UiT The Arctic University of Norway

Faculty of Science and Technology, Department of Physics and Technology

Automatic snow layer detection in drone-borne radar data using edge detection and morphology

Martin Markus Løthman Blengsli

Master's Thesis in Energy, Climate, and Environment - EOM-3901 – December 2021

Abstract

Several studies and techniques exist in automatically detecting objects and layers in ground penetrating radar (GPR) data and its closely related field of seismic data. However, a technique that specifically can detect the primary interfaces in GPR data collected from an airborne drone over snow-covered terrain remains to be accomplished. Finding a suitable automatic technique could be faster and more accurate than manual interpretation and could save money and manpower. This thesis tries to detect the top and bottom layer of the snow-pack by using Canny edge detection and morphological operators on filtered GPR data. As a result, a root-mean-square-error deviation within the range resolution of the radar system was achieved on the top layer, tested on representative subsection, excluding outliers and signal dropouts. On the other hand, the bottom layer struggles to be close to the ground truth but has a cross-correlation value above 0.9, with outliers in complex weak bottom environments. With Canny edge detection and morphological operators, we could reliably detect the top layer in data with a distinct return signal. However, the method struggled with the bottom layer but consistently followed the layer with a varying degree of separation.

Acknowledgements

To my friends, family and classmates who have followed me through five years of education, thanks.

Special thanks to my supervisor Dr. Anthony Doulgeris and co-supervise Dr. Rolf-Ole Jenssen which have gone above and beyond in making this thesis what it is. Thanks again to Jenssen which provided me with the data and the master thesis.

Contents

List of Figures

List of Tables

List of Abbreviations

CFAR constant false alarm rate

CNN convolutional neural network

DEM digital elevation models

EM electromagnetic

GPR ground penetrating radar

GPS global positioning system

LOESS locally estimated scatterplot smoothing

Norce Norwegian Research Centre

RF radio frequency

RMSE root mean square error

SE structuring element

SVD singular value decomposition

SWE snow water equivalent

Selected Symbols

$\cap$	Intersection
$\cup$	Union
$\bullet$	Closing
$\circ$	Opening
$\ast$	Convolution
μ	Mean
$\ominus$	Erosion
$\oplus$	Dilation
σ	Variance
$\sum$	Summation
$\times$	Times
η	N-by-M local neighbourhood
ν	Noise variance
b	Wiener approximation
H	Hermitian transpose

1

Introduction

The concept of radar is familiar for most people, where it is used to detect the location and the range of, e.g., a plane or ship. However, electromagnetic waves can also penetrate certain materials, allowing imaging of the substructure. This process of utilising the returning signal from an underlying structure is called ground penetrating radar (GPR). The name reflects the first geological usage for the method, but applications now range from sand and concrete to ice and snow.

Like radar, GPR works by transmitting electromagnetic waves and recording the reflected waves. However, the difference is that the signal of interest is not primarily the first reflection of a layer but the reflection within the medium of interest. A representation of the subsurface structure can be evaluated from this returning signal from the recorded wave. In our case, the medium of interest is snow; this primarily influences the signal's range where the transmitted signal encounters a more extensive diversity of mediums than in air. Factors that influence the returning signal are the electromagnetic properties of the mediums, the shape of the interfaces, the distance the wave has travelled, and more. The technique is non-invasive, so there is no need to displace material for a survey to be undertaken. This property is advantageous in instances where a mass removal would lead to excess work, like locating pipes in an urban environment or measuring the thickness of a glacier by drilling. GPR could also be used in earth science, where it is used to detect and study bedrock, aquifers, and subsurface layers. It is also used in archaeology to detect patterns that could indicate the presence of earlier human activity like settlements or

artefacts without damaging them.

The process of gathering GPR data can be done in multiple ways, depending on the goal and need for accuracy. It can be gathered by having the antenna coupled to the medium of interest or with an air gap from the medium. The antenna could be fixed on a rolling platform with separation in order of tens of cm or even with tens of meters of separation on a drone [1, 2] or plane [3, 4]. Both methods have their advantages and disadvantages where the most suitable method must be chosen depending on the goal of the survey space. In short, having the antenna directly coupled to the medium increases the penetration depth and accuracy but limits the total surveying area compared to airborne surveys. Conversely, if there is a gap between the antenna and the medium, the opposite is true regarding coupling to the medium. It is characterised by having a lower resolution and penetration depth but does benefit from observing patterns and trends from a higher perspective. Additionally, the penetration and resolution of the returning signal depend on the wave's frequency. The waves have an inverse relationship, where a low-frequency wave has high penetration but lower resolution, and vice versa.

With the great utility of GPR, there exist some difficulties which stall use and implementation of it in other fields. Amidst its problems, there is the complexity of interpretation of the wave and the interplay between the GPR wave and the electromagnetic (EM) properties of the medium of interest. For example, materials with considerable conductivity increase attenuation, which decrease signal strength. Water content presents the same problem with a high permittivity, which further increases the attenuation and severely limits the range of the signal. A medium with these elements and sediments of different sizes could render the GPR data challenging to interpret. Under favourable circumstances, the ideal GPR environment would be one at which the mediums are dry and are separated with mediums of homogeneous sediment sizes.

In addition to the problem of heterogeneous layers of different EM properties, GPR images are also heavily corrupted with noise of various sorts. First, noise poses a challenge because it corrupts the desired image and makes it harder to interpret. It blurs the distinction between the area of interest and the background, making the interpretation harder for humans and other detection methods.

GPR images, like regular images, have a presence of white noise across the dataset, which introduces distortions of various intensity. In addition to white noise, GPR data can also contain noise from other radio frequency (RF) sources nearby, as well as clutter from multipath waves and crosstalk (direct wave). This direct wave clutter expresses itself horizontally in the data and is usually bright compared to the other signals, but it decreases with distance. Another

kind of noise is narrowband noise, which is an artefact from other RF sources, such as telemetry links, and is often manifested as vertical lines in the radar image. Even though all noise poses challenges, filter parameter optimisation can minimise its impact. Therefore, optimisation will be an essential part of the study's data analysis.

Depending on the data, a trained person can detect buried objects or sparse layer boundaries because of humans' pattern recognition and interpolation ability if signals are weak or missing. However, due to the noise mentioned in the previous paragraph and variation in mediums, it has been challenging to automate the interpretation of the GPR data to any form of image processing technique. This problem is because of a lack of processing techniques that are targeted towards the task of detecting lines in GPR data and the accuracy of preexisting methods. As a result of a lack of interpretation tools, analysing the data has previously been done manually. Due to an increasing amount of GPR data, the problem has boiled down to the lack of manpower and the time it takes a person to analyse and interpret the incoming data. Therefore, it would be of great interest in speeding up the process of interpreting the image by automating or partly automating the process with the help of computers. Finding a suitable method for detection of layers in GPR data could be faster, cheaper, more consistent, and potentially more accurate than humans. Depending on the implementation of the method, it could also enable real-time processing and interpretation.

1.1 Background

The complex abilities of human vision and how we perceive the world around us became evident when replicating these abilities with computers. When this was first attempted in the 1970s and 80s, the pioneering techniques that tried to map the world through computers settled on the idea that detecting edges was an excellent way to start perceiving the world. The decision was based on the edges often marking the boundary from one object to the next. In addition, edges could be marked with a change in various image characteristics like colour, intensity, and texture.

The first pioneering method was the Sobel edge detector [5] which works by finding the gradient in an image by convolving the image with the Sobel operator, which worked as the first derivative. Areas with a significant intensity gradient, usually an edge, will be marked with a new corresponding value depending on the change in intensity. However, basing our edge detection on the first derivative can result in some edges being small and some thick in size, depending on the nature of the edge. Irregular edge thickness and using the

first derivative can make it more challenging to interpret the image, and it is also sensitive to noise. One way to solve the problem of noise and thick edges is to use the Marr-Hildreth or also known as the zero-crossing edge detection method [6] which is one of the other pioneering methods. The method works by smoothing the image with a Gaussian kernel and then convolving it with the Laplacian operator, where the Laplacian works as a second derivative. Following from the Laplacian, areas that show a change in intensity will be noted with a zero-crossing in the second derivative, hence the name. This zero-crossing in the second derivative will be marked as an edge. The method makes edges that gradually increase in intensity over multiple pixels marked as only a single-pixel wide compared to the Sobel edge detector.

Even with smoothing from the Gaussian kernel, the zero-crossing method is still sensitive to noise due to the effect of deriving a second time. Therefore it has been suggested to filter the image with the Gaussian and keep it to the first derivative using the Sobel operator. The edges' strength and direction are also noted with the Sobel operator. This is the start of the Canny edge detection [7] method, where a non-maximum suppression is applied to trim the edges, and a directional aware hysteresis is applied to isolate the most robust edges. Canny is one of the edge detection systems with the most extended longevity and is still in use today.

As a result of the lower computational capability of the age computers, these early techniques followed what is called a model-based edge detection system. This approach uses mathematical, physical, or biological models to derive an edge detector applied to the whole image [8]. However, this can lead to problems because of the variety, complexity of colour, intensity and texture in an image.

In helping with the interpretation and the processing of the resulting image, analysing the image using mathematical morphology could be applied. Mathematical morphology follows a practical approach where it processes objects in an input binary image based on their shape. The objects or pixels of interest are the ones and zeroes, defined as foreground and background. Operations that can be applied to the objects are growth or shrinking of an image region, called dilation and erosion, respectively. The number of pixels added or removed from an image depends on the size and shape of the structuring element (SE) used. The SE is an operator of a chosen shape based on the task and intention of the given morphological operator. Other frequent morphological operations are opening, which removes sharp edges, and closing, which smooths edges.

In the decades to follow and an increase in computing power in tandem with Moore's law, a new method that uses information theory to detect edges under previous difficult conditions was introduced. This new method utilises the

availability of a vast amount of data and the techniques introduced by machine learning. Because of the use of a large amount of data, the new method is said to be data-driven [8].

Under this new approach, we find statistical edge detection [9] which uses pre-segmented images to learn the probability distributions on and off the edges. This technique only looks at the intensity value in the image to determine the location of edges. Building on the idea of statistical edge detection, there is also the possibility to combine it with the change including both colour and texture [10] to increase the detection of edges further. The relationship between the intensity, colour and texture in the image is essential to get the most accurate interpretation of the image. In [11] these three parameters were put into a globalised framework which was based on spectral clustering to achieve state-of-the-art performance in the ability to detect edges.

As a subcategory under data-driven design on edge detection, learning-based systems are introduced, including machine learning. These learning-based systems outperform model-based approaches in their accuracy, but they need a large amount of annotated data for training. Moreover, the selection of training data makes it prone to error if presented with an imaging scenario in which it has not been trained [8]. Under learning-based systems, we have, amongst others, holistically-nested edge detection [12] which performs an image to image prediction based on the training from a convolutional neural network (CNN). In Cunha, et al., 2020, [13] a CNN that has been trained on a synthetic dataset, is used to detect seismic faults in real-life data using transfer learning. The transfer learning method uses knowledge acquired from a previously trained model and adapts it to a similar learning task by tuning a few hyper-parameters [14].

There are already some GPR methods that automatically detect features in GPR images. For example, there is the model-based method from [15] which successfully detected layers in the pavement automatically. However, that was on an experimental pavement site under favourable conditions. Nevertheless, newer detection tools usually gravitate towards a learning-based method with a focus on CNNs because of its accuracy. This shift can be seen amongst others in [16] which have found a way to detect moisture damages in asphalt, and multiple papers on automatically detecting rebars [17, 18] both by using a CNN.

In the methods stated above, all are specified towards image processing of the given data, but none are about the collection of data in the first place. However, data collection is essential since insufficient raw data could make cascading effects down the processing line, resulting in a poorer result. One way to improve the resulting data could be to use the constant false alarm

rate adaptive algorithm, which could improve the base data used for detection based on statistical properties in the data [19]. It does it by increasing and lowering a threshold value on a local level to maintain a constant probability of false alarm during the acquisition of the data [20]. Changing the threshold on a local level bypasses the problem where a low threshold will increase the probability of noise, but a high threshold will reduce the wanted signal.

It is essential to point out that in the methods presented thus far, there has been a lack of papers that have as the main focus of detecting inconsistent and interfaces in noisy GPR data and the problems they entail. Therefore, problems that revolve around exactly this problem will be further examined in this thesis.

1.2 Motivation

In this paper, our task would be to find an edge detection technique that can detect the primary interfaces in a GPR image. The GPR dataset used in this thesis has been collected and provided by Norwegian Research Centre (NORCE) and shows the data from an airborne drone flying over snow-covered terrain. The primary interfaces are the top layer which shows the transition between air and snow, and the bottom layer, which marks the transition between snow and ground. There are also interval layers between these two primary layers that represent the snow stratigraphy. Finding a suitable edge detection technique will be done by implementing the techniques and concepts mentioned in the previous section. A focus will be on the early model-based methods. However, expanding into more complex edge detection models will be needed if the former method is not sufficiently feasible and accurate. As alluded to earlier, if a suitable technique is developed, it will help NORCE and others to handle a more considerable amount of GPR data without being limited by manual interpretation.

In finding a suitable layer detection method, the overall objective will be divided into three sub-goals: detecting the top layer, detecting the bottom layer, and quality check of the results. Separating the task of detecting the top and bottom layer is needed not only because they are two distinct layers but because they have significantly different characteristics from an image processing perspective. The approach for each layer would be to test out different methods on a small but representative subset of the dataset of interest. Each method tested will be argued for, why it was tested and the logical reason why it was considered a suitable fit for the task at hand. Once the method is deemed sufficiently accurate against a reliable ground truth, the subset analysed will be gradually increased until it includes the whole dataset. After

detecting both the layers, an attempt will be made to quality check and control the detection's reliability. For each step, notes will be made of what shows an improvement and what does not for further discussion.

It is important to emphasise that finding the top and bottom layer in GPR datasets is the first step in finding the snow depth. Since the material properties of the snow, such as the density and liquid water content, is unknown, it is only possible to approximate the depth from the results of this thesis. Ideally, all parameters, including the two mentioned, would be known to precisely know the propagation velocity, hence the thickness.

1.3 Objective

The overall objective of this master thesis is to develop a layer detection method that can accurately detect air-snow and snow-ground interfaces in a GPR image. With a precise prediction of the position of the layers found through this thesis, knowledge of the density and water content of the snow, in-situ or estimated, an estimate of snow depth and SWE can be given. Being robust, this could be a valuable tool for mapping the snow depth over a large area. Mapping the snow depth from an airborne drone would be faster, more accurate and more consistent than manual measurements. The mapping could be beneficial in, for example, predicting meltwater throughout spring and summer for water reservoirs used for both drinking and hydropower. Forecasting the amount of water would be especially useful for the future. It is predicted that the effects of climate change will increase both the periods of rainfall and drought of specific areas, making it more critical to forecast and plan ahead of time.

The overall objective of the thesis will be accomplished by:

- Acquiring a suitable dataset and creating ground truth for later quantitative evaluation.
- Preparing the dataset by reducing noise and enhancing the areas of interest.
- Finding a method that can accurately and robustly detect the uppermost layer. Start with a subset and then increase the area to the whole dataset.
- Do the equivalent procedure for the bottom layer as for the top layer.
- Finding a method to quality check the estimates.

