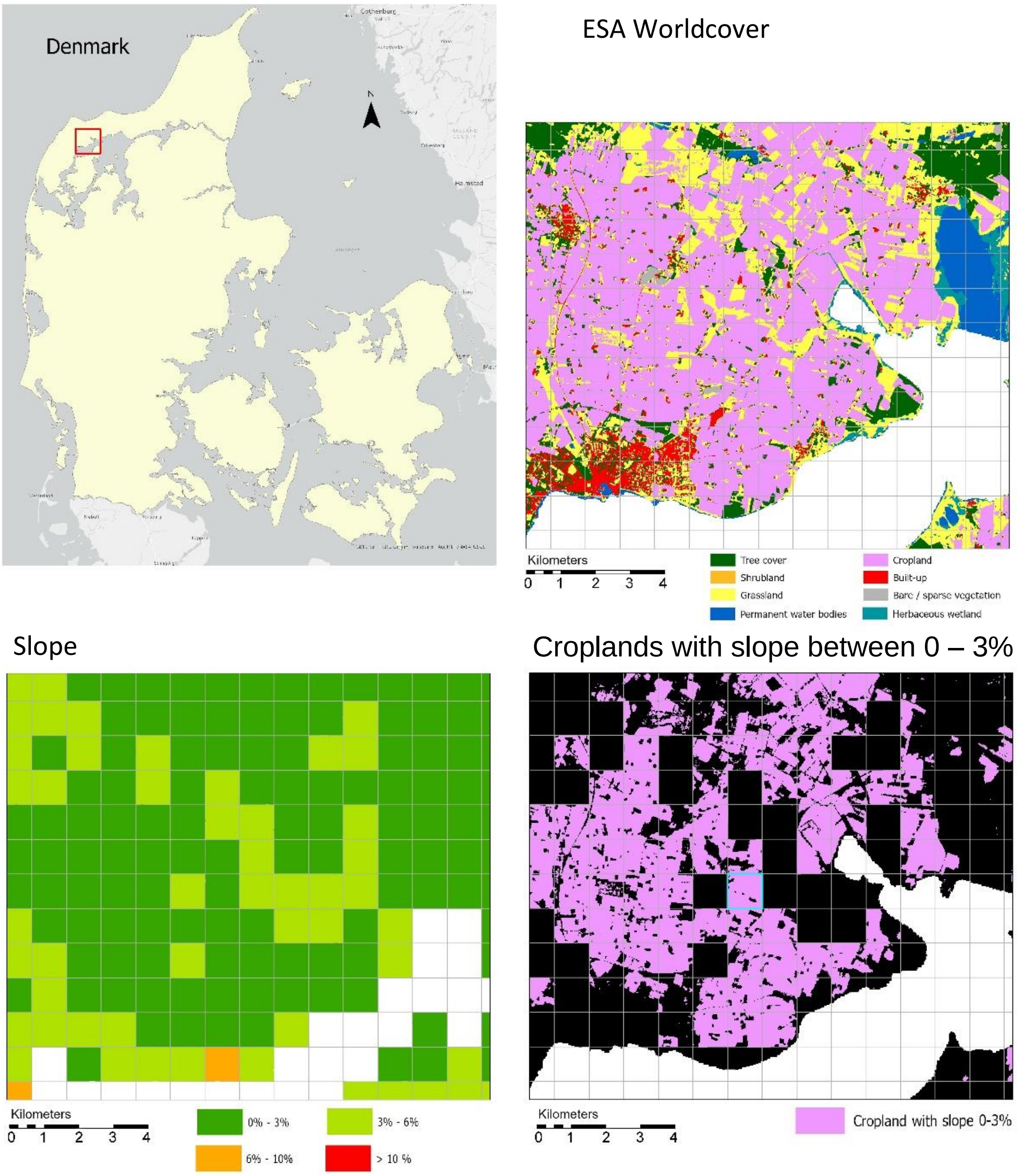


Figure 12. Map of Denmark (a). Map (b) shows the ESA landcover of the red rectangle in map (a). Map (c) shows the average slope in individual tiles of LAS files. Map (d) is the croplands with slope between 0 – 3%.

With the combination of slope and landcover together, I have 9 categories. I decided to choose two tiles for every category (Sample I and Sample II), to make sure about the results and to see if the algorithms working somehow similar in both same category tile.

For making map (c) in Figure 12. I used Zonal Statistics command, and as input for Zone Data, I used the 1 km by 1 km tiles of LAS file tiles (just the boundary of LAS files as .shp file) and for the Input Value Raster, I put the slope raster that I calculated from the DTM. And for the Statistics Type, I used MEAN, because I needed the average of slope in one tile, so that further I could choose and categorize them in different slopes.

Table 4. Samples of point clouds (LAS) that are used in this thesis.

Category	**Cropland**					
Slope	**Slope < 3%**		**3% < Slope < 6%**		**6% < Slope < 10%**	
Sample	**Sample I**	**Sample II**	**Sample I**	**Sample II**	**Sample I**	**Sample II**
Tile No.[4]	6316_485	6340_548	6304_565	6316_563	6313_561	6315_557
No. of Points	7,128,123	7,970,781	6,748,606	7,174,162	9,202,681	8,612,837
No. Ground Points	6,991,078	7,253,171	6,436,635	6,815,301	7,195,849	7,417,146
Percent of Ground Points	98.08	91	95.38	95	78.19	86.12
Average Percent of Ground points	90.63					
Average No. of Points per square meter	**7.13**	**7.97**	**6.75**	**7.17**	**9.2**	**8.6**
Category	**Grassland**					
Slope	**Slope < 3%**		**3% < Slope < 6%**		**6% < Slope < 10%**	
Sample	**Sample I**	**Sample II**	**Sample I**	**Sample II**	**Sample I**	**Sample II**
Tile No.	6343_548	6389_581	6307_522	6308_527	6360_588	6377_561
No. of Points	7,463,955	6,875,758	8,161,414	8,156,491	9,174,177	8,547,748
No. Ground Points	7,016,277	6,156,494	7,348,671	7,245,149	6,004,680	6,675,736
Percent of Ground Points	94	89.54	90.05	88.83	65.45	78.1
Average Percent of Ground points	84.33					
Average No. of Points per square meter	**7.46**	**6.88**	**8.16**	**8.16**	**9.17**	**8.55**
Category	**Tree Cover**					
Slope	**Slope < 3%**		**3% < Slope < 6%**		**6% < Slope < 10%**	
Sample	**Sample I**	**Sample II**	**Sample I**	**Sample II**	**Sample I**	**Sample II**
Tile No.	6338_533	6382_583	6294_553	6296_554	6340_577	6345_573
No. of Points	11,418,310	16,151,478	17,702,374	19,298,021	15,704,242	19,068,163
No. Ground Points	3,728,394	4,665,878	5,027,719	4,078,629	4,335,998	3,809,375
Percent of Ground Points	32.65	28.89	28.4	21.13	27.61	19.98
Average Percent of Ground points	26.44					
Average No. of Points per square meter	**11.42**	**16.15**	**17.7**	**19.3**	**15.7**	**19.07**

[4] It is downloadable from “https://dataforsyningen.dk/” with this number.

3.6. Ground Filtering algorithms

For this thesis I used R for coding and used lidR package. This package has three algorithms for classify ground points: PMF[5], CSF[6] and MCC[7].

3.6.1. Progressive Morphological Filter (PMF)

Progressive Morphological Filter is a method for classifying the ground points by gradually increasing the window size of the filter and using elevation difference thresholds (K. Zhang et al., 2003). In lidR package, PMF has just initially two parameters to set. Window size in filtering ground returns (*ws*) and Threshold heights above the parameterized ground surface to be considered a ground return (*th*). Both parameters should be positive and could be one number or a sequence.

There are two ways for setting these parameters. First, setting them manually, second, using the *util_makeZhangParam()* function. This function makes the sequence for *ws* and *th* parameters based on some initial parameters that has itself (K. Zhang et al., 2003) .I just tried to change 3 (out of 6) parameters. The parameters that I used in loop for coding are:

dh_0 that is e initial elevation difference threshold,
dh_max that is maximum elevation difference threshold,
and *s* that is slope.

For *b* I just used 2 based on the default value mentioned in the paper (K. Zhang et al., 2003). The second reason was in my primary investigations, it did not have so much influence on the results like other parameters.

For *max_ws*, that is maximum window size, I used constant 15. This parameter limits the sequence of *ws* to not making window for filtering larger than this.

exp parameter is for choosing that increasement of the window size is linearly (0 or FALSE) or exponential (1 or TRUE) that I constantly chose exponential increasement for this thesis.

3.6.2. Cloth Simulation Filtering (CSF)

Cloth Simulation Filtering is an algorithm that works like simulating a piece of cloth draped over a reversed point cloud (W. Zhang et al., 2016). In this method the point cloud is flipped upside down, and then a cloth is placed on the inverted surface. The interactions between the cloth nodes and the inverted surface are used to determine ground points. In lidR package, this filtering has 6 parameters to set.

The parameters that I used as constant are, *iteration, sloop_smooth* and *rigidness*. I set *iteration* to 500, because after around 100 the average height variation is so low and after

[5] Progressive Morphological Filter

[6] Cloth Simulation Filter

[7] Multiscale Curvature Classification

150 maximum height variation is around 0.0033 m (W. Zhang et al., 2016). Therefore, 500 is a trustable value for *iteration*. *sloop_smooth* parameter is used when the area is so steep, due to the slopes of the case study, I set it as FALSE. *rigidness* is the rigidness of the cloth that the default value is 1 so I kept it.

The 3 parameters (out of 6) that I used to change their value during the loop in the code are:

class_threshold, the distance to the simulated cloth to classify a point cloud into ground and non-ground.

cloth_resolution, the distance between particles in the cloth. This is usually set to the average distance of the points in the point cloud.

And *time_step*, that it is the time step when simulating the cloth under gravity.

3.6.3. Multiscale Curvature Classification (MCC)

Multiscale Curvature Classification is an automated approach for classifying LiDAR returns that incorporates the strengths of curvature filtering, adds a scale component in the interpolation phase, and a variable curvature tolerance to account for slope interaction with the LiDAR measurements (Evans & Hudak, 2007).

This algorithm has 2 parameters to set that I changed them during the loop in code: Scale parameter as *s*. The optimal scale parameter is a function of the scale of the objects (e.g., trees) on the ground, and the sampling interval (post spacing) of the LiDAR data, and Curvature threshold as *t*.

3.7. Range of tested parameters of ground filtering algorithms

The first step was to choose the approximate amount for the parameters. There are some helpful papers about it that I will describe in the next parts. I wrote three codes for these three different algorithms. The code that I wrote has some loop inside loop that gradually increase the parameters with specific steps.

Due to the duration of calculations (for example 720 times of running the classification in just one file out of 18 for PMF algorithm, it takes about 26 hours) I needed to choose the range of the parameters more precisely. Therefore, I ran the algorithms with different values on a cropped area (150 m by 150 m) from the specific category for faster results to obtain a better value of range and steps. And then with these ranges, I ran the codes on the 1 km x 1 km tiles for each category. Tables 5-7 show the different range of the parameters I used in PMF, CSF and MCC algorithms.

Table 5. Range of input parameters for PMF algorithm

Category	*dh_0*	*dh_max*	*slope*
Tree cover	0.05 – 0.27	0.5 - 4	0.2 – 1.2
Cropland	0.05 – 0.19	0.5 - 4	0.2 – 1.2
Grassland	0.05 – 0.19	0.5 - 4	0.2 – 1.2

Table 6. Range of input parameters for CSF algorithm

Category	*class_thershold*	*cloth_resolution*	*time_step*
Tree cover	0.4 – 3.1	0.4 – 1.8	0.5 – 1.2
Cropland	0.1 – 1.9	1 – 8.5	0.6 – 3.8
Grassland	0.1 – 1.9	1 – 8.5	0.6 – 3.8

Table 7. Range of input parameters for MCC algorithm

Category	*s*	*t*
Tree cover	0.1 - 2	0.1 – 1.5
Cropland	0.1 - 2	0.05 – 0.5
Grassland	0.1 - 2	0.05 – 0.5

3.8. Validation of filtering algorithms

The classified point clouds were compared with the original classification of reference data to quantify the performance of individual filtering methods. I calculated the Type I error (omission error), Type II error (commission error) and Success rate. Type I error, representing the percentage of ground points that are incorrectly classified as non-ground as:

$$Type\ I\ error = \frac{b}{a+b}$$

Where *a* is the number of correctly classified ground points and *b* is the number of ground points misclassified as non-ground points.

Type II error representing non-ground points incorrectly classified as ground points was calculated as:

$$Type\ II\ error = \frac{c}{c+d}$$

Where *c* represents the number of non-ground points misclassified as ground points and *d* stands for the number of correctly classified non-ground points.

Success rate is the ratio between the number of correctly classified points and the total number of points.

$$Success\ rate = \frac{a+d}{e}$$

Where *e* stands for total number of all points.