1.4 Structure of Thesis

This thesis is divided into parts: some introductory chapters, Part I Finding the top layer, Part II Finding the bottom layer, and a final overall concluding chapter. In the introductory chapters we have a Introduction provided in chapter 1, with the theoretical background needed in chapter 2 under Theory. A description of the data gathering and the different datasets are given in chapter 3, Datasets. Since the two methods of finding the top and bottom layers have been found to be different, they have been separated and contain the same chapter headings but for each layer. Part I of the thesis addresses the detection of the top layer, where chapter 4, Method Development, presents the preparation of the data and the layer detection. Because of the nature of the thesis the Method Development chapter is written in a way that tries to intertwine the Method and the Result chapters of a regular paper. This is to give the paper a better flow and give the reader a better insight into the thought process of the thesis. Chapter 5, Results, will apply this method to some selected areas from each dataset with their corresponding accuracy and quality. Arguments for improvements and weak points in the method will be presented in chapter 6, Discussion. Part II has the same chapters as part I just for the bottom layer. Lastly, a overall conclusion for the two detection methods is given in chapter 10.

2 Theory

2.1 Canny Edge Detection

Most of this section is based on the description of the Canny edge detection algorithm described in Digital image processing [21]

As mentioned in the introduction, the Canny edge detector has been one of the most popular edge detectors throughout the ages, with lengthy longevity and consistently good accuracy for specific tasks. The algorithm is based around three objectives which try to combine the best things from preexisting methods but also learn from their weaknesses [21]. The objectives are low error rate, single edge point response and accurate edge point localisation. These points are difficult to all be satisfied at the same time, but through the different kinds of mathematical formulations, an attempt is made to maximise each of them.

The first step in the edge detector is to smooth the image by convolving the original image by the first derivative of a Gaussian, see equation 2.1. Smoothing is done to lower the impact of noise, but it also has the drawback of blurring out weak edges. Therefore, smoothing is done by adjusting the standard deviation (σ) value in the Gaussian to find the best balance between the desired reduction of noise and the distortion weak edges. Padding is also implemented to counter

image size reduction due to convolution.

$$f_s(x,y) \;=\; G(x,y) * f(x,y) \tag{2.1}$$

Table 2.1: Parameters in equation 2.1, smoothing by convolution

Parameter	Description
$f_s(x,y)$	Smoothed image
$G(x,y) = e^{-\frac{x^2+y^2}{2\sigma^2}}$	First derivative of a Gaussian
$*$	Convolution
$f(x,y)$	Original image

The magnitude and direction of the smoothed image are calculated using a given kernel. In this thesis, the Sobel kernel is used and is given below in both x- and y-direction. The Sobel operator finds the derivative in both directions by looking at the change in pixel value, emphasising the value closest to the centre pixel, given the value 2. So, if there is a drop or an increase in pixel value in the x- or y-direction, it will be marked with either a high positive or low negative value. On the other hand, if there is an area with relative constant intensity values, the divergent value will be close to zero.

$$G_x = \begin{bmatrix} -1 & 0 & +1 \\ -2 & 0 & +2 \\ -1 & 0 & +1 \end{bmatrix}, G_y = \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ +1 & +2 & +1 \end{bmatrix}$$

The divergent is later used to find the magnitude and the direction of the edges. This is done by using equation 2.3 and 2.2, and is used to find the angle and the strength respectively, of the edge. The strength will later be used when reducing the chance of false edges by hysteresis thresholding.

$$M_s(x,y) = \sqrt{g_x^2(x,y) + g_y^2(x,y)} \tag{2.2}$$

$$\alpha(x,y) = tan^{-1}\left[\frac{g_y(x,y)}{g_x(x,y)}\right] \tag{2.3}$$

In the new gradient magnitude image given by finding the gradient magnitude of every pixel edges are typically displayed by wide ridges. This is a result of

some edges do not have a clear edge, but a linear increase in pixel values. To reduce the width of these edges a nonmaxima suppression is used to lower the width to one pixel. In short the normal of the edges is found using the direction found earlier and divided in to one out of four directions. E.g. if the edge normal is found in between angel -22.5° and 22.5° or -157.5° and 157.5° it will be marked as a horizontal edge. In the same example of a horizontal edge, if the magnitude is less then one or both of its neighbours in the edge normal direction (up or down), it will be given the new value of zero (suppression). If it is higher than both, the value is kept. The same thing is also true for a vertical and diagonal edges in each direction.

In the Marr-Hildreth (zero crossing) edge detector a global threshold is used to filter out false edges by setting all values below the threshold to zero. If the threshold is set to low the risk of false edges (false positive) is high, but if the threshold is set to high the risk of removing true edges (false negative) is also high. Canny edge detection is trying to solve this by introducing a hysteresis thresholding, which works by using two thresholds, one higher and one lower. All edges above the higher threshold is said to be strong edges whereas the ones above the lower threshold (weak edges) is to be decided through the hysteresis.

The hysteresis works by starting from a strong edge and marking all weak edges which are connected through a e.g. 8-connectivity, as a strong edge. This process is continued to the point where there are no more any connectable weak edges and they are then put to zero.

2.2 Image Morphology

The word morphology comes from Greek and means "study of shape", where it commonly denotes the branch of biology which looks at the shape and structure of plants and animals. In image processing, morphology refers to the broad set of operations that processes images based on their shape. These operations include but are not limited to removing/adding or reducing/amplifying pixels values, closing or opening gaps in areas with none or weak connectivity and filling in closed areas. Such operations could be helpful in the interpretation and analysis of the shapes in our data. Morphological operators can be used on grayscale and binary images, 2D and 3D, but we will only look at the 2D binary case in the following text and examples.

In image processing, morphological operators can be seen as the operation between two sets of operators: Object and structuring element (SE). The object is the binary input image, where ones are defined as foreground and zeroes as

background. The SE can have any selected shape and can be specified as both foreground and background depending on the operation. The most common operations are dilation, erosion, opening and closing.

2.2.1 Dilation and Erosion

With both the erosion and the dilation, it can be helpful to look at it as a convolution between A, the original binary image and B, the SE. The formula for both erosion and dilation can be found in Equation 2.5 and Equation 2.4, respectively. The difference between the operators is the value the neighbouring pixels are switched into and the activated binary value. In erosion, the convolution condition is satisfied when the centre of the SE is at a 0 value in the original binary image A. Being on a 0 turns all elements in the shape of the SE into 0 as well, leaving the new image trimmed from the original image A. In dilation, the opposite is true, where the result is a thicker image than the original. The dilation is done by the convoluted condition being satisfied when the centre of the SE is 1, turning neighbouring pixels to 1 in the new image.

To test out and show dilation and erosion, Figure 2.1 is used, which contain three areas that are going to showcase key takeaways from the operations. These include a single pixel at the top, a broad line and a set of single columns with missing pixels in the middle two columns.

$$A \oplus B = \bigcup_{b \in B} A_b \tag{2.4}$$

$$A \ominus B = \bigcap_{b \in B} A_{-b} \tag{2.5}$$

Table 2.2: Parameters in equation 2.5 and 2.4

Parameter	Description
A	Binary image
B	structuring element (SE)
$\ominus$	Erosion
$\oplus$	Dilation
$\bigcap$	Intersection
$\bigcup$	Union

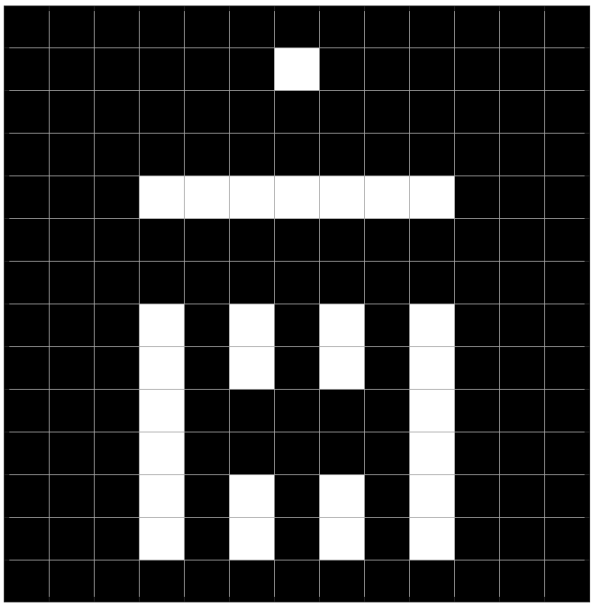
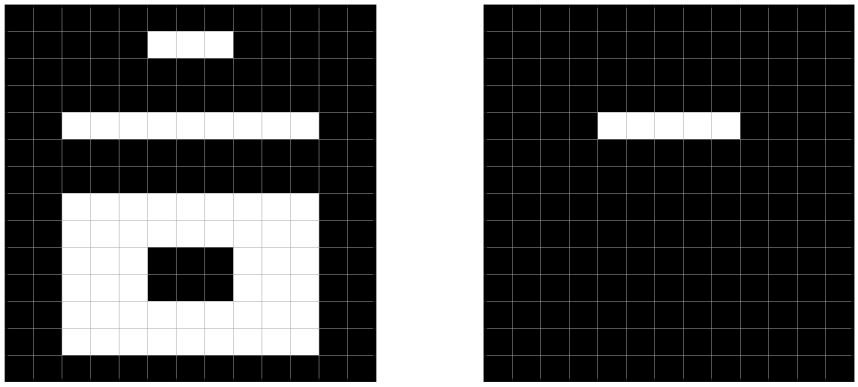


Figure 2.1: Base image for morphological testing.

In Figure 2.2a and 2.2b it is possible to see the result of using both dilation and erosion on a original image in Figure 2.1 respectfully, using a SE of size 1×3. Note from Figure 2.2a that the single pixel at the top has taken the form of the SE used and that the line has expanded with one pixel in each direction. The stripy area has connected into the shape of a doughnut because the stripes were only separated with a distance of one pixel. The opposite is true for erosion in Figure 2.2b where only the thick line survives the erosion due to it being thicker than at least 3 pixels wide.



(a) Dilation. **(b)** Erosion.

Figure 2.2: Result of (a) dilation and (b) erosion on the original Figure 2.1 with a SE of size 1×3.

2.2.2 Opening and Closing

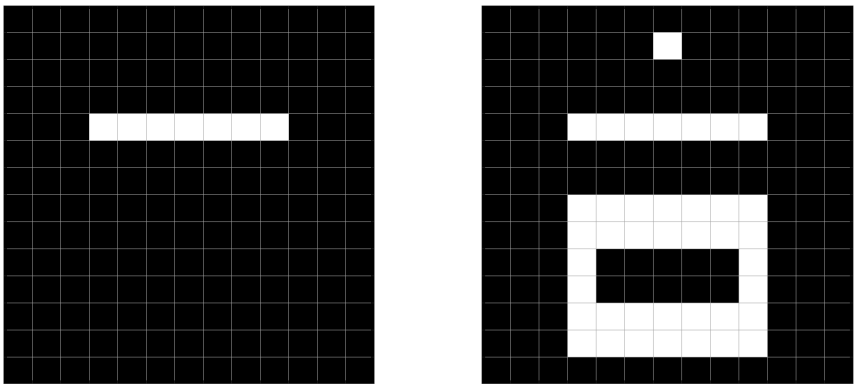
Morphological opening and closing are built on the same principles as erosion and dilation in the previous subsection, differing in that they are a combination of both, depending on which one is applied first. Looking at the formulas given in Equation 2.6 and 2.7, we can see that opening works by first eroding the original image then dilating it. In contrast, closing works by eroding a dilated image. For both opening and closing, the SE stays the same. The result of opening and closing can be seen in Figure 2.3 where all of the original pixels in Figure 2.1 which were thinner than 3 pixels in the horizontal direction disappeared. The thick line remains unchanged from the original image because it did not disappear in the erosion step. Closing, on the other hand, starts with dilation, which connects the columns in the bottom, creating a doughnut shape witch remains after the subsequent erosion, Figure 2.3b. Both the single pixel and the horizontal line is left unchanged.

$$A \circ B = (A \ominus B) \oplus B \tag{2.6}$$

$$A \bullet B = (A \oplus B) \ominus B \tag{2.7}$$

Table 2.3: Parameters in equation 2.6 and 2.7

Parameter	Description
○	Opening
•	Closing



(a) Opening. **(b)** Closing.

Figure 2.3: Result of (a) opening and (b) closing on the original Figure 2.1. with a SE of size 1×3.

2.3 Noise and Noise Removal

Noise is unwanted and semi-unpredictable data values in the dataset, which distort the underlying data and can make it challenging to interpret. For example, in the case of GPR images, depending on the severity, noise could make it harder to detect layers and distinguish them from each other. For some images, it is possible to differentiate the noise from the signal of relevant information, but this could pose a problem when using computer methods such as edge detection to identify boundaries. Edge detectors, particularly early ones, utilize a change in intensity values. This change in intensity values makes them vulnerable to false positives and false negatives depending on the noise level.

The noise can come in multiple forms where data values can be evenly distributed across the dataset, occur at regular intervals, or show a predilection towards specific intensity values. Noise predominantly results from the process of acquisition and transmission [21]. During these transitions and between, recording or transmission tools are vulnerable to, e.g. environmental distortions

like background noise, interference in the transmission channel and sensor temperature. Therefore, it is desired to reduce the significance of noise in the image and promote the signals of interest. The removal or mitigation of the noise depends on what kind of noise is present.

2.3.1 Filters

A simple form of reducing noise in the dataset is by running filters over the data depending on the characteristics of the noise present. The simplest filter is the arithmetic mean filter. A set kernel is convoluted over the given dataset where all values within the kernel are added and divided by the number of elements contained in the kernel. This filter blurs the data points with its neighbours, where it helps blur out outliers and smooth areas for further filtering. Note that the size needs to be chosen carefully since this filtering also blurs edges.

For data that has been corrupted by salt and pepper noise, a median filter could be the best approach to reduce the noise where extreme values within a kernel are downplayed. The benefit of the median filter is that it perseveres edges better than the mean filter.

2.3.2 Wiener Filter

The Wiener filter is a powerful noise removal filter that approximates the undegraded image by minimising the mean square error between the reconstruction and the original signal. The filtering technique is initially complicated and does utilise the frequency domain [22]. However, for using the Wiener filter in this thesis, an approximation will be used employing the mean and variance around each pixel in the original image. Equation 2.8-2.10 shows the step of this approximation where μ is the mean value in the area N×M, σ^2 is the variance in this area, and b is the Wiener approximation for the centre pixel. ν denotes the noise variance, given by the average of all the local estimated variances.

$$\mu = \frac{1}{NM} \sum_{n_1, n_2 \in \eta} a(n_1, n_2) \tag{2.8}$$

$$\sigma^2 = \frac{1}{NM} \sum_{n_1, n_2 \in \eta} a^2(n_1, n_2) - \mu^2 \tag{2.9}$$

$$b(n_1, n_2) = \mu + \frac{\sigma^2 - \nu^2}{\sigma^2}(a(n_1, n_2) - \mu) \tag{2.10}$$

Table 2.4: Parameters in equation 2.6 and 2.7

Parameter	Description
μ	Mean
σ	Variance
$\sum$	Summation
η	N-by-M local neighbourhood of each pixel in the image
b	Wiener approximation
ν	Noise variance

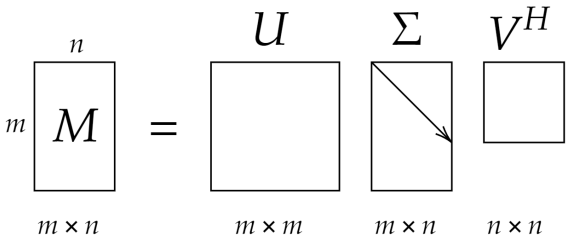
2.3.3 Singular Value Decomposition Filtering

In linear algebra, the method of singular value decomposition (SVD) is factorizing a matrix into three matrices which consists of some beneficial algebraic properties. For example, with SVD the matrix M can be factorized into three matrices: U, Σ and V^H see Equation 2.11, where H stands for the transpose-conjugate indicating complex coefficients, transpose if real coefficients. Commonly SVD can be used as the basis for the principal component analysis, which is one of the most used techniques in statistics for taking high dimensional data to understand it in terms of its most dominant patterns of correlations.