3.9. Coding with R-Studio

lidR library was used for testing the ground filtering algorithms. RCSF library is essential for CSF algorithm and RMCC for MCC algorithm. Below is an example of code for MCC algorithm that I used in this thesis. There are two loops inside each other that gradually increase the *s* and *t* parameters for using as the input for *classify_ground()* function. And after every classification, the Type I and II error and Success rate is calculated and stored in a table.

```r
## loading necessary libraries
library(lidR)
library(RCSF)
library(RMCC)

laz_path     <- "...\\Slope )6 Tree\\6340_577.laz"
csv_path_MCC <- "...\\Slope )6 Tree\\6340_577(MCC).csv"

# Create table for result
table <- data.frame()

# loading reference LAS or LAZ file
las_ref <- readLAS(laz_path, select = “xyz”)

# Classification of points in reference file
classification_ref <- las_ref$Classification

s <- 0;
for(k in 1:20){
  # step for s parameter
  s <- s + 0.1;
  t <- 0;
  for(j in 1:15){

    #starting time
    time1 <- Sys.time()

    #step for t parameter
    t <- t + 0.1;

    # Ground filtering by specific algorithm
    las_filtering <- classify_ground(las_ref, mcc(s, t))

    # Classification of points in filtered file
    classification_filter <- las_filtering$Classification
    a <- 0
    b <- 0
    c <- 0
    d <- 0


     for(i in 1:length(classification_ref)){

      if(classification_filter[i] == 2){
        if(classification_ref[i] == 2){
          a <- a + 1
        }
      }

      if(classification_ref[i] == 2){
        if(classification_filter[i] != 2){
          b <- b + 1
        }
      }

      if(classification_ref[i] != 2){
        if(classification_filter[i] == 2){
          c <- c + 1
        }
        if(classification_filter[i] != 2){
          d <- d + 1
        }
      }
    }

    type_I_error <- b/(a + b)
    type_II_error <- c/(c + d)
    success_rate <- (a + d)/length(classification_ref)

    # ending time
    time2 <- Sys.time();
    # calculating the time of calculation
    cal_time <- difftime(time2, time1, units = "mins");

    # put results in an array
    new_raw <- c(s, t, type_I_error, type_II_error, success_rate,
cal_time)

    # add new row of results to  result table
    table <- rbind(table, new_raw)
  }
}
# insert names of the columns
names(table) = c("s", "t", "Type I error", "Type II error","Success
rate", "Calculation time (min)")
# writing the table in a .csv file
write.csv(table, csv_path_MCC, row.names = FALSE)
```

3.10. Choosing the best parameters.

The method I used for choosing the best parameters was that, first I sorted the results by the Type II error, from smallest amount. And then by scrolling down, I found a point that Type II is increasing so much, around that point, I was searching for some highest Success Rate and finding the less amount of Type I error. I chose 2 or 3 rows (gray rows). After this, I checked these parameters in the other sample and chose the best of them among two samples (Gray bold row).

Table 8. Choosing the best parameter in the result.

Loop	**Class threshold**	**Cloth resolution**	**Time step**	*Rigidness*	*Iterations*	*Sloop smooth*	**Type I error**	**Type II error**	**Success rate**
1	1.3	0.4	0.5	1	500	0	99.94	0	71.13
2	1.3	0.6	0.5	1	500	0	99.94	0	71.13
3	1.3	0.8	0.5	1	500	0	99.94	0	71.13
4	1.3	1	0.5	1	500	0	99.94	0	71.13
. . .	. . .	. . .	. . .	. . .	. . .	. . .	. . .	. . .	. . .
77	0.4	1	0.5	1	500	0	6.42	3.98	95.32
78	0.4	1.8	0.6	1	500	0	7.15	4.04	95.06
79	**0.4**	**0.8**	**0.5**	**1**	**500**	**0**	**5.28**	**4.09**	**95.56**
80	0.4	1.6	0.6	1	500	0	6.63	4.1	95.17
81	0.4	1.4	0.6	1	500	0	5.99	4.15	95.32
. . .	. . .	. . .	. . .	. . .	. . .	. . .	. . .	. . .	. . .
638	3.1	0.6	1.2	1	500	0	0	12.23	91.3
639	3.1	0.4	1.1	1	500	0	0	12.26	91.28
640	3.1	0.4	1.2	1	500	0	0	12.35	91.21

3.11. Creating DTM

For visualizing the result with chosen parameters, I tried to make a DTM from classified point cloud and compare it with the original DTM from website. For generating DTM from point cloud, I used rasterizing command with TIN algorithm.

```
DTM <- rasterize_terrain(las_file, algorithm = tin())
```

4. Results

4.1. Effects of parameters setting on Type I and Type II error

In total, 9 classifications were performed (Table 9-11). Different settings produce different Type I and Type II errors even within individual algorithms. Here I illustrate the effects of parameters setting only for the Tree Cover category, which had the highest effect on the results (Figures 13 – 15).

4.1.1. MCC

The mean Type I error typically decreased with increasing parameter *t* as well as with parameter *s* but only up to approx. 0.7 when it was constant or slightly increased depending on the slope of the terrain. On the other hand, the mean Type II error increased with parameter *t*, while parameter *s* had a minimal effect. The best balance between the Type I and Type II error is for parameter *s* around 0.8 and for parameter *t* it depends on the terrain slope and ranges between 0.3 to 1.3. Surprisingly, the Type II error tends to be higher in flat terrain than on the steeper slopes.

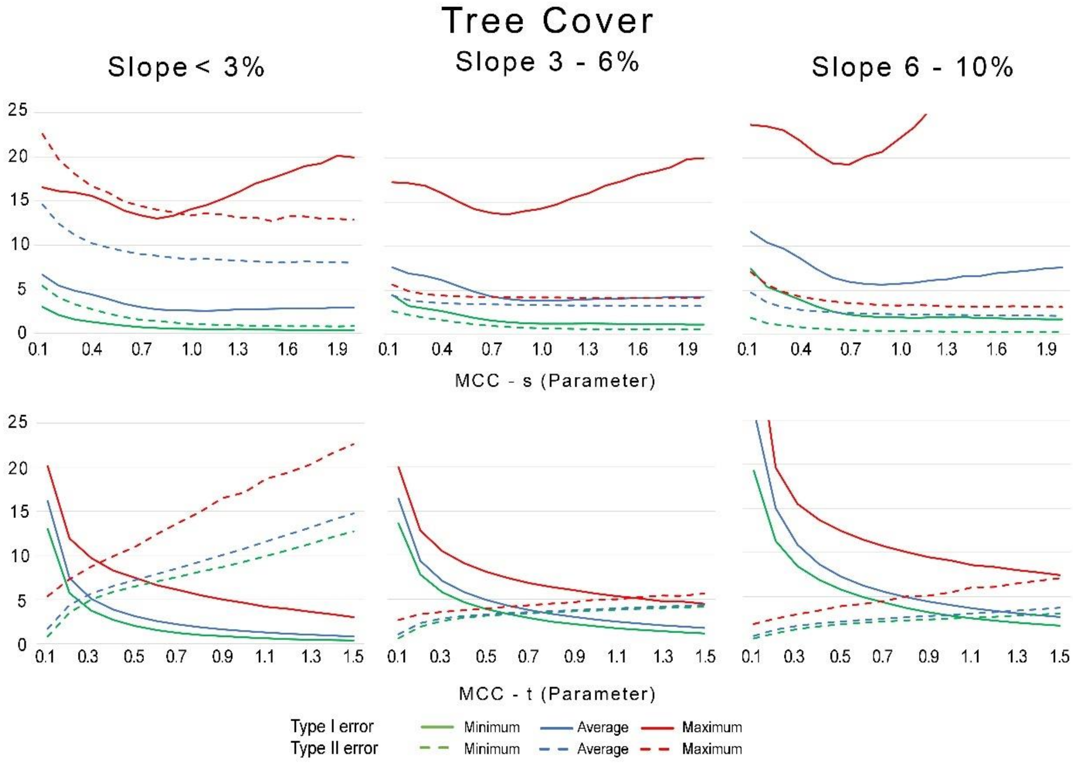


Figure 13. Result of MCC algorithm in Tree Cover category for both s and t parameters. Lines show the Type I error and dash lines show the Type II error. For example, for *t = 0.2* we have Maximum (red), Average (blue) and Minimum (green) of the errors. It is because of that, for *t = 0.2*, we have different amount of *s* that leads to different results and could be considered in *s* parameter chart.