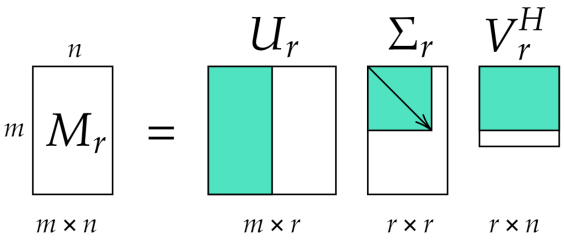
In a matrix M of size (m,n), U and V are two singular vectors forming two orthogonal matrices of size (m,m) and (n,n), respectively. Where Σ is a diagonal matrix of the same size as M, see Figure 2.4a. This Σ matrix is known as the singular value of M. The non zero elements in Σ displays singular components in descending order. The value of these singular components can be interpreted as the magnitude of a semiaxis in an ellipsoid of n-dimensions. The magnitude of each axis can be understood as the standard deviation or spread of data along that particular dimension.

Singular values are related to the more commonly known eigenvalue and are related in that the singular values in our matrix M of size (m,n) would be the square roots of the eigenvalues in the matrix $M^H M$ of size (n,n). However, they differ in finding the eigenvalues through eigenvalue decomposition where the input matrix must be square, whereas the corresponding matrix in SVD does not.

$$M = U\Sigma V^H \tag{2.11}$$



(a) SVD equation visualized.



(b) SVD reduction operation visualized.

Figure 2.4: Visualization of (a) the SVD equation and (b) how SVD reduction influences the data.

Using the equation in reverse, multiplying all the singular vectors with each singular component creates the original matrix. However, it is possible to remove some of the singular values and their corresponding U and V^H values to alter the image without altering its shape. In Figure 2.4b we can see the result of removing low-value components of the Σ and its impact on the rest of the expression. Areas marked in turquoise do contain variables that contribute to matrix M_r, where index r indicates that the matrix has been reduced. By removing specific singular values, we can disproportionally remove some of the noise, leaving the areas of interest relatively untouched [23]. Removing singular components works because some components contribute more to the noise than others.

2.3.4 Thresholding and Trimming

Thresholding and trimming could be valuable tools in removing points or areas that are not in use or contain a large amount of noise. This method is a practical solution in lowering the amount of filtering needed in the image and reducing the tedious process of tuning the parameters for each dataset. Therefore, to implement this practical approach, finding a method that can adapt to the layers and the given dataset would be helpful.

Basic global thresholding could help separate the image into two classes, objects

and background. The thresholding technique is done automatically for data where the separation between classes are distinct, but this is not always the case. Therefore, it could be helpful to use optimum global thresholding using Otsu's method. Otsu's method works by using the PDF of the intensity values, or an approximation of the PDF like the histogram, in the data to increase the intra-class intensity variance [24]. Increasing the intra-class intensity variance could be a problem in datasets with a large amount of noise. However, it can be improved by reducing the noise through noise reduction techniques, like mentioned above.

Trimming can also be implemented where values in certain areas that are not of interest can be removed entirely, reducing the amount of filtering required and leaving the areas of interest more intact.

3

Datasets

The GPR datasets used in this thesis was collected by NORCE researcher and co-supervisor of this thesis Dr. Rolf-Ole R. Jenssen, on a field excursion during the period from 14-19 of April 2021. The excursion was conducted in the context of testing the performance of a radar system made and developed by Jenssen & Jacobsen [25], mounted to an airborne drone. During the excursion, the surveys were conducted at different locations with varying degrees of snow depth, internal layers, wind conditions, drone altitude and velocity, surface topography, altitude and more—all these parameters contributed to making different looking datasets. Furthermore, the data was handed over from Jenssen preprocessed, where the option to change the preprocessing was available but not addressed in this paper.

Figure 3.1 shows a representative subset of a GPR dataset, displayed with some noise reduction already applied. For reference, the axis of this image is given in terms of its real-life distance. The x-axis is in meters and is calculated from the GPS data, and the y-axis represent the distance of radar waves in air, given in cm. The bottom and internal layers in the figure will appear deeper in the dataset (in air cm) than in real in situ measurements since the radar wave travels slower in snow than air, and how much deeper depends on the density.. The forthcoming GPR figures in the thesis will be depicted in terms of the sample points instead of distance. As in Figure 3.1, all figures will also be shown with their colour map inverted to better fit with the surrounding page.

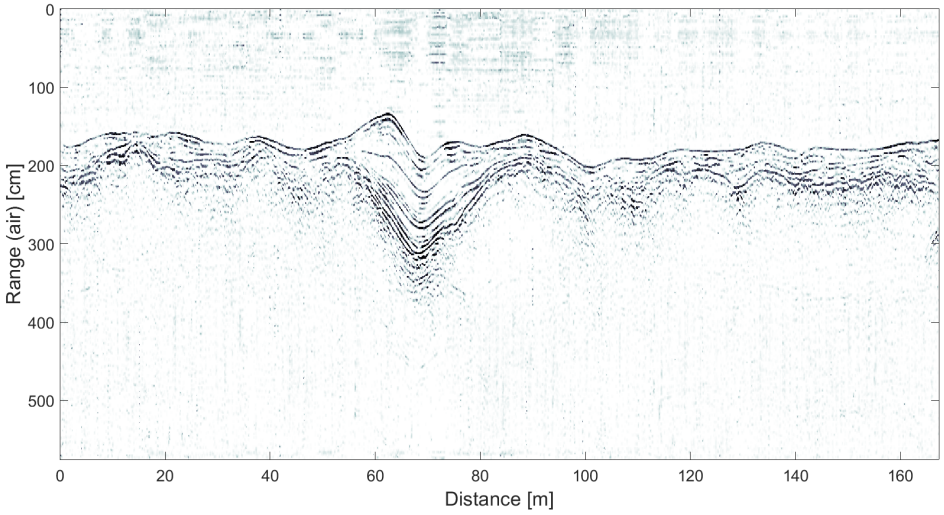


Figure 3.1: Subset a GPR image with filtering applied. Real-life distances are displayed.

3.1 Data Gathering

The drone used for conducting the surveys and carrying the antenna system was a NOrce-developed octocopter called Cryocopter FOX. The choice of the drone was made due to payload capacity and weather resilience compared to other airborne drones [1]. The data gathered from this drone got collected with two Vivaldi antenna arrays containing two antennas each, in a bistatic configuration. The advantage of using a Vivaldi antenna is that they have relatively high bandwidth and medium directivity compared to other antennas of similar size. With these antennas, the total system bandwidth results in a range resolution of approximately 5cm. The range resolution comes from the total system bandwidth of the radar. I.e. the bandwidth of the antennas and the bandwidth of the waveform transmitted to the antennas. Including in the antenna choice and setup, some modifications have been made to the traditional Vivaldi design to improve its characteristics, [1].

In addition to the antenna, a laser altimeter was mounted on the platform to locate the first layer. However, due to significant amounts of noise, possibly due to random specular reflections on snow crystals, this data is not adequately consistent to determine the first layer. In Figure 3.2 we can see another section of the filtered GPR data where the laser altimeter measurements are displayed. Notice how it manages to follow close to the top layer but is not accurate, and sometimes it dives down into the layers. Note that when using the laser

altimeter, the mounting position offset on the platform was noted and corrected for, since it affects the final detected range of the ground.

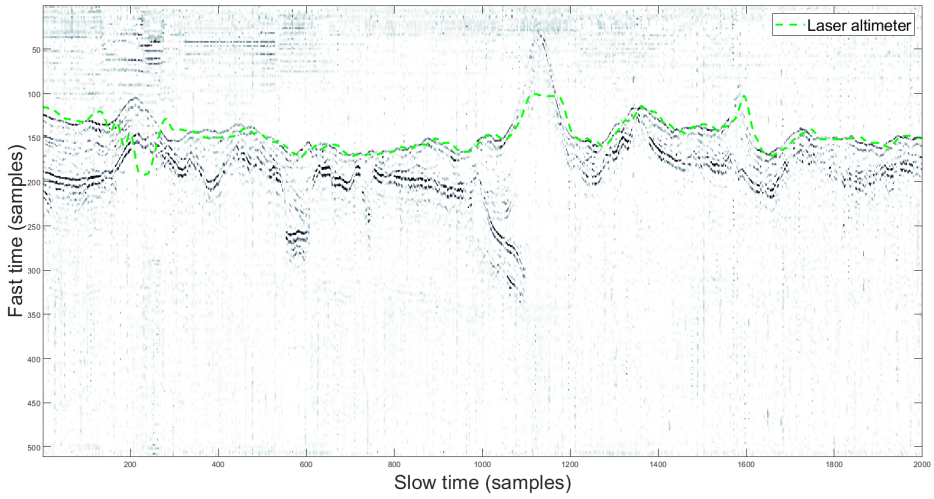
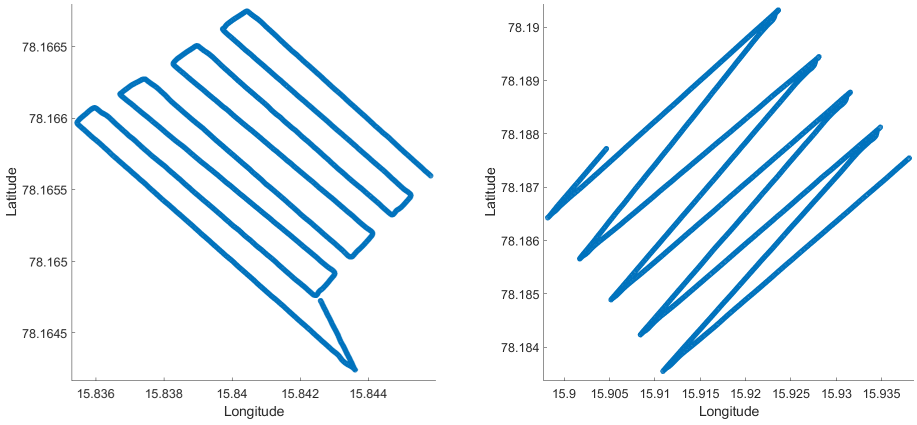


Figure 3.2: GPR data with the laser altimeter measurements displayed.

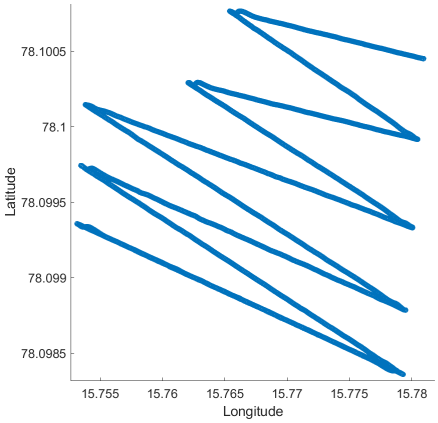
During the data collection, the corresponding UNIX timestamp is recorded, in addition to the global positioning system (GPS) location of each sample, see Figure 3.3. This data is used to georeference the data. Each gathered measurement is based on the transmission interval of the signal and is susceptible to a variation in the drone's speed. Therefore, with data of the location and time, an interpolation of the image can be made to adjust for the variation in speed.

To summarise, in the data that was provided, we have a 2D intensity image, plus the corresponding altitude collected from a laser altimeter and the GPS positions. All this data will be utilised to find a method to find the main layers and comprehensively present the results.



(a) Dataset 1.

(b) Dataset 2.



(c) Dataset 3.

Figure 3.3: Flight pattern and GPS location for the different datasets.

3.2 Description of Data

The data collected depends on various components, some already alluded to in the previous section. These are things that a person with some or even no insight into GPR would be able to point out. In addition, some easily overlooked problems become evident when using GPR from a drone, but also some familiar artefacts are seen in data handling of electromagnetic waves.

If we look close enough in all radar data, we can observe convex hyperbolas in the image. The hyperbolas are imaging artefacts arising from radar detecting the subsurface on a moving platform. The main principle of the hyperbolas is

a result of the directivity of the antenna, where the GPR can detect an object, e.g. a rock, before and after it is straight above the object. If we use an antenna with a hypothetical directivity equal to a laser beam, we would only get a response when we are strait above the object. On the other hand, if we use an antenna with a lower directivity, we would pick up on the object before the antenna is above the object. Running the antenna with low directivity will record a hyperbola shape. The tail end of the hyperbolas is when the distance is further apart. The hyperbolas are primarily expressed in the bottom layer in our datasets, making them appear thick, but also in other places though less pronounced.

In short, these components include: aliasing of the returned wave outside the unambiguous range [26], which are out of bound, hyperbolas which makes up the bottom layer, noise associated with the drone RF telemetry, ringing along boundary lines and other kinds of noise.

3.2.1 Aliasing

In the circumstance of when a target moves out of the unambiguous range of the radar, a sampling artefact called aliasing will occur. The artefact expresses itself in that data points, which we would expect below the sampling range to wrap itself around and display at the image's top, see Figure 3.4. This aliasing gives the impression that, e.g. layers goes through the bottom and appear at the top. For most instances, it is easy to see the connection between the layer going from the bottom to the top, but it adds an extra challenge to automatically finding the layers. In the datasets looked on in this thesis, aliasing is not a big problem and mainly shows in the initial and end phase where the drone approaches its target altitude or landing. However, these areas are prone to other noise and errors due to the adjustment of the drone altitude. Therefore, these areas are not of interest in the survey and will not be looked at when evaluating the feasibility of the detection method. The origin of this artefact is two folded and comes from the periodicity of the transmitted signal and the cyclic nature of the match filter processing [26].

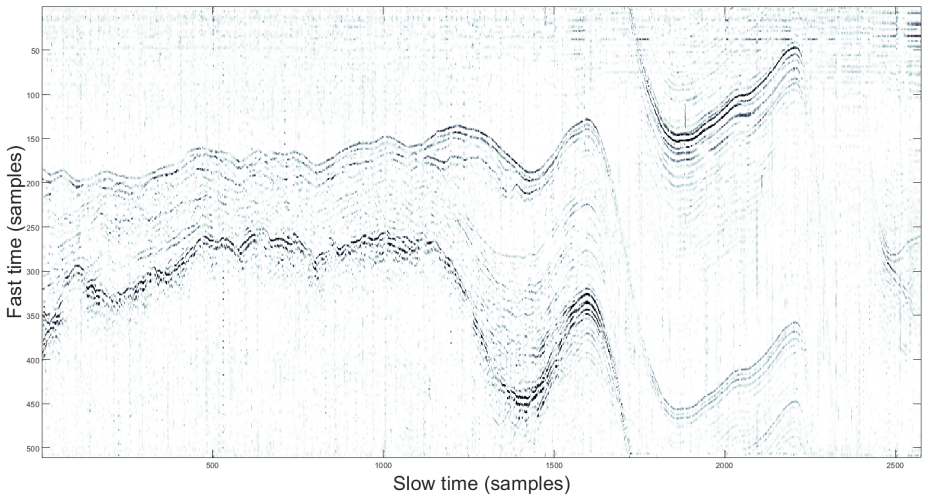


Figure 3.4: Aliasing, example of how it moves out of the unambiguous range of the radar. Here we can see how the bottom layer goes below the range and wraps itself to the top of the data.

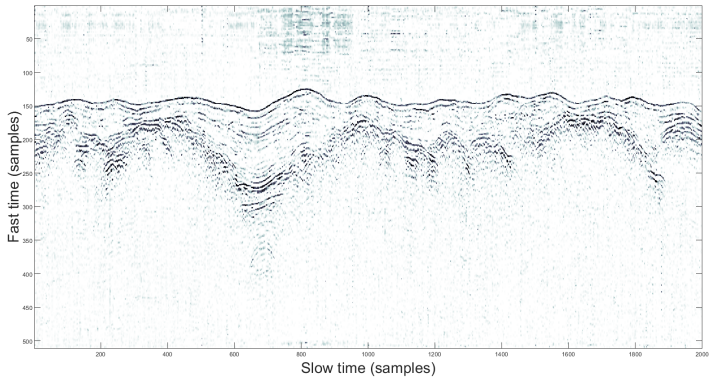
3.2.2 Difference in Datasets

In this thesis, we will look at three different datasets taken at three different days, locations, variations in conditions, and flight patterns. These differences will contribute to testing the robustness of the edge detection technique under different scenarios. A subset from each dataset can be seen in Figure 3.5 where a representative subset has been chosen to illustrate the key features of the subset. Each dataset will be referred to as dataset 1, 2 and 3.

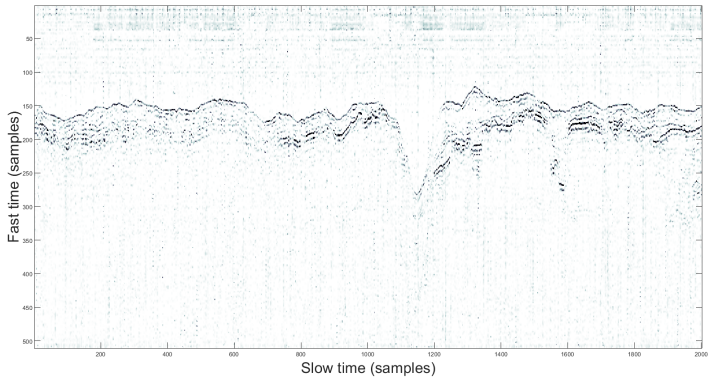
The drone had an altitude of about 8 meters above the snow during the survey in the provided datasets. For dataset 1, we have a grid-like survey which we can see in Figure 3.3a where a total of 1.8km distance was covered. The survey area was characterised by having a shallow snow level and flat topography with some local increases in-depth, see Figure 3.5a. These local sudden drops and increases in depth can come from wind-packed snow variations in terrain or around other local objects, like a rock. The depth variability is more prevalent in the other datasets than dataset 1. Dataset 2 depicts a new area where the drone has a higher speed and has changed to a sawtooth survey pattern, see Figure 3.3b. Because of the higher speed and less turning in the flight pattern, the drone traverses a distance of 7km. A shallow snow level characterises the data collected from dataset 2 as in dataset 1. However, it does have more variation in topography where we can have a sudden drop or increase in altitude. One of the sudden altitude changes can be seen in Figure 3.5b where there is a

drop in elevation of the terrain.

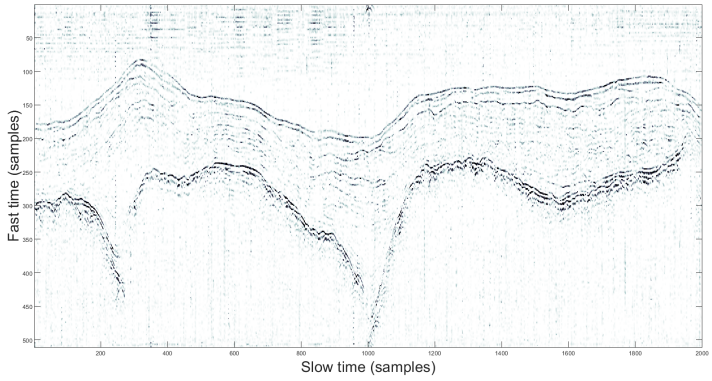
In the last provided dataset, we have a sawtooth survey pattern as in the second dataset, see Figure 3.3c and a distance travelled of 5.5km. However, the dataset is different compared to the two other ones in that it depicts a survey area with a higher snow depth. It also has a more varied topography change, see Figure 3.5c, where some places we do have occurrences of aliasing. It should be noted that changes to the elevation in the data could also be because of a readjustment of the drone height. The ambiguity of the source of an elevation change in the data is important to recognise as we go forwards in the thesis.



(a) Dataset 1.



(b) Dataset 2.



(c) Dataset 3.

Figure 3.5: Subsets of the three different datasets used for testing.

Part I

Finding Top Layer

In contrast to the traditional structure of a thesis with a clear distinction between method and results, in this thesis, we have chosen to blur the distinction. The decision to blur the chapters comes from the nature of the thesis and how the method was developed. The method in the following chapter will, step by step, show the partial result as the method develops towards finding a technique that can detect the top layer. Showing the partial result as the method develops will give the text a better flow and invite the reader into the thought process of the choices. The chapter with the final results will be reduced in size, but it will be showing the result from a couple of subsections from each dataset. These will be shown with ground truth and the estimate from the method, the given accuracy in root mean square error (RMSE) in air equivalent cm and data samples, and cross-correlation. A quality value will be depicted at the bottom of each image to show the estimate's reliability. In helping with the explanation in the discussion, an image will depict each dataset's flight pattern in the result, with each subset marked.

The method development for the top and bottom layer has been chosen to be conducted on a subset from dataset 1. Along with the development, we will adjust and introduce solutions to problems that arises from other parts of the dataset and datasets 2 and 3. This decision in using dataset 1 was made based on the observation where the dataset had a brighter top layer and that it was a middle ground between the snow depth between dataset 2 and 3.

All programming in this thesis was done in MATLAB, where most functions used were from MATLAB toolboxes. If the author or others make some custom functions, a note will be given.

4

Method Development

4.1 Preparation of Data

The preprocessing stage mentioned in the previous section outputs a relatively "raw" dataset. This dataset is challenging to read and interpret but can be processed by filtering out the noise to enhance the signal of interest. Figure 4.1 shows a subset of the radar image with no filtering. Notice the distinct horizontal lines at the top of the image and the layers, which can be seen in samples 150-200 in fast time. It should be noted that the colour map has been inverted and will be inverted on the following images in the thesis to better fit with the page. The choice of inverting the colour map will be better visible after some filtering steps.

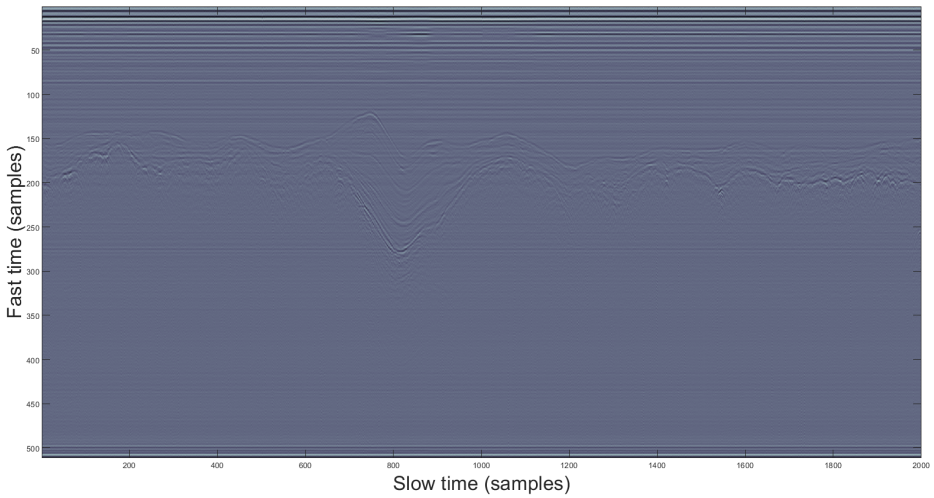


Figure 4.1: Raw radar image with direct wave at top.

This horizontal noise originates from the direct wave between the transmitting and receiving antenna and poses a challenge for interpreting the image. Therefore, it would be beneficial to remove it without disturbing the signal of interest.

Since the direct waves are relatively stationary horizontally, a row-wise mean subtraction would effectively remove a large part of its contribution while leaving other dynamic areas with little change. In Figure 4.2 we can see the plot of the row-wise mean value of Figure 4.1, with the signal of interest marked with a red circle. The figure shows a strong oscillating signal at the start, where the horizontal noise is the most prevalent. This oscillation does not stop throughout the plot, but it dampens when moved away from the initial rows. Therefore, removing the mean value from each row will heavily impact the noisy areas and minimally impact the rest of the image. Filtering in the frequency domain was also attempted but did not remove the horizontal lines.

In Figure 4.3 the image can be seen with the mean value removed from each row. With the mean value removed and the horizontal noise reduced, it has brought forward both the layers of interest and other underlying noise in the image. With the layers of interest more visible than beforehand, we would like to make them more visible by further filtering.

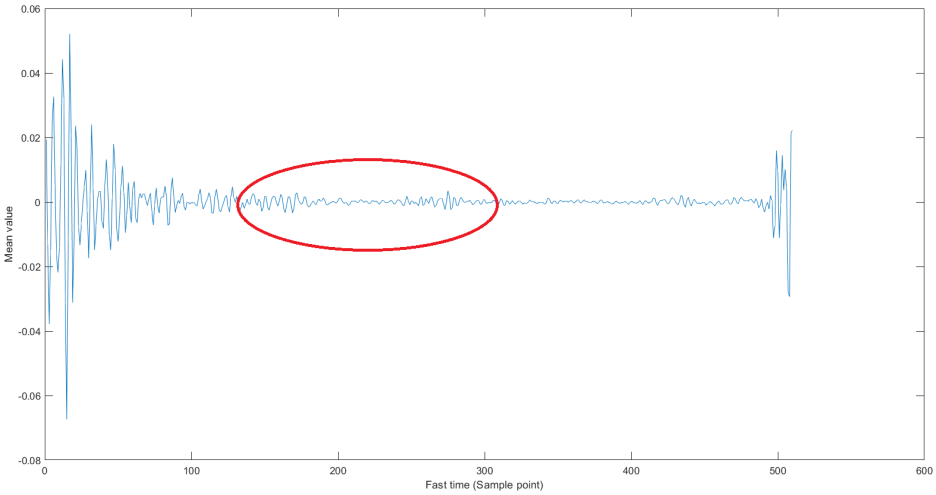


Figure 4.2: Mean value for each row in Figure 4.1 with the signal of interest marked with a red circle.

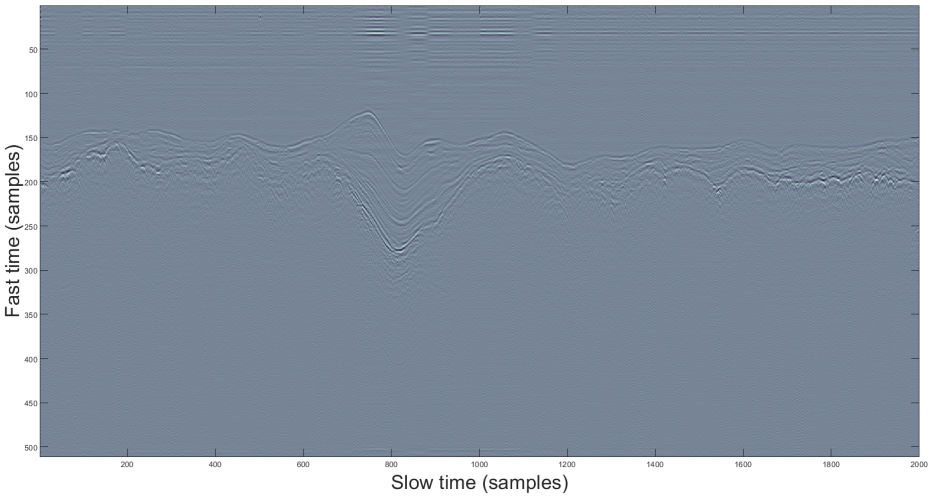
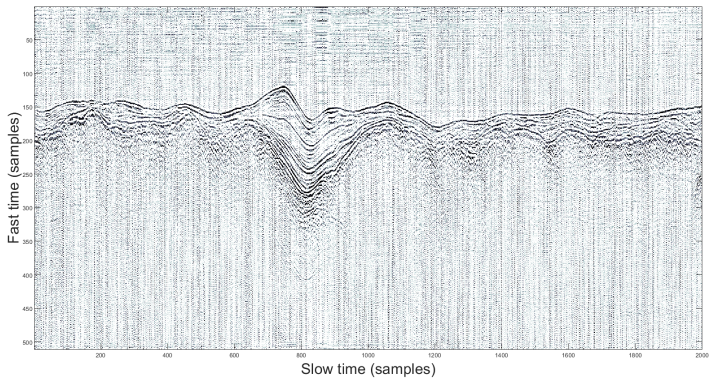


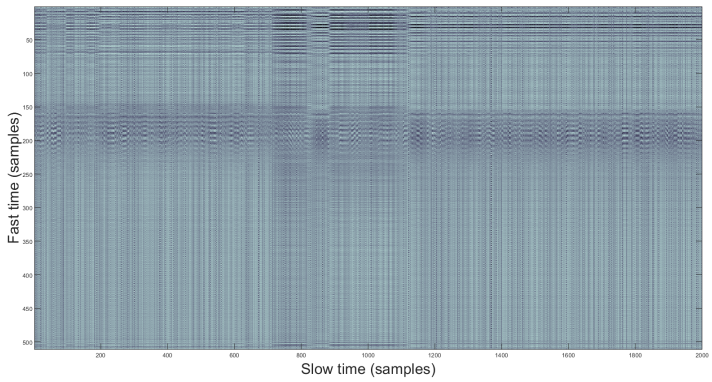
Figure 4.3: Mean value for each row removed from Figure 4.1.

To enhance the layers and remove noise, we will use singular value decomposition (SVD) to remove singular components which contribute disproportionately to part of the noise. We do this by first contrast stretching the top and bottom 5% of the intensity values for better detection. Subsequently, we use SVD to remove the ten first singular components in the image. The result of the reduc-

tion of the image with a further contrast stretching the top and bottom 1% can be seen in Figure 4.4a, where the removed values can be seen in Figure 4.4b. This image has a substantial amount of less horizontal noise across the image. However, the contrast stretching has brought forward other noise, including vertical lines coming from a narrowband (telemetry) RF link. We further reduce this noise by running both a 3×3 median and Wiener filter over the image in that order. Figure 4.5 shows the result of the filtering. The median filter is chosen to reduce the amount of small-scale salt and pepper noise, whereas the Wiener filter is chosen because it filters noise but preserves the main structure (i.e. snow layers) in the image.



(a) Filtering by removing singular components.



(b) Removed components from SVD filtering.

Figure 4.4: (a) Result of filtering by removing singular components on Figure 4.3. (b) Removed values from the SVD filtering. .

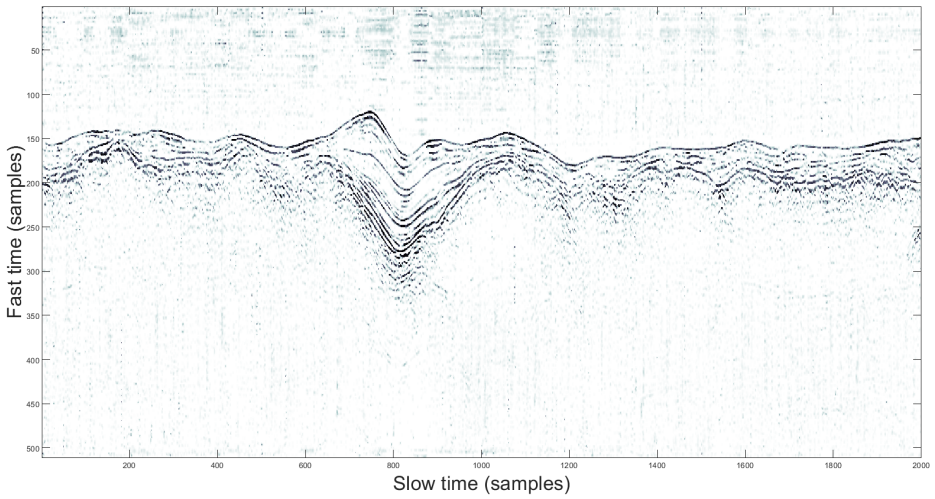


Figure 4.5: Result of additional filtering on Figure 2.11 by a median and Wiener filter of size 3×3.