4.1.2. CSF

All the 3 parameters that I changed in the loop in coding play a significant role in the algorithm. All the parameters are so sensitive to the slope, and as the slope increases, the range of Type I error gets higher and higher.

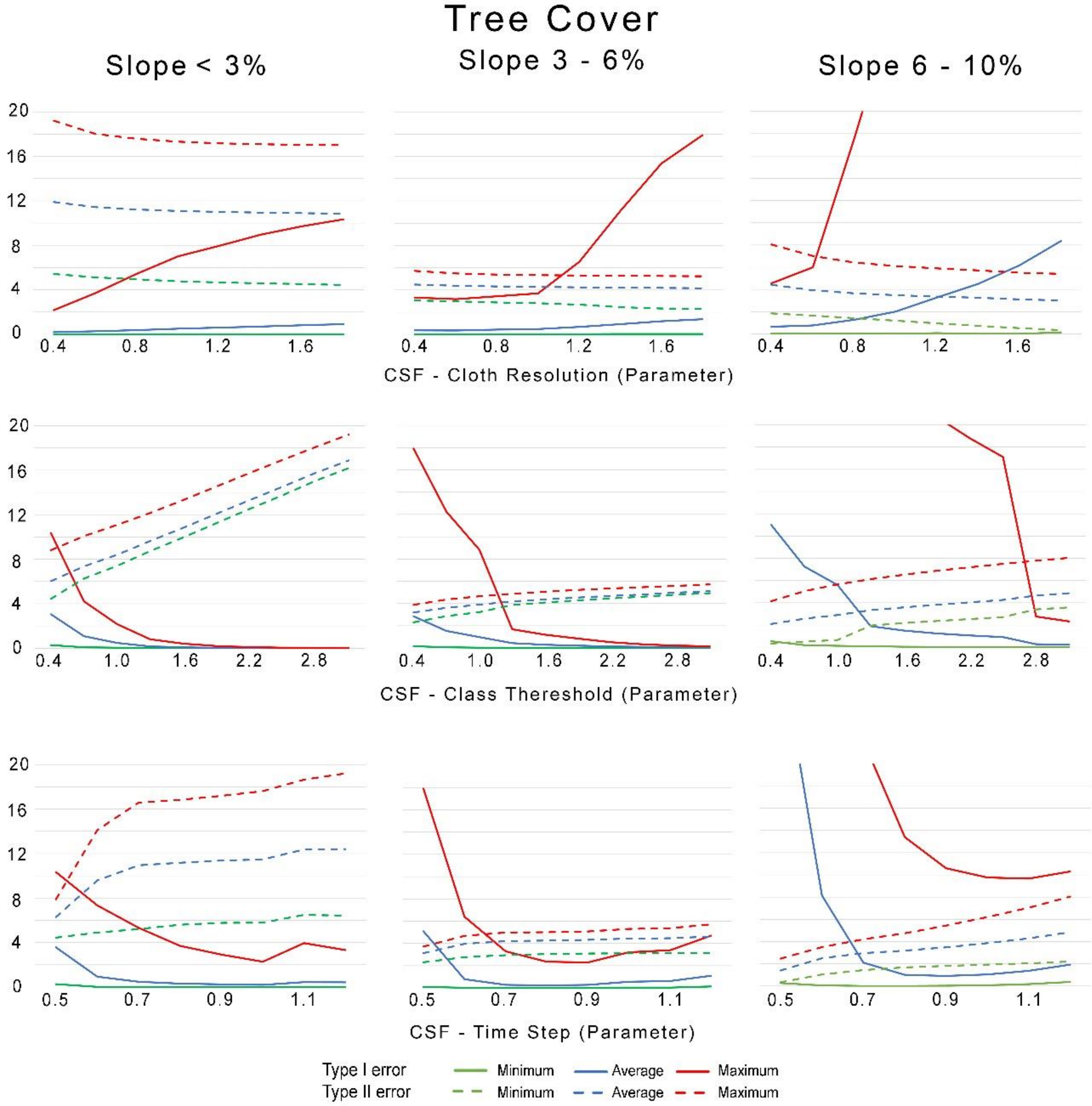


Figure 14. Result of CSF algorithm in Tree Cover category for different parameters

Time Step, less than 0.5 - 0.6 seems to have a vast range of errors. In general, Type II error is higher in the flatter area. For the average of Type I and II errors, the balance point between them increases for *Time Step*, meanwhile for *Class Threshold* and *Cloth Resolution*, this point decreases by the slope.

By increasing *Class Threshold,* the Type I error decreases, but Type II error increases and vice versa for the *Cloth Resolution.* In slope less than 3 (almost flat area) mostly the whole range of the Type II error is significantly more than Type I error, but in the fields with more slope, there is a goof balance between them.

4.1.3. PMF

The main parameter and the one has more influence on the result in PMF algorithm is the d_h0. As shown in Figure 15 the range of the result (min, max and avg) is not so much and it shows that different parameters do not play a significant role in this algorithm.

The balance point between two errors in *d_h0* decreases by increasing the slope. The range of the Type II error (space between min and max) is almost zero, and it shows that Type II error is just changing by the *d_h0* parameter and other parameters has no effect on it.