This image has been sufficiently prepared where the layers have been enhanced and the noise reduced. The next step is to have a method to detect the top layer with a minimum amount of disturbance from the remaining noise.

4.2 Layer Detection

As seen in previous pictures of the provided datasets, the two layers of interest, the top and bottom layers are different in their characteristics. Typically, the bottom layer appears thicker due to the hyperbolas' tail extent and complexity of the snow-ground interface. Therefore, two different approaches are needed to find a method with acceptable accuracy. In Figure 4.6 we can see the subset with the ground truth for the top layer displayed. The ground truth was drawn manually with guidance from an experienced GPR researcher, Dr. Rolf-Ole R. Jenssen. Notice how the top layer is marked with an explicit edge boundary from the transition between air and snow. However, the top layer is not as explicit in some areas outside the subset and other datasets. In addition to varying degrees of noise, the weak layers could pose challenges in finding a consistent layer detector.

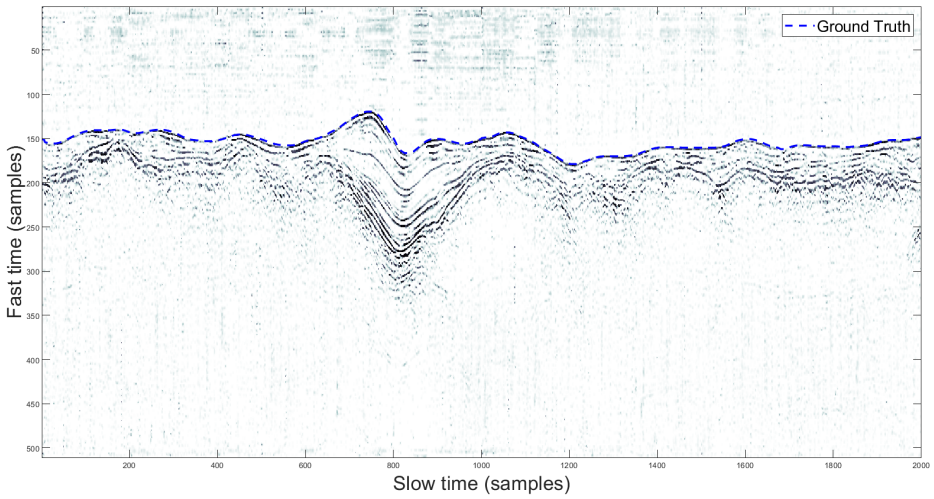


Figure 4.6: Prepared image with ground truth (blue) of top layer.

The first step in finding the top layer is trimming the image using the laser altimeter data to cut away noise which could be a problem later. We do this by taking the index of the laser altimeter, moving it up by x-amount of samples to give it a buffer zone. In this subset and in the rest of dataset 1, which we are using, it has been found through testing that a shift of 20 pixels is adequate in removing noise and avoiding cutting into the top layer. For the other datasets, the shift is different because of variations in the laser altimeter performance. All values above this shift in the original image are set to zero. Doing these steps reduces the chance of the method picking up on noise, reducing the amount of filtering needed and preserving the wanted data. Trimming using the altimeter partly overweight with the mean removal from earlier where the trimming would remove the strong horizontal lines at the top of the images. However, the altimeter cannot reduce the amount of horizontal clutter from the area with relevant info, i.e. snow layers, which leaves us with a reason to keep both steps.

These intensity values of the trimmed image are inserted into a histogram where Otsu's method finds a threshold that will be used in Canny edge detection. We use Otsu's method mainly because it is adaptive to the given histogram and maximises the variance between the two separated regions. Furthermore, using Otsu's method removes the need to set a threshold and adjust it for each dataset, which could be inaccurate and time-consuming. When using the threshold from Otsu's method in Canny edge detection, testing found that lowering the threshold to 70% increases the probability, including weaker layers. However, the lowering comes with the expenditure of some added noise.

In Figure 4.7 we can see the result of using the Canny edge detection algorithm with a threshold of 70% of Otsu's method and a standard deviation of 1. Canny is chosen because of its robustness and the possibility of setting a threshold. We have also added the trimming boundary where we, from the previous step, removed all values above the altimeter plus a shift.

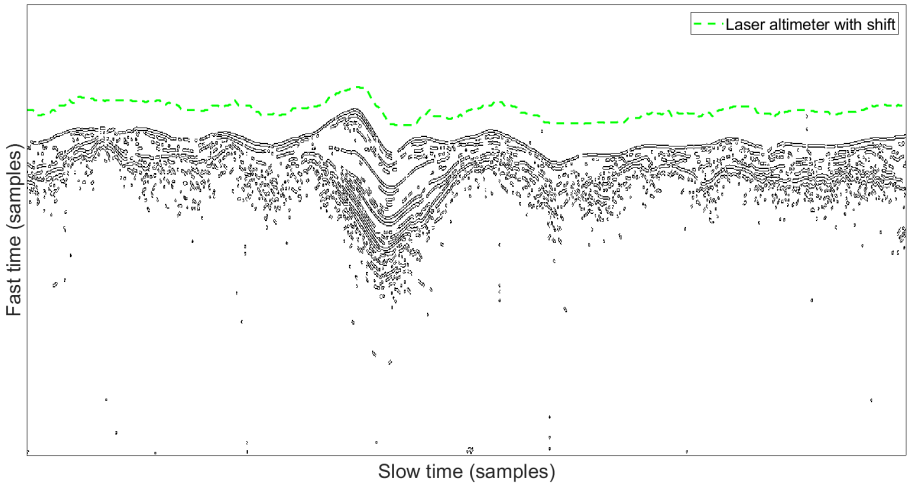
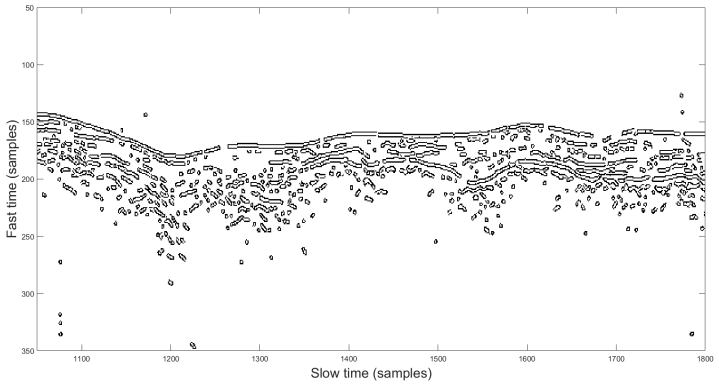
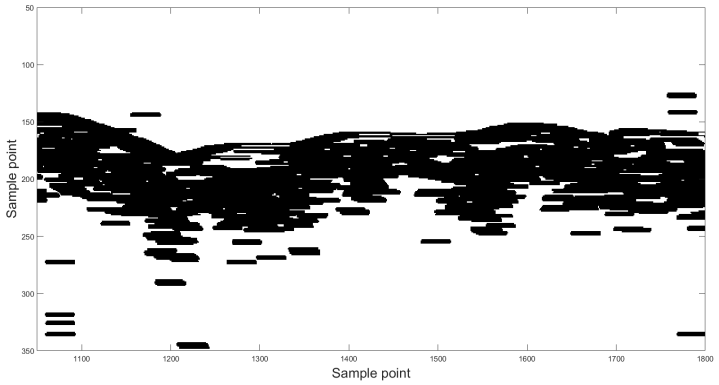


Figure 4.7: Canny edge applied to Figure 4.5 with a threshold of 70% of Otsu's method and a standard deviation of 1. The laser altimeter with a shift (green) is added to illustrate the boundary of trimming.

By the looks of Figure 4.7, it would be tempting to remove small areas in this binary image to remove the last bit of noise which are above the top layer. However, this is not necessarily a good idea since some tiny and weakly connected layers are also in the top layer. Therefore, we would connect these areas using morphological operations, starting with dilation.



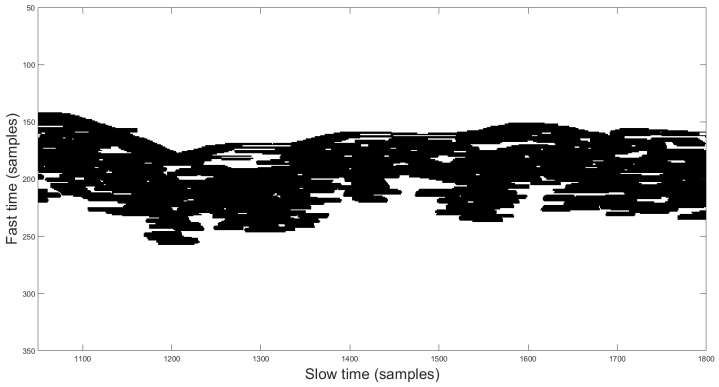
(a) Canny.



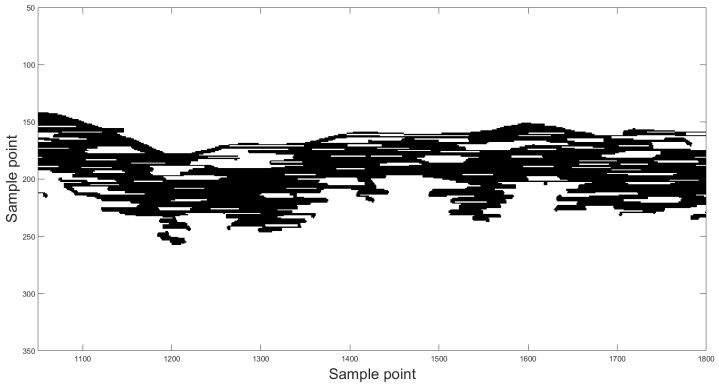
(b) Dilation.

Figure 4.8: Comparing a subset of the result of (a) Canny with result of (b) dilation with a SE 1×31.

In Figure 4.8 we can see the same area from Figure 4.7 with and without dilation applied. The dilation applied in Figure 4.8b is using a SE of size 1×31. The SE is given a horizontal bias because of the horizontal orientation of the snow layer. Notice how the weakly connected layers at point 1250 and 1690 in Figure 4.8a is now connected after the dilation in Figure 4.8b. This is a good thing as we can now remove objects below a certain size without running the risk of removing points on the top layer. In Figure 4.9a we can see that we were able to remove most of the noise above the top layer with a noticeable exception of a point at 1150, which connected with the slope. Erosion with the same SE is applied to the image to reduce the excess pixels from the dilation, see Figure 4.9b.



(a) Area removal



(b) Erosion

Figure 4.9: Reducing noise by (a) removing areas below a certain size and (b) eroding the image to balance the image from dilation.

The exact process of dilation and erosion is again done on the image where the SE is replaced with a disk of the same size. The choice of using a disk-shaped SE is made to connect areas that are still separated vertically and to smooth some of the boundaries, see Figure 4.10. The last operation that alters the image is removing objects less than 5'000 pixels in size with 8-pixel connectivity. The removal is done to reduce noise clusters that could have formed above the top layer. The last process is essential for datasets where the laser altimeter is unreliable.

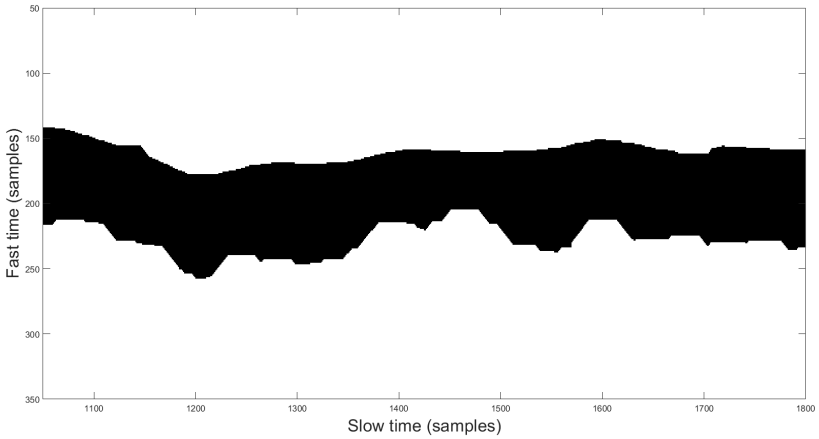


Figure 4.10: Result of using morphological closing on Figure 4.9b.

With the results from the morphological operations and area removers in Figure 4.10, we find our estimate of the top layer by marking the first transition from white to black as a top point. The resulting vector is now our top estimate, and it is smoothed with a local regression algorithm regressive locally estimated scatterplot smoothing (LOESS) with a sample of 50 points. This algorithm combines the moving average and polynomial regression, assigning lower values to anomalies. Regressive LOESS is also used to interpolate if there are missing values in the vector. A figure with the plot of the ground truth and the estimate can be found in Figure 4.11.

An attempt was made to correct the top estimate by updating its altitude by snapping it to the closest substantial value in the y-direction. The correction was meant to address instances where the estimate was incorrect because of dilation errors or the smoothing. It performed well on estimates which already did well by making them better but did worse in estimates where it was struggling. It was decided to abandon this idea because of a lack of significant increase in accuracy values like RMSE and cross-correlation.

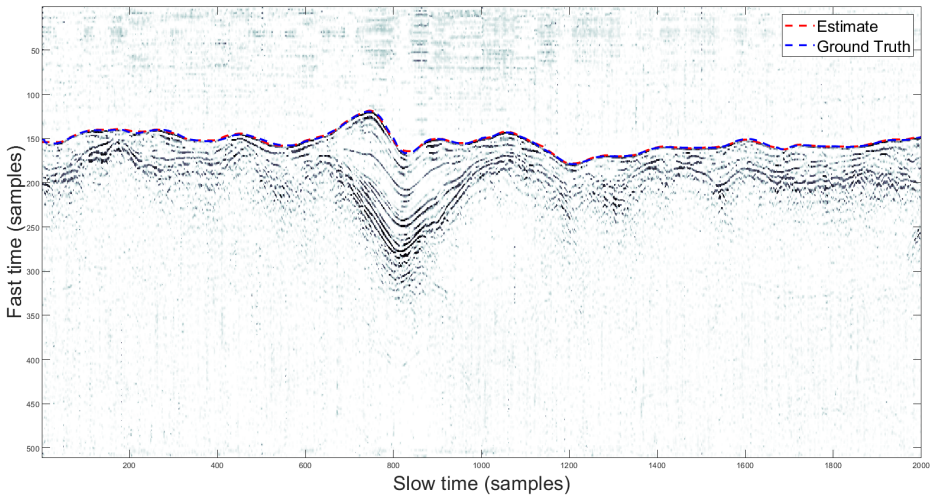


Figure 4.11: Estimate of our method (red) plotted together with top ground truth (blue).

The estimated layer has a varying degree of accuracy, where some places have a high degree of precision and others do not. Therefore, a quality check will be implemented to point out areas in the estimate with a high and low degree of certainty. In this thesis, the quality check is implemented by a self-made code that normalises the average value in a 5×5 area around the index of each estimated point. The estimate is taken from the finished image from the Preparation of Data section. The thought process of implementing the quality check is that it will give a high average value if it is close to the top layer and a low value if not, or if the signal has dropped out in this region. In Figure 4.12 it is possible to see the quality check at the bottom of the figure with its y-axis at the right side of the image. It has been judged that the cut between good and bad estimated values is approximately 0.3 through inspection. Depending on the dataset, a normalised mean value of 0.3 usually translates to the turquoise colour. Note that the area around sample point 830 in slow time for the quality check; here, we can see a drop in quality. This drop is because the estimate is hovering above the ground truth and the top layer, resulting in a low mean value. In sample points 350-400, we also see an area with a low-quality score. This time, it results locally from the top layer in the area having a lower intensity value. Since we use a normalised value, the quality measure can sometimes be slightly off in datasets with an especially good or bad estimate. Since the values in the quality check have been normalised, the values are unitless.

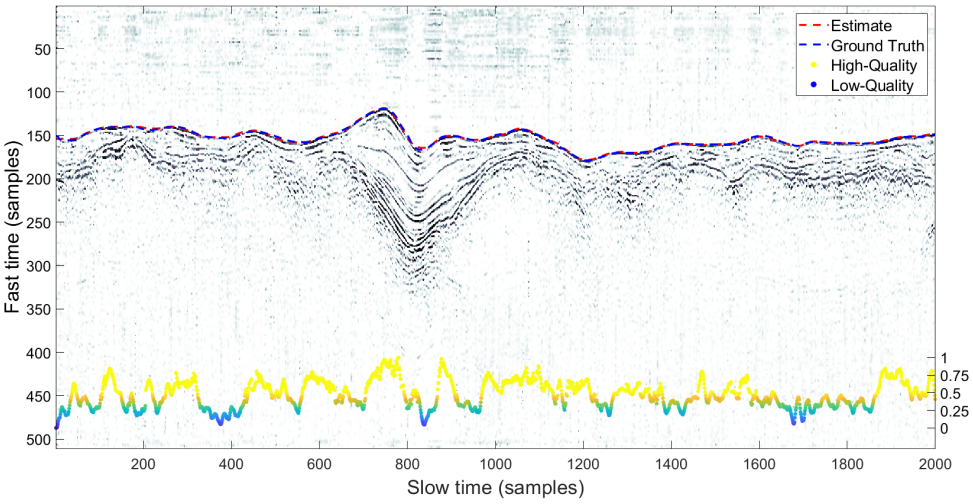


Figure 4.12: Quality check of the top layer estimate from Figure 4.11. Yellow areas translate to good quality, and blue is low.

On the other hand, the quality measure has the inherent problem arising from the simplicity of its design. It can not detect the difference between the layer of interest, internal layers and noise. To counteract the noise and internal layers, we use the same smoothing filter on the quality measure used on the top layer's estimate. After the smoothing, the quality measure works well for the vast number of cases, even when the estimate goes above and below the top layer. In Figure 4.13 we can see the result of smoothing the quality estimate in Figure 4.12

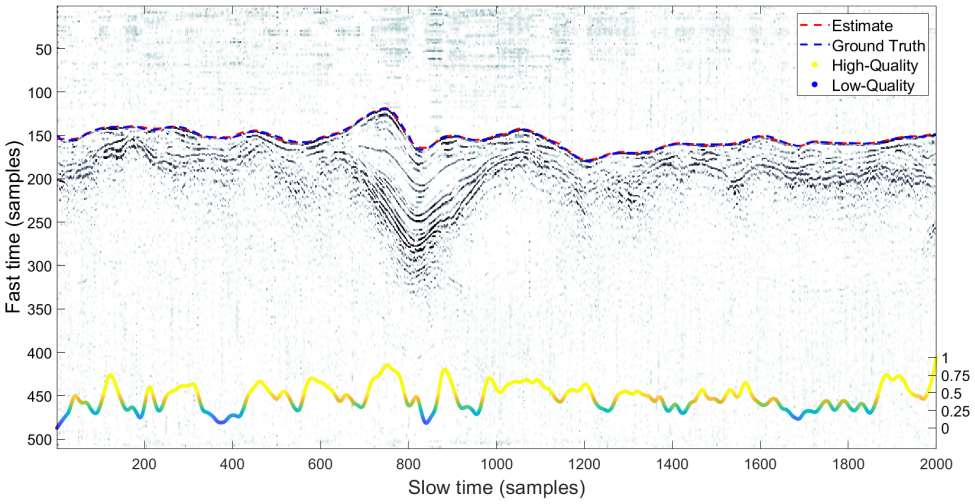
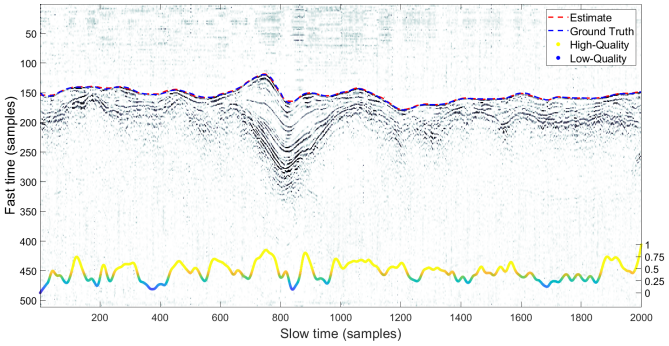


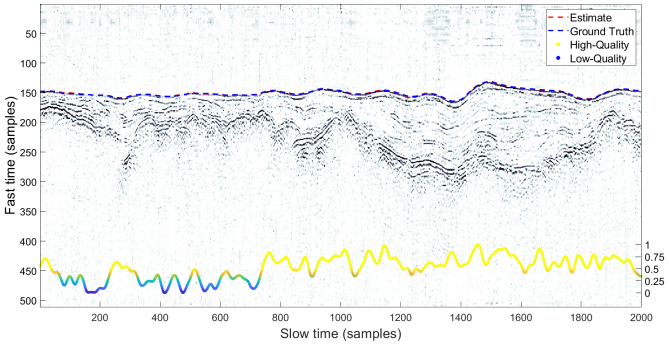
Figure 4.13: Effect of smoothing the quality measure. Dataset 2, subset 2.

5

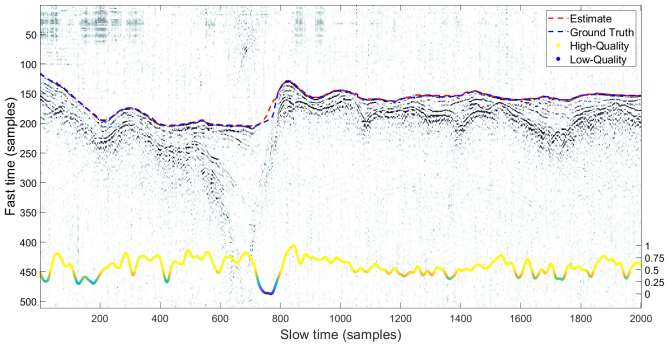
Applied Results



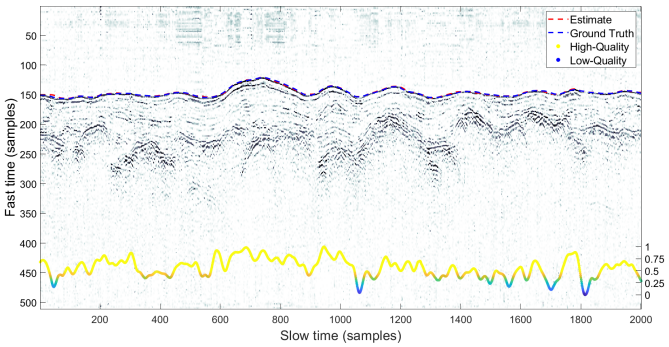
(a) Subset 1, 16'000-18'000.



(b) Subset 2, 13'000-15'000.



(c) Subset 3, 950-2950.



(d) Subset 4, 8800-10'800.

Figure 5.1: Top results from dataset 1 with their given ground truth and estimate from the method.

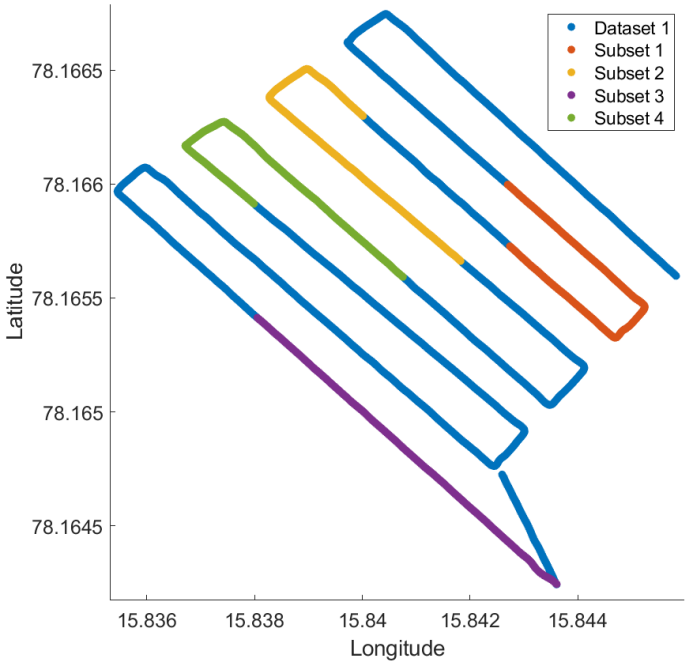
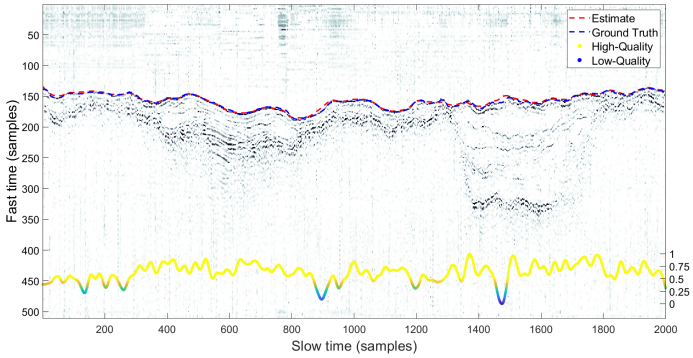


Figure 5.2: Flight pattern for Dataset 1 with the different subsets marked.

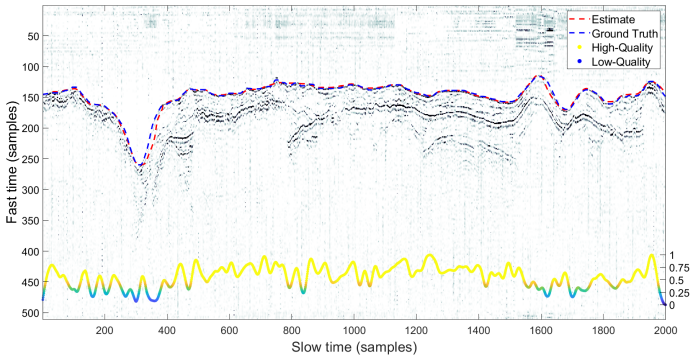
As stated in the method development, the applied result of dataset 1 was gattered by shifting the laser altimeter up by 20 samples and reducing the threshold from Otsu's method to 70%.

Table 5.1: Top RMSE in air equivalent cm and samples, and Cross correlation for dataset 1.

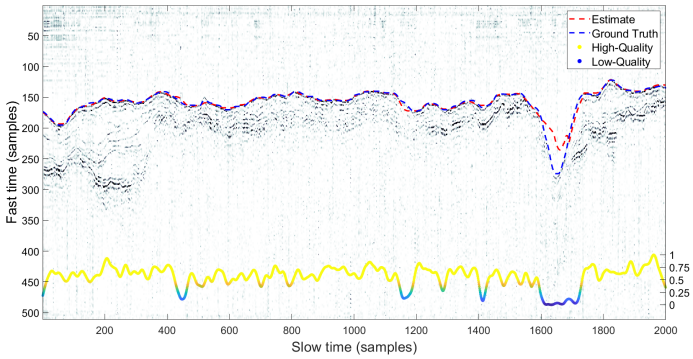
Subset	RMSE (cm)	RMSE (samples)	Cross correlation
Subset 1	1.0951	0.8531	0.9967
Subset 2	0.8772	0.6751	0.9930
Subset 3	2.8336	2.4701	0.9940
Subset 4	1.0581	0.8332	0.9940



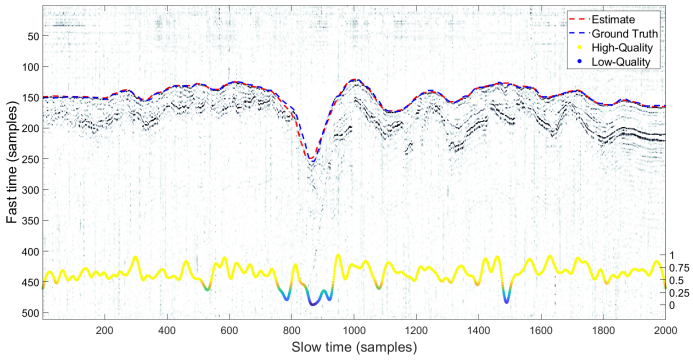
(a) Subset 1, 6000-8000.



(b) Subset 2, 8100-10100.



(c) Subset 3, 12'500-14'500.



(d) Subset 4, 25'000-27'000.

Figure 5.3: Top results from dataset 2 with their given ground truth and estimate from the method.

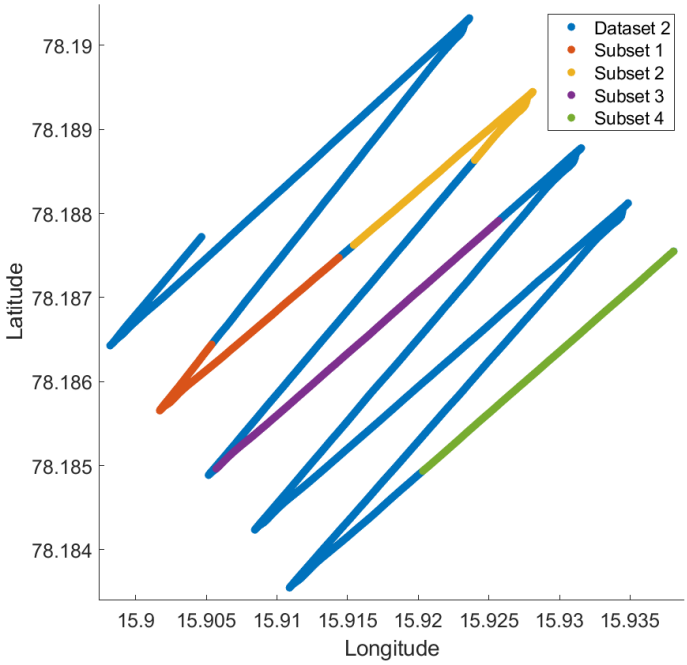
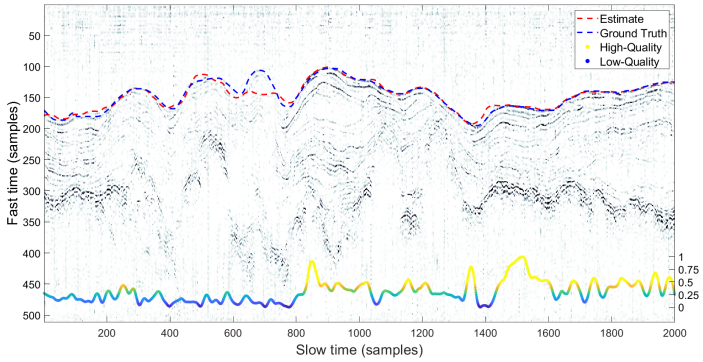


Figure 5.4: Flight pattern for Dataset 2 with the different subsets marked.