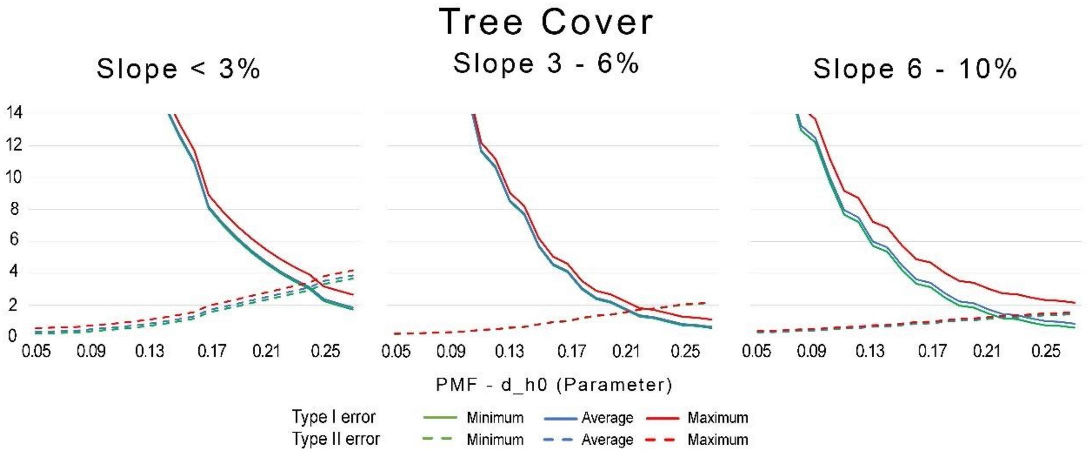


Figure 15. Result of PMF algorithm in Tree Cover category for d_h0 parameter

4.2. Chosen parameters

Among so many results, I chose one series of parameters with the best results. The chosen parameters and the result of selecting them in all categories are in Tables 9-11.

Table 9. Chosen parameters for MCC algorithm.

Landcover	Slope	s	t	Type I error	Type II error	Success rate
Cropland	< 3%	1	0.1	4.03	3.78	95.99
	3 - 6%	0.6	0.05	7.18	8.56	92.75
	6 - 10%	1	0.1	1.58	1.21	98.47
Grassland	< 3%	1	0.1	2.14	1.41	97.90
	3 - 6%	1.6	0.1	4.73	3.18	95.42
	6 - 10%	1	0.2	1.34	2.38	98.43
Tree Cover	< 3%	1.5	0.2	6.80	3.32	95.68
	3 - 6%	2	1	2.94	3.00	97.01
	6 - 10%	2	1.2	2.62	1.77	98.06

Table 10. Chosen parameters for CSF algorithm.

Landcover	Slope	class threshold	cloth resolution	time step	*sloop smooth*	*Rigidness*	*Iterations*	Type I error	Type II error	Success rate
Cropland	< 3%	0.1	1	1	*0*	*1*	*500*	3.80	6.05	96.00
	3 - 6%	0.1	3	0.6	*0*	*1*	*500*	6.22	20.08	93.09
	6 - 10%	0.1	1.5	3	*0*	*1*	*500*	2.58	2.76	97.40
Grassland	< 3%	0.1	1.5	2.6	*0*	*1*	*500*	2.97	2.06	97.08
	3 - 6%	0.1	1.5	1	*0*	*1*	*500*	4.68	4.74	95.31
	6 - 10%	0.3	1.5	1	*0*	*1*	*500*	2.23	2.80	97.65
Tree Cover	< 3%	0.4	0.8	0.5	*0*	*1*	*500*	5.28	4.09	95.56
	3 - 6%	0.4	1.2	0.6	*0*	*1*	*500*	2.81	2.59	97.36
	6 - 10%	0.4	0.8	1.1	*0*	*1*	*500*	2.74	1.55	98.21

Table 11. Chosen parameters for PMF algorithm.

Landcover	Slope	dh_0	dh_max	Slope	*Max Window Size*	*Exp*	*b*	Type I error	Type II error	Success rate
Cropland	< 3%	0.11	0.5	0.2	*15*	*1*	*2*	0.58	5.3	99.32
	3 - 6%	0.09	0.5	0.2	*15*	*1*	*2*	9.16	6.92	90.94
	6 - 10%	0.18	2	0.4	*15*	*1*	*2*	0.3	1.81	99.48
Grassland	< 3%	0.15	2	0.4	*15*	*1*	*2*	1.13	1.63	98.83
	3 - 6%	0.15	2	0.4	*15*	*1*	*2*	3.12	3.21	96.87
	6 - 10%	0.15	2	1	*15*	*1*	*2*	0.9	1.76	98.91
Tree Cover	< 3%	0.23	2	0.2	*15*	*1*	*2*	1.98	2.41	97.71
	3 - 6%	0.21	2	0.4	*15*	*1*	*2*	1.71	1.54	98.41
	6 - 10%	0.24	2	0.6	*15*	*1*	*2*	0.77	0.92	99.11

Figure 16 shows the Type I and II error and success rate of the chosen parameters. In Cropland and Grassland categories, the slope 3 – 6 % seems was trickier for all algorithms and due to that, they have less success rate than other slopes. By increasing the slope, the Type I and Type II error decreases and due to that, success rate increases smoothly.

Highest Type I and II errors are in the Cropland category. The Tree Cover category totally has a better result and the highest success rates. In all categories PMF has less Type I error than other algorithms and except one category (Cropland, 3 - 6 %) it has the highest success rate too.

Table 12. Average Result of different algorithms

Landcover	Algorithm	Type I error	Avg.	Type II error	Avg.	Success rate	Avg.
Cropland	MCC	4.26	3.94	4.52	6.27	95.74	95.94
	CSF	4.20		9.63		95.50	
	PMF	3.35		4.68		96.58	
Grassland	MCC	2.74	2.58	2.32	2.58	97.25	97.38
	CSF	3.29		3.20		96.68	
	PMF	1.72		2.20		98.20	
Tree Cover	MCC	4.12	3.07	2.70	2.35	96.92	97.46
	CSF	3.61		2.74		97.05	
	PMF	1.49		1.62		98.41	