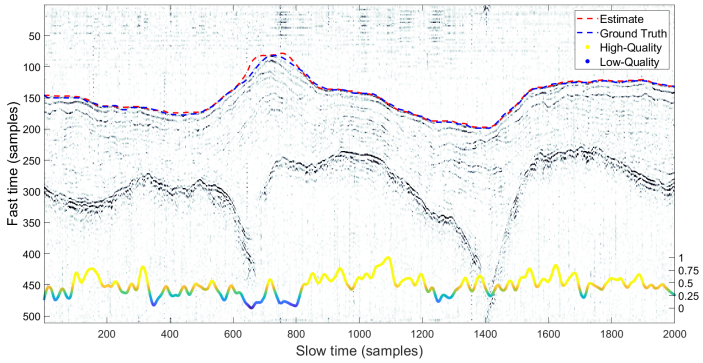
The applied result of dataset 2 was gattered by shifting the laser altimeter up by 90 samples and reducing the threshold from Otsu's method to 70%. This extent shifted by the altimeter amounted to 90 samples due to the unreliability of its measurements in this dataset.

Table 5.2: Top RMSE in air equivalent cm and samples, and Cross correlation for dataset 2.

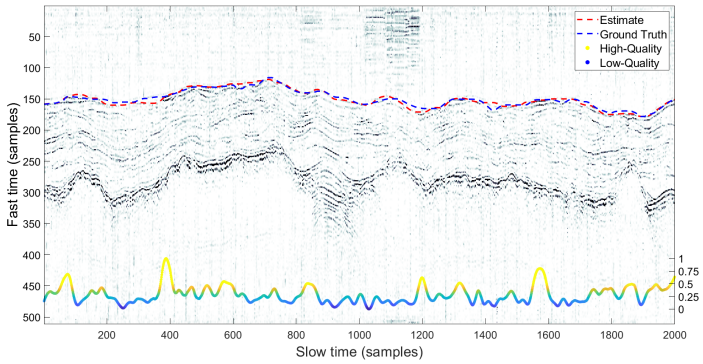
Subset	RMSE (cm)	RMSE (samples)	Cross correlation
Subset 1	2.4096	2.1029	0.9876
Subset 2	4.7532	4.2003	0.9883
Subset 3	9.1458	8.0997	0.9539
Subset 4	3.8271	3.3482	0.9891



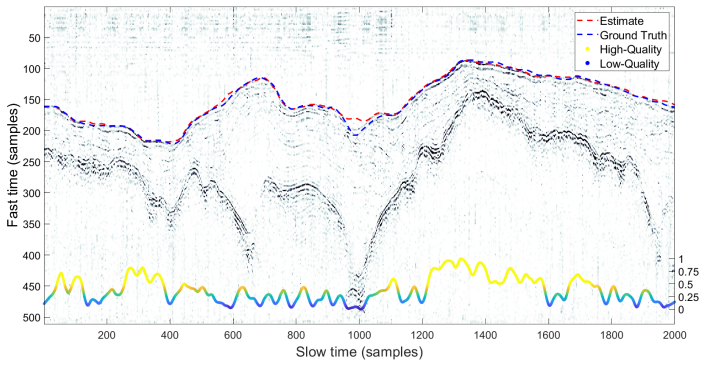
(a) Subset 1, 6000-8000.



(b) Subset 2, 8100-10100.



(c) Subset 3, 12'500-14'500.



(d) Subset 4, 25'000-27'000.

Figure 5.5: Top results from dataset 3 with their given ground truth and estimate from the method.

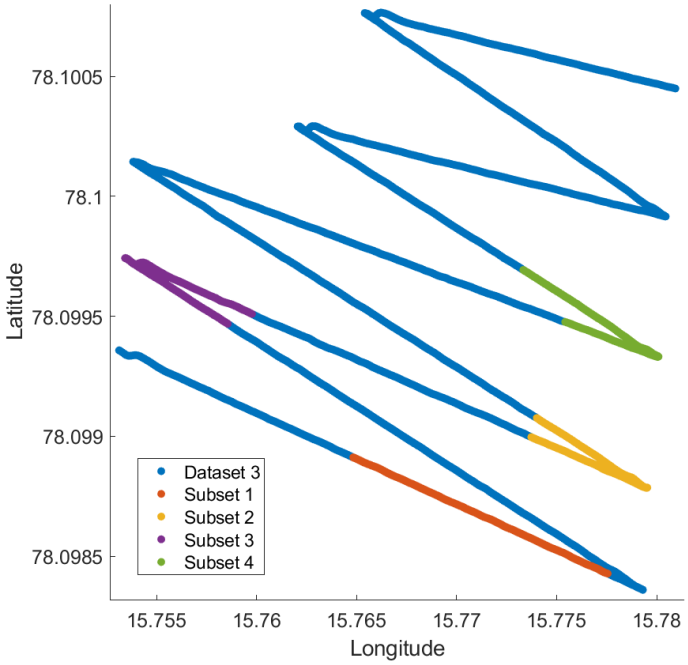


Figure 5.6: Flight pattern for Dataset 3 with the different subsets marked.

The applied result of dataset 3 was gattered by shifting the laser altimeter up by 50 samples and reducing the threshold from Otsu's method to 40% of its original value. The Otsu's method was further reduced from 70 to 40% due to areas in the dataset with weak values of the top layer.

Table 5.3: Top RMSE in air equivalent cm and samples, and Cross correlation for dataset 3.

Subset	RMSE (cm)	RMSE (samples)	Cross correlation
Subset 1	8.5729	7.5902	0.9459
Subset 2	3.7860	3.3321	0.9949
Subset 3	4.1420	3.6350	0.9700
Subset 4	5.0928	4.4970	0.9941

6 Discussion

Since we in this thesis has presented a novel solution to the problem of automatically detecting the primary interfaces in GPR data collected from an airborne drone, we do not have any other research to compare results. Furthermore, because these results stand on their own, it can be challenging to know the performance of the method other than looking at the accuracy in RMSE and cross-correlation. Therefore, without comparing it to other papers, we will in this chapter discuss the data provided, results and have a look at the strengths and limitations of the method.

6.1 Data Provided

The data was, as mentioned before, provided pre-processed with a 2D intensity image, plus the corresponding altitude collected from a laser altimeter and the GPS positions. These data varied in quality where e.g. the prominence of the top layer and the laser altimeter accuracy could vary. A varied accuracy of the laser altimeter had consciences in that the shift used when trimming the data needed to be adjusted for each dataset. The adjustment of each dataset introduces one of few variable changes which need to be changed for each dataset and limits the usefulness of using trimming in the first place. The other variable which needed to be changed from the datasets was the percentage of the thresholding found from Otsu's method and used for Canny edge detection. The top layer prominence varied most likely due to different characteristics of

the snow, like air and water content or compression due to wind. Overall the need for parameter adjustment reduces the method's robustness.

Another aspect of the provided data that can present a problem for the detection and interpretation is the variation in altitude of the drone. The variation can create false tops and valleys due to abrupt changes in the altitude, creating difficult areas for the method. In addition, these altitude variations create a false sense of topography in the image. The topography we see in the provided GPR data is not the same as real-life topography, where it is often an artefact due to losing or gaining altitude. It is possible to correct for the variation of the altitude by flattening the estimated layer, see Figure 6.1. This flattening also, to some degree, removes variations in the topography, but it is ambiguous because the top snow layer is not an exact representation of the underlying ground. If there is a need to associate the radar data to the real-life topography, digital elevation models (DEM) can be used.

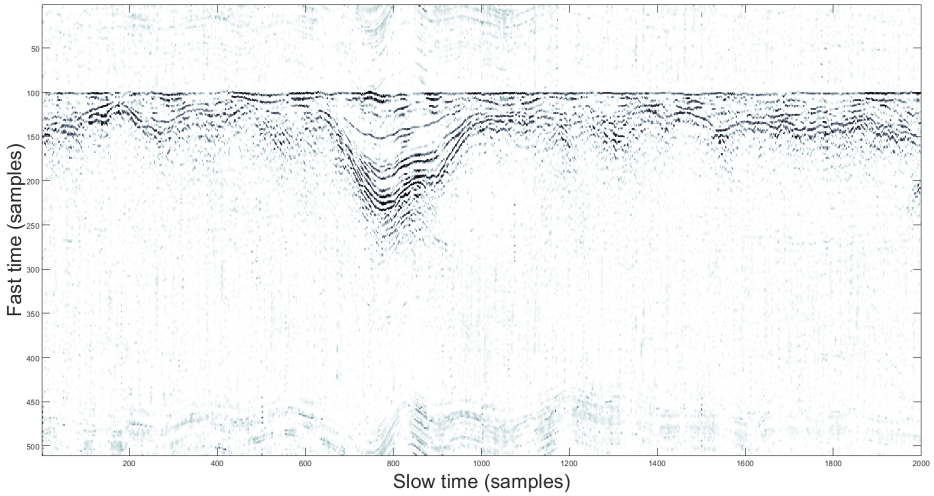


Figure 6.1: Flattening the top layer to see the underlying quasi-topography and layer thickness better.

6.2 Method Strengths and Limitations

Like any other method, the method presented in this thesis has its strengths and limitations. The method's strength in finding the top layer is that it is fast, accurate for the datasets shown and has a decent quality measure that marks the top layer estimate's chance of being correct based on the mean value of the surrounding pixels. Unfortunately, we had some unintended consequences

because normalisation was thought to be a reasonable choice for the quality estimate. Using normalisation individually for each subset resulted in some subsets with a generally high or low-quality value being repressed. Figure 5.1b, which is the subset with the highest accuracy in the whole dataset, shows that most of the left areas are questionable even though the top estimate is correct. This is a result of the normalisation where other places are affected but to a lesser degree. one way to fix this is to put all the quality values in a global vector before normalisation.

It should be noted that the speed of the code is based on pre-processed data where the bulk part of the computational load is skipped.

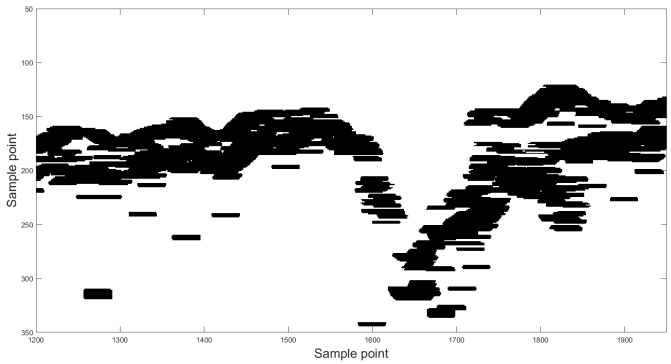
Early on, the decision fell on using dataset 1 as the base dataset when developing the method. Compared to the two other datasets, this dataset generally has a brighter top layer, possibly due to a higher snow density (wind packed) in the top layer. In addition, using dataset 1 could make detecting the first layer in the other datasets more challenging because the layer could blend into the ambient noise. The difference in the datasets was a challenge, but a middle ground was found by first developing the method for dataset 1 and then making adjustments by visiting datasets 2 and 3.

In how the method was developed, each step is essential for the subsequent step, except for some steps that do have an overlap, like the mean removal and laser altimeter trimming. Even though each step has its purpose and benefit, it also has its drawbacks. For example, the mean removal of each row in preparation of the datasets removes most of the noise. However, it also removes values from the layer of interest, especially generally flat intervals. Therefore, it would be beneficial to have another technique that removes the noise without influencing the layer of interest or removing the direct waves in the first place.

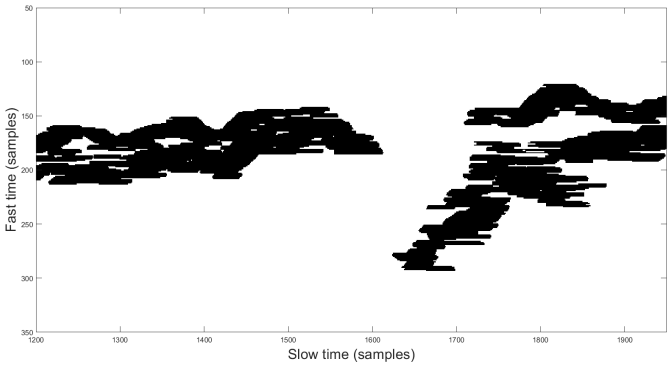
Another flawed step that could benefit from an improvement is the horizontal line dilation which tries to connect the layers. The limitation of this step is that it assumes that the next point in the top layer is horizontal to the other point. Assuming horizontality is a valid assumption of most places. It also introduces the problem where noise close to the top layer connects with the layer and that valleys that are shorter than the dilation distance are being cut prematurely. However, the main limitation of the method is that each step chosen and tuned was done by the developer's judgment and not by numerical measurements. The tuning has made the method work particularly well for these datasets but would probably need some adjustments for other datasets.

In dataset 2 subset 3 we can see that the top detection method has the highest RMSE of all datasets tested. We can see from Figure 5.3c that the problem which gives the dataset a lower accuracy value is the failure of the method

to follow the top layer down the drop in the data. This decrease in ground elevation could result from the drone adjusting its altitude, but as we can see from the GPS Figure 8.6 and Figure 5.3b and 5.3b from subsets 2 and 4, the drone is going over this feature multiple times. The failure to follow the layer is unexpected since it could detect the layer to a higher degree in the other datasets. If we assume that the feature, which could be a small stream of water, is approximately the same for the different drone crossings, it is concerning that we do get different results. That is because the number of times crossing an area should not matter, as long as the conditions are the same. When we take a closer look at the area in the subset, we can see where the problem arises. In Figure 6.2 we can see the two steps which result in the detection failure in the top layer. Here we can see that the layers fail to connect after dilation due to the descending layer, Figure 6.2a, leaving it small enough to be removed with area removal, see Figure 6.2b. As expected, removing parts of the top layer raises downstream problems, leaving us with a suboptimal estimate of the top layer.



(a) Patchy areas after dilation.



(b) Removal of areas below a pixel number.

Figure 6.2: Reason for detection failure in top estimate in Figure 5.3c. Dilation fails to connect the descent layer, leaving it small enough for area removal.

Part II

Finding Bottom Layer

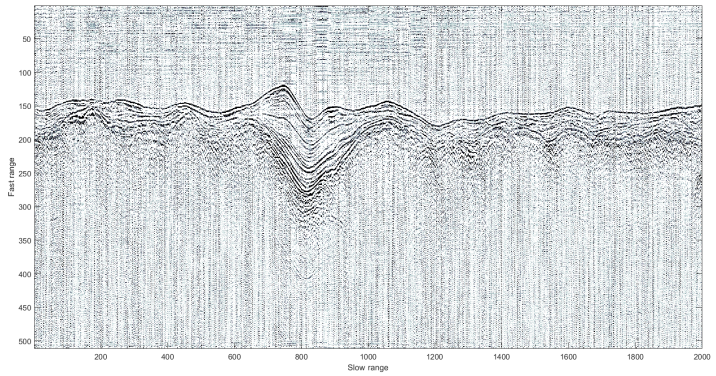
7

Method Development

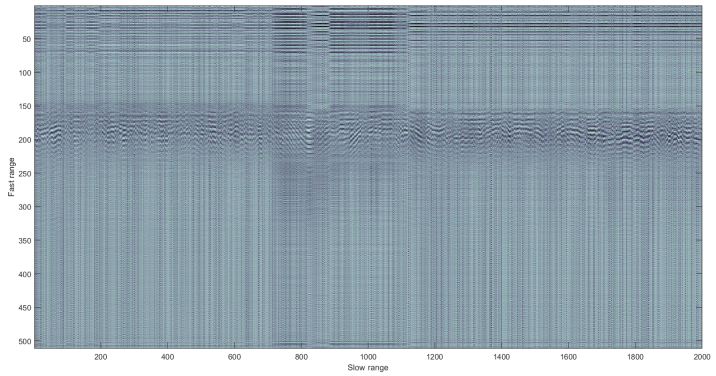
As mentioned in the Data Preparation for the top layer, finding the top and bottom layers differs because they have different characteristics. The top layer has a clear boundary between the transition from air to snow, excluding weak layers and noise. Contrarily, the bottom layer does not, and the transition from snow to the ground is not necessarily as abrupt. The reason is that the bottom layer appears thicker due to the hyperbolas' tail extent and the snow-ground interface's complexity. These variables make the bottom layer more complicated and the approach different from the top layer.