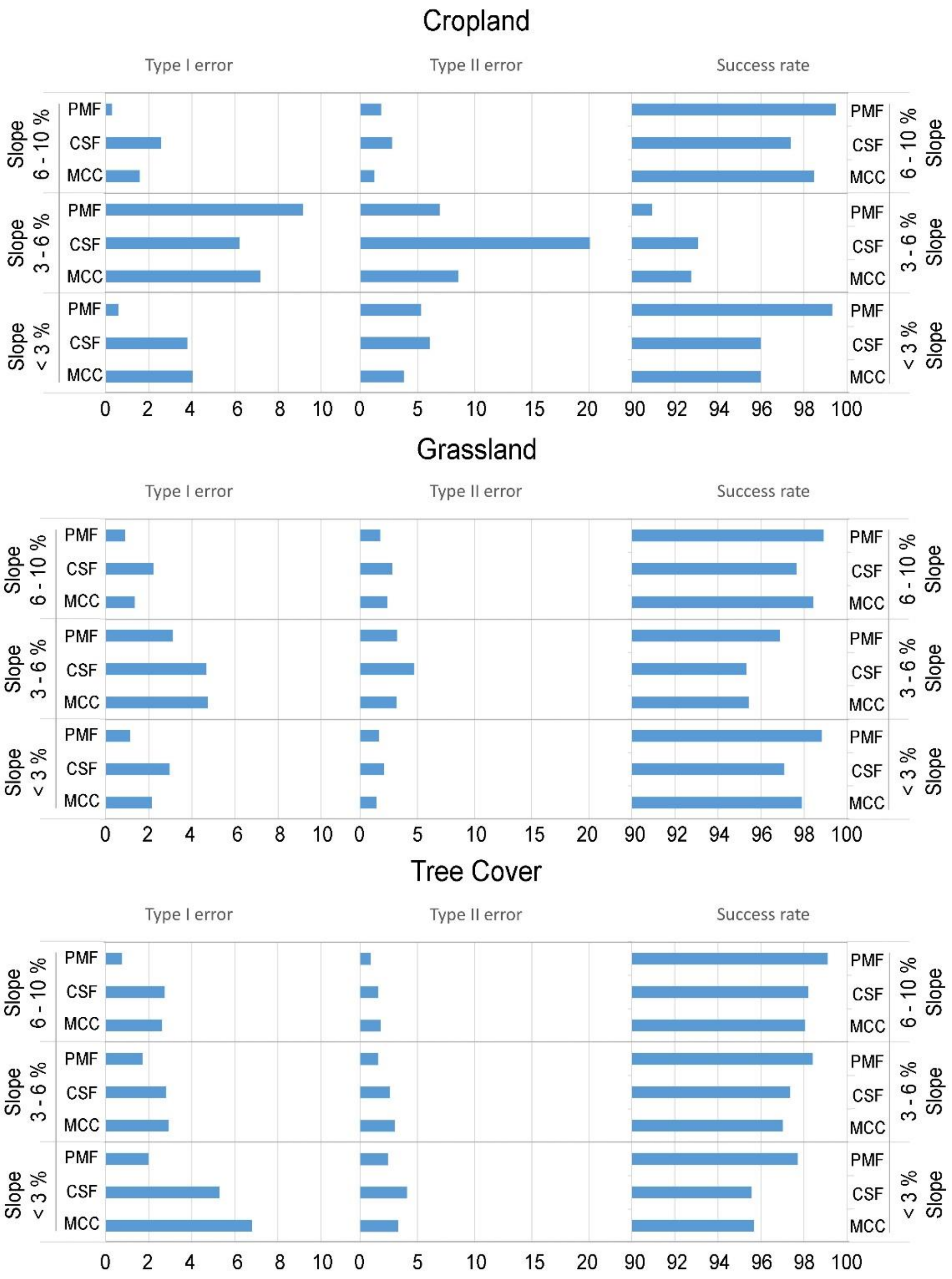


Figure 16. The chart of the chosen parameters.

Totally in average, the Tree Cover category has better average success rate than other categories. Cropland landcover has the highest and Tree cover has the lowest Type II error. Grassland has almost equal Type I and II error and Tree cover has a lower Type II error than Type I error.

4.3. Sections

Figure 17 shows the same section in different algorithms. I tried to find a section with some trees that canopy would be visible.

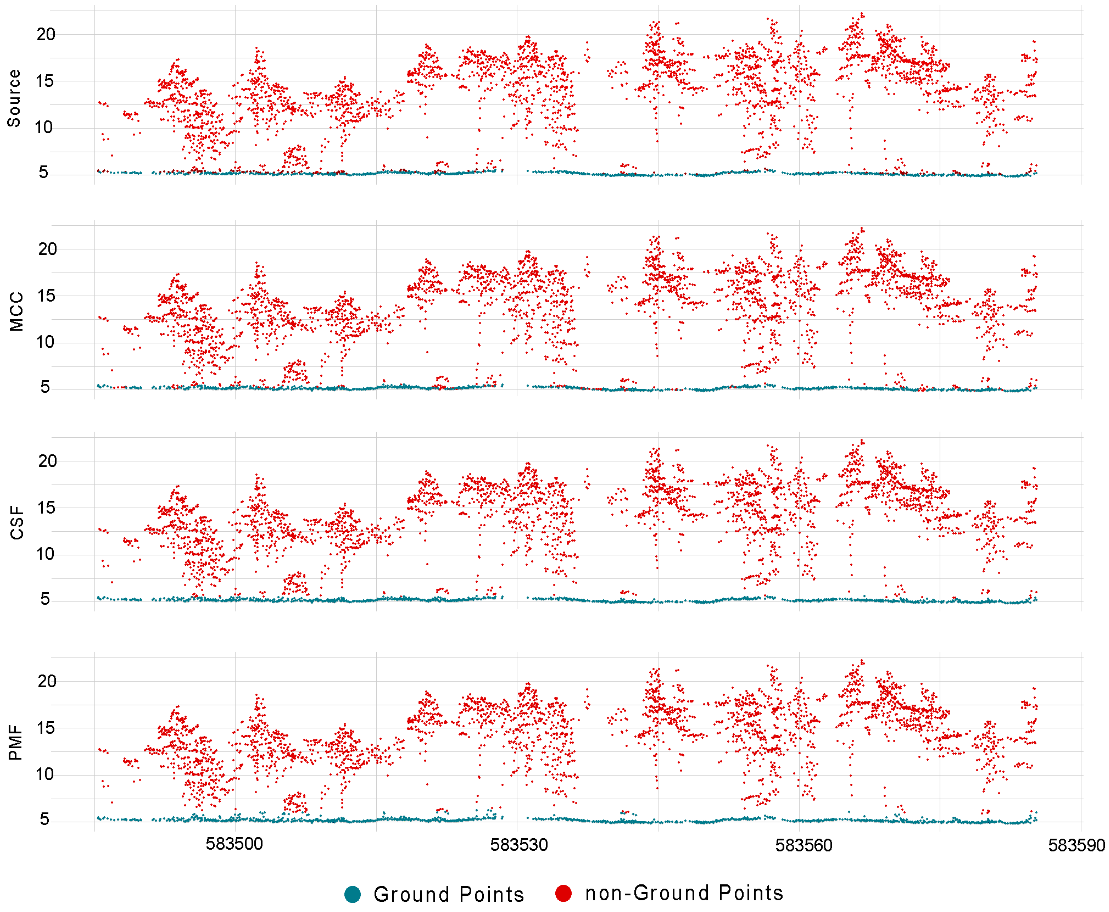


Figure 17. Section on Point Cloud in different algorithms (Tree Cover category in slope < 3%. The section is 100m in length in the center of the tile.)

In these sections (Figure 17), MCC algorithm defines some ground points as non-ground (Type I error). Meanwhile, seems there are some inaccuracies in the source too. PMF defines some points that are not ground as ground point (Type II error), they seem that they are some bushes or plants. In general (in these sections) CSF shows a better performance than others.

The interesting thing is that none of the algorithms has the significant error of defining a non-ground point as ground point. The highest one is for PMF and it is less than 2 meters.

4.4. DTM

I made the DTM for same part of a tile from the classified ground points with different algorithms for Tree Cover category and slope < 3%.

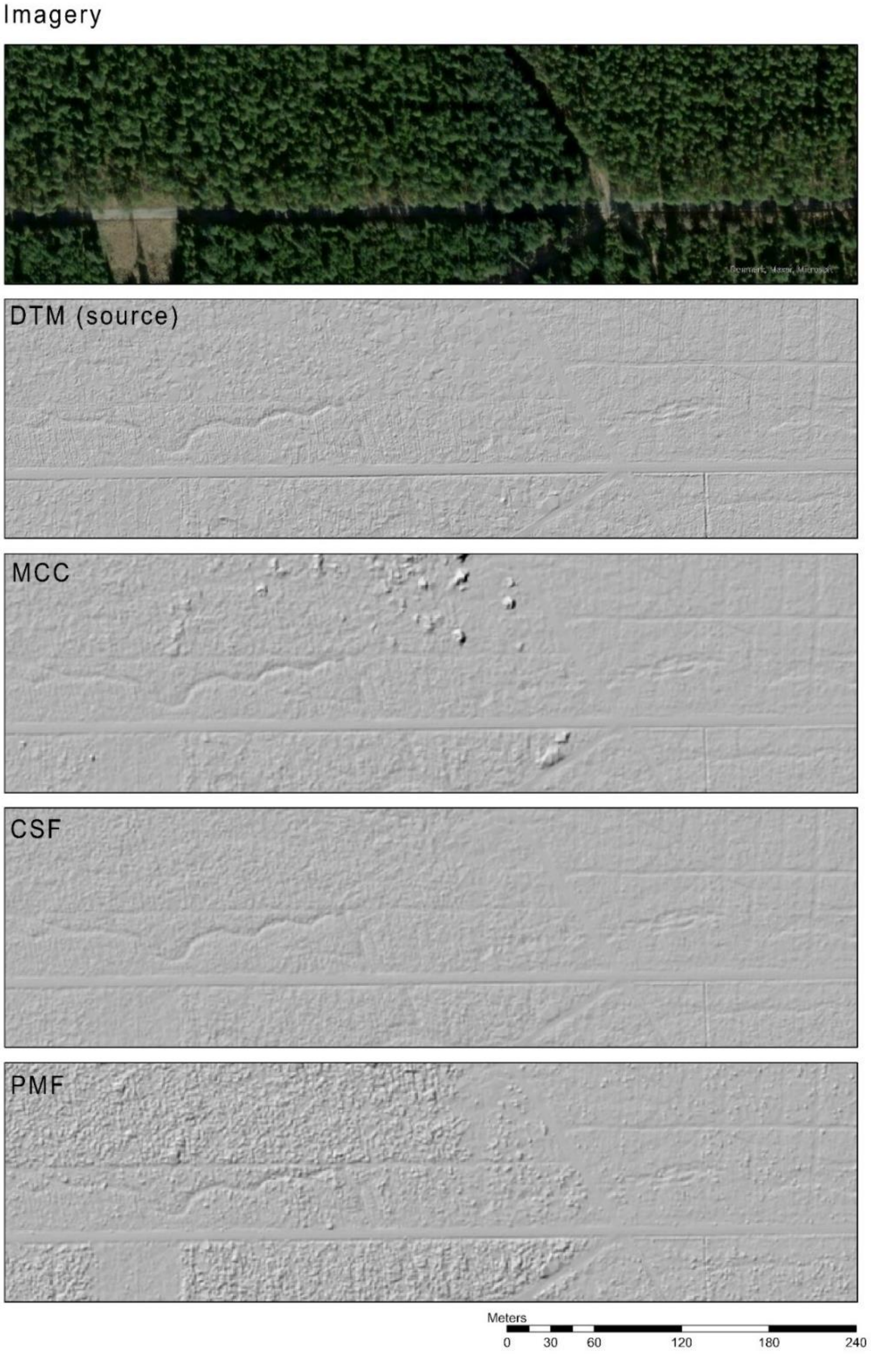


Figure 18. DTM of Tree Cover category created from point cloud with different algorithms.

MCC has so many extra bumps in the result than the source DTM, and PMF and CSF define them much better. There are some small details in the source DTM that PMF shows it but seems it is more exaggerated. And in general, CSF has a better result with no unexpected bumps, but in comparison with PMF it has less details. Surprisingly, All the three algorithms define the roads in the tile so clear and well. The interesting thing is that there is one winding road (or stream) has appeared in the DTMs that is not visible in aerial imagery.

4.5. Time of calculation

After I ran the codes, the result for every category (9 categories) was a table. The running time of the calculation was about 54[8] days. Below is a brief table of the average of running codes in different algorithms. Due to running codes in different computers, all the calculation times converted to a unique computer specification[9].

Table 13. Average of Calculation Time for one loop (Minute)

Landcover	CSF	MCC	PMF
Cropland	0.092	3.48	3.8
Grassland	0.112	3.12	3.77
Tree Cover	0.65	8.45	2.68

I just want to mention here that the huge difference of calculation time in Tree Cover for CSF and MCC may be is the number of the points in this category that is almost more than two times of other categories. But, in PMF it shows different behavior. In general, MCC has the most calculation time than other algorithms. This time of calculation is not tangible while using for just one or two calculations, but for calculating so many times in loop, it makes a huge difference for approaching the result.

[8] The process ran on different computers and in parallel.

[9] Core i5 – 3570K CPU @ 3.4GHz, 16 GB Physical Memory (RAM)

5. Discussion

In this thesis, I compared the performance of three algorithms for ground filtering of LiDAR. All the three algorithms showed a good and acceptable performance, but for a user that do not have experience in parameters setting and their influence, MCC may be a good algorithm to start with it due to its low number of input parameters (*s* and *t*). I used the three free accessible algorithms that were available in R studio in lidR package. The biggest obstacle in this study was calculation time that limited me to go further for evaluating other algorithms. Both vegetation and slope has influence on the results and other papers agree with this too (Klápště et al., 2020). It seems the Grassland and Cropland, due to the less height of vegetation is trickier for the algorithms.

In all categories by increasing the slope, the errors decreases and success rate increases, and this result is in contrast with the result with the work of (Moudrý et al., 2020). But the point is that in the range between 0 to 10% slope that I worked on, in this paper has a very limit change in errors, and the critical slope defined 15% in their paper that I did not have that in my case study. In general, the success rate is lower in the low vegetation landcovers, but the Tree Cover had higher success rate and this conclusion complies with paper of (Moudrý et al., 2020).

The reason that the Tree Cover category has the better result maybe is the density of the points that the files in this category have and it is almost more than two times of the other categories. The filtering performance decreases as the point density increases (Serifoglu et al., 2016). All the DTMs that were made by the filtered point cloud had an acceptable quality, but CSF had less errors and bumps and PMF had keep tiny details better.

Further studies that I recommend is as follows:

- Other algorithms from other software like LAStools and ArcGIS could be compared with these ones.
- In this thesis, for CSF and PMF, I just worked on 3 parameters of them. There is a potential of investigating the influence of the other parameters on the algorithms.
- It is possible that after one ground filtering, doing another (second or maybe the third) ground filtering on the ground points. This could be a combination of algorithms too. (Like first try with MCC and second one with CSF on the result of MCC algorithm).
- The density of the points is different in different LAS files. Is the density of the points having the influence on the algorithms? Which parameters are more sensitive to that? Or is there any parameter that behaving dependent of the density of the points?
- I just worked on three landcover categories, there is potential to work on more landcovers with different slopes too.
- The maximum percentage of the slope I used was 10%. It is possible to compare the algorithms in steeper areas too to see their performance.

6. Conclusion

The MCC algorithm's parameters, *s* and *t*, have a significant impact on the Type I and Type II errors. The Type I error typically decreased with increasing parameter t and s but only up to approx. 0.7 when it was constant or slightly increased depending on the slope of the terrain. On the other hand, the mean Type II error increased with parameter *t*, while parameter s had a minimal effect. The best balance between the two errors was found when parameter s was around 0.8, and for parameter *t*, it depended on the terrain slope and ranges between 0.3 to 1.3.

The CSF algorithm's three parameters: *time step*, *class threshold*, and *cloth resolution* are sensitive to slope. The Type II error was higher in flatter areas, and for the average of Type I and II errors, the balance point between them increased for Time Step, but for Class Threshold and Cloth Resolution, this point decreased by the slope. Increasing Class Threshold decreases Type I error but increases Type II error, and vice versa for Cloth Resolution.

The PMF algorithm's parameter *d_h0* has a significant impact on the result, while other parameters have a minimal effect. The balance point between the two errors in *d_h0* decreases by increasing the slope, and the range of the Type II error is almost zero, indicating that Type II error changes only by the *d_h0* parameter.

All these three algorithms for ground filtering, have good performance in defining the ground points. Different types of land cover and slope has influence in the result of the ground filtering, and due to that the input parameters of the algorithms should change to. Increasing slope led to decreased errors and increased success rates, which differed from a previous study. Tree Cover had the highest success rate, possibly due to its higher point density, which can decrease filtering performance.

7. Bibliography and sources

Bandyopadhyay, M., van Aardt, J. A. N., & Cawse-Nicholson, K. (2013). Classification and extraction of trees and buildings from urban scenes using discrete return LiDAR and aerial color imagery. *Laser Radar Technology and Applications XVIII*, *8731*, 873105. https://doi.org/10.1117/12.2015890

Brydegaard, M., Merdasa, A., Gebru, A., Jayaweera, H., & Svanberg, S. (2016). Realistic Instrumentation Platform for Active and Passive Optical Remote sensing. *Applied Spectroscopy*, *70*(2), 372–385. https://doi.org/10.1177/0003702815620564

Chen, Q., Wang, H., Zhang, H., Sun, M., & Liu, X. (2016). A point cloud filtering approach to generating DTMs for steep mountainous areas and adjacent residential areas. *Remote sensing*, *8*(1). https://doi.org/10.3390/rs8010071

Chuvieco, E. (n.d.). *Fundamentals of satellite remote sensing : an environmental approach*.

Dong, P., & Chen, Q. (n.d.). *LiDAR Remote sensing and Applications*.

Evans, J. S., & Hudak, A. T. (2007). A multiscale curvature algorithm for classifying discrete return LiDAR in forested environments. *IEEE Transactions on Geoscience and Remote sensing*, *45*(4), 1029–1038. https://doi.org/10.1109/TGRS.2006.890412

Klápště, P., Fogl, M., Barták, V., Gdulová, K., Urban, R., & Moudrý, V. (2020). Sensitivity analysis of parameters and contrasting performance of ground filtering algorithms with UAV photogrammetry-based and LiDAR point clouds. *International Journal of Digital Earth*, 1672–1694. https://doi.org/10.1080/17538947.2020.1791267

Kraus, K., & Pfeifer, N. (n.d.). *ADVANCED DTM GENERATION FROM LIDAR DATA*.

Kraus Ł, K., & Pfeifer, N. (1998). Determination of terrain models in wooded areas with airborne laser scanner data. In *ISPRS Journal of Photogrammetry & Remote sensing* (Vol. 53).

LAS Specification 1.4-R15 Release Information. (2002).

Li, Z., Zhu, Q., & Gold, C. (2005). *Digital terrain modeling : principles and methodology*. CRC Press.

Liu, X. (2008). Airborne LiDAR for DEM generation: Some critical issues. In *Progress in Physical Geography* (Vol. 32, Issue 1, pp. 31–49). SAGE Publications Ltd. https://doi.org/10.1177/0309133308089496

McManamon, P. F. (2019). *LiDAR Technologies and Systems*. SPIE Press. https://books.google.cz/books?id=cIIIwAEACAAJ

Moudrý, V., Klápště, P., Fogl, M., Gdulová, K., Barták, V., & Urban, R. (2020). Assessment of LiDAR ground filtering algorithms for determining ground surface of non-natural terrain overgrown with forest and steppe vegetation. *Measurement:*

Journal of the International Measurement Confederation, *150*. https://doi.org/10.1016/j.measurement.2019.107047

Priestnall, G., Jaafar, J., & Duncan, A. (n.d.). *Extracting urban features from LiDAR digital surface models*. www.elsevier.com/locate/compenvurbsys

Schmid, K., Carter, J., Waters, K., Betzhold, L., Hadley, B., Mataosky, R., & Halleran, J. (2012). *LiDAR 101: An Introduction to LiDAR Technology, Data, and Applications*. www.csc.noaa.gov

Serifoglu, C., Gungor, O., & Yilmaz, V. (2016). Performance evaluation of different ground filtering algorithms for uav-based point clouds. *International Archives of the Photogrammetry, Remote sensing and Spatial Information Sciences - ISPRS Archives*, *2016-January*, 245–251. https://doi.org/10.5194/isprsarchives-XLI-B1-245-2016

Vosselman, G. (2000). Slope based filtering of laser altimetry data LoD2 Building Modelling by Roof Surface Topologies View project SLOPE BASED FILTERING OF LASER ALTIMETRY DATA. In *IAPRS: Vol. XXXIII*. https://www.researchgate.net/publication/228719860

Wang, C., & Glenn, N. F. (2009). Integrating LiDAR intensity and elevation data for terrain characterization in a forested area. *IEEE Geoscience and Remote sensing Letters*, *6*(3), 463–466. https://doi.org/10.1109/LGRS.2009.2016986

Zhang, K., Chen, S. C., Whitman, D., Shyu, M. L., Yan, J., & Zhang, C. (2003). A progressive morphological filter for removing nonground measurements from airborne LIDAR data. *IEEE Transactions on Geoscience and Remote sensing*, *41*(4 PART I), 872–882. https://doi.org/10.1109/TGRS.2003.810682

Zhang, K., & Whitman, D. (2005). *Comparison of Three Algorithms for Filtering Airborne LiDAR Data*. http://www.

Zhang, W., Qi, J., Wan, P., Wang, H., Xie, D., Wang, X., & Yan, G. (2016). An easy-to-use airborne LiDAR data filtering method based on cloth simulation. *Remote sensing*, *8*(6). https://doi.org/10.3390/rs8060501