7.1 Preparation of Data

The steps needed to prepare the datasets are the same as the top layer. The only difference is in the tuning of the variables in the filtering. In the SVD filtering, the singular values removed is increased to 20, and the contrast stretch after filtering is decreased from 0.04 to 0.01. In Figure 7.1a the product of the resulting SVD filtering can be seen with the sum of the removed components displayed in Figure 7.1b.



(a) Result of SVD filtering.



(b) Removed components from SVD filtering

Figure 7.1: (a) Result of filtering by removing singular components on Figure 4.3. (b) Removed values from the SVD filtering.

The second change in the preparation of the dataset is that the median and wiener filter has been increased. The median filter has been increased from 3×3 to 4×4, and the wiener filter has been increased from 3×3 to 5×5. The result of the filtering can be seen in Figure 7.2 where the ground truth of the bottom layer is marked in blue.

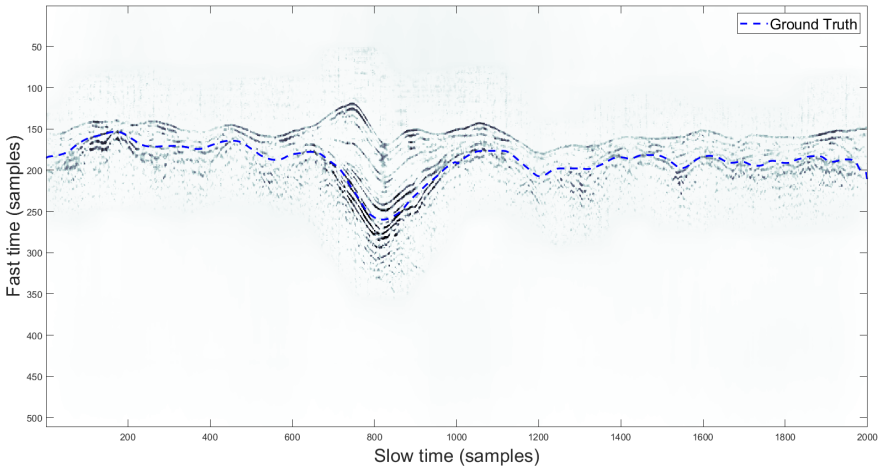


Figure 7.2: Prepared image for detection of the bottom layer. Ground truth shown in blue.

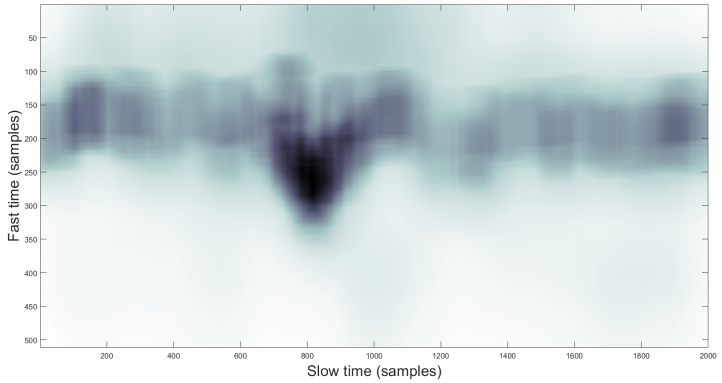
7.2 Layer Detection

The presumption in the last part where we found a method to find and detect the top layer was that it was bright and could distinguish itself from the foreground. However, the method presented in this section uses two presumptions. First, the hyperbolas tail extent of the bottom layer follows the top of the hyperbolas and could therefore be used to detect the layer. By looking at Figure 7.2, which displays the ground truth, we can see that the layers, even though it is coarse, correlate to each other. We will therefore play on this presumption in detecting the bottom layer. The second presumption is that the top layer is known, where the top layer's estimate from the last part will be used. From this point onwards in the thesis, the top and tail extent will refer to the top and bottom part of the hyperbolas, where the bottom layer would be used for the top extent of the hyperbola.

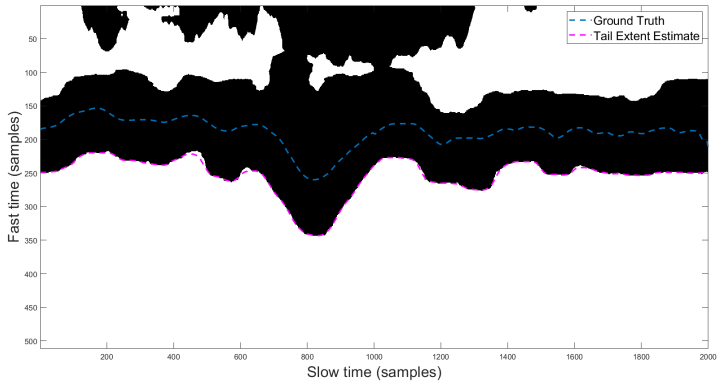
Our task would be to reduce the noise as much as possible in the area underneath the bottom layer to find the tail extent accurately. The noise has already been heavily reduced in the preparation of the data by the Wiener filter, but a mean filter of size 50×50 is also used to flatten isolated intensity values and to blur the tail extent, Figure 7.3a. The blurring is done to smooth the transition between intensity value changes in the tail extent and make thresholding more consistent. Mean filters of different shapes and sizes were tested, e.g. with a horizontal bias, but a square with size 50 was judged best. Since edges are unnecessary for this approach, binary thresholding is conducted instead of one

through Canny edge. The threshold value is calculated from Otsu's method, with no reduction of its value needed. The result of the thresholding with the subsets ground truth of the bottom layer can be seen in Figure 7.3b, notice how the tail extent correlates with the ground truth.

A closing morphological operator is then run over the image with a SE in the shape of a disk the size of 50 pixels. The closing operator is used to close the distance between layer regions separated because of the thresholding. The result of thresholding and closing on the subset can be seen in Figure 7.3b. With this image, a vector can be created of the tail extent estimate where points are smoothed with the regressive LOESS function.



(a) Smoothing.



(b) Thresholding.

Figure 7.3: Result of (a) filtering 7.3a by a mean filter of size 50×50 and (b) thresholding image with value chosen by Otsu's method. Ground truth is displayed in blue and the tail extent estimate is in pink.

With a vector representing the tail extent estimate, we can shift this vector up in

the image and cut out all values above it. We find the amount of shifting needed by empirically finding the mean separation between the bottom bottom-layer and the ground truth of the bottom layer. Through testing, these values became around 55 for multiple subsets, which is used to shift the vector value upwards. In addition to shifting the vector by the mean difference, the vector is also shifted one standard deviation of the tail extent estimate upwards. The extra shift makes it so that the vector is not directly on top of the bottom ground truth, but it finds itself a bit above. The extra shift is done because we do want to remove values above this vector value in the original image in Figure 7.2 in the hopes of arriving at an approximation of the bottom value from "above". To eliminate the chance that the bottom estimate is above our top estimate, values that are above the top estimate and 20 pixels down are removed by setting them to zero. The trimming of the top layer also removes the relatively strong signal from the top layer, leaving us with signals from internal layers. Figure 7.4 shows the result of the trimming.

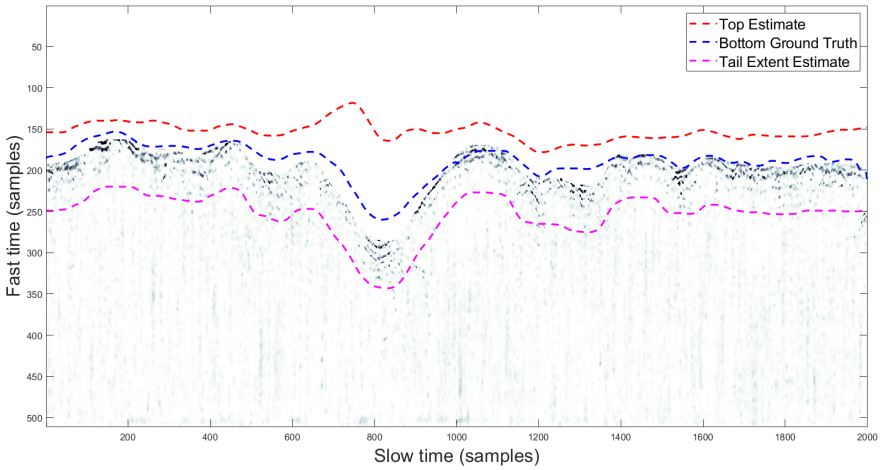
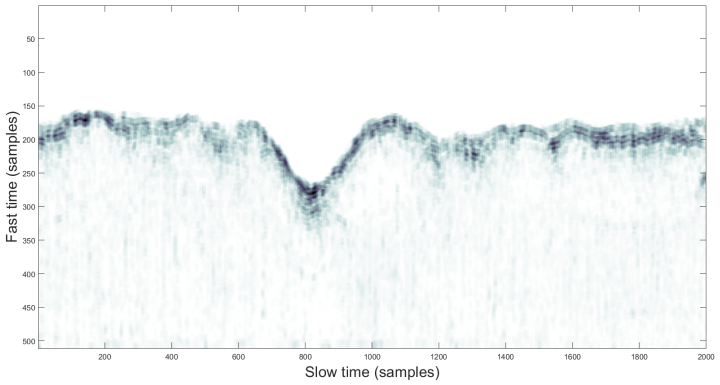
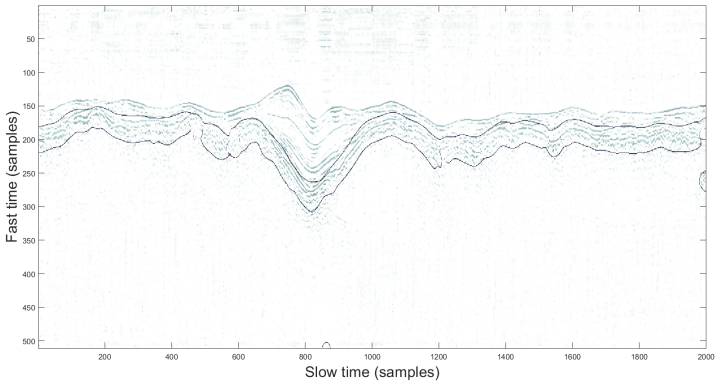


Figure 7.4: Figure 7.2 trimmed by moving tail extent estimate (pink) up by its standard deviation and the mean value between it and bottom ground truth (blue). 20 samples down from top estimate (red) are also removed.

An estimation of the bottom layer could be approximated with the given trimming, but an attempt at filtering out weak internal layers will be made. Using the trimmed image, we apply the mean filter as used before, with a filter size of 15×15. The filtered image is then put through Canny edge with the threshold from Otsu's method and a standard deviation of 15 to connect and smooth edges. Results can be seen in 7.5.



(a) Smoothing.



(b) Canny.

Figure 7.5: Filtering image 7.4 with a (a) mean filter of size 5×5 and then using (b) Canny edge on the result.

Like has been done before in the process of extracting the estimated vector, the bottom estimate is gathered from the top of the Canny edge image and smoothed with regressive LOESS function. The result of the bottom estimate, and this method, is shown in Figure 7.6 with the ground truth of the section. A quality estimate like the one in the top layer was tried implemented but was removed because it was considered too inaccurate

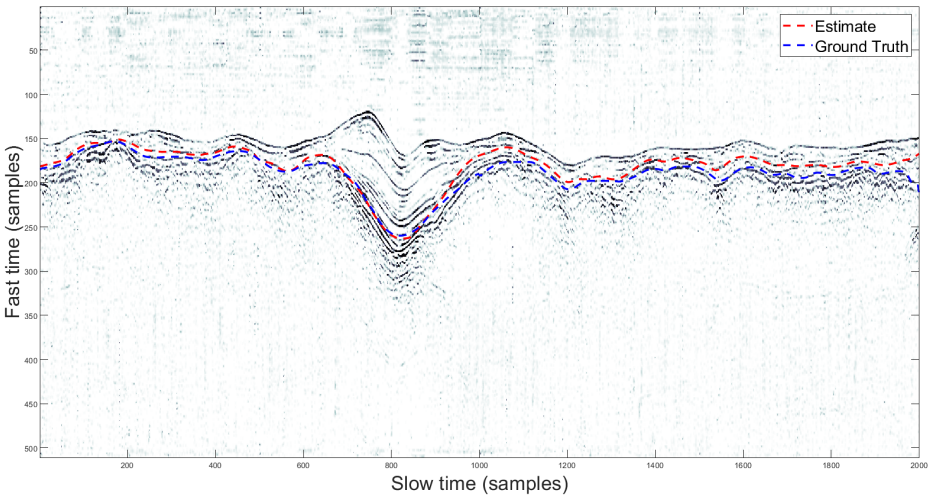
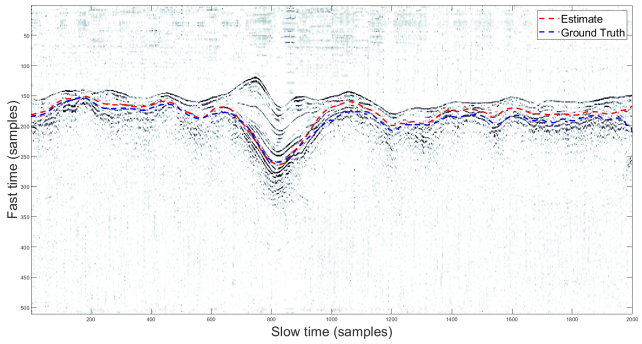


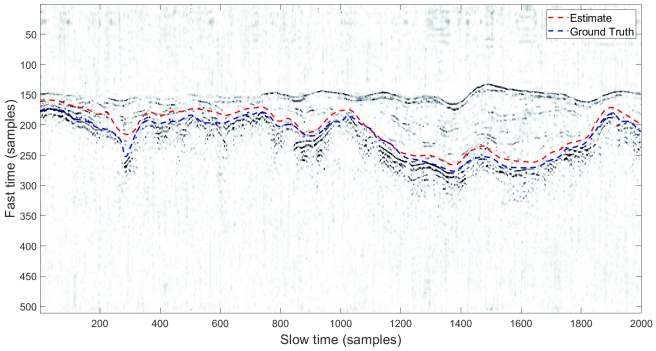
Figure 7.6: Result of the bottom estimate method (red) shown with the ground truth (blue).

8

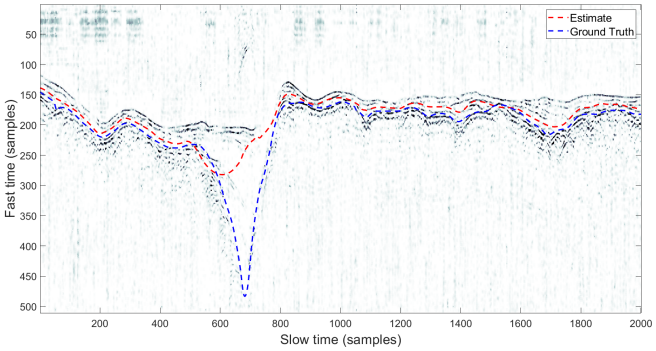
Applied Results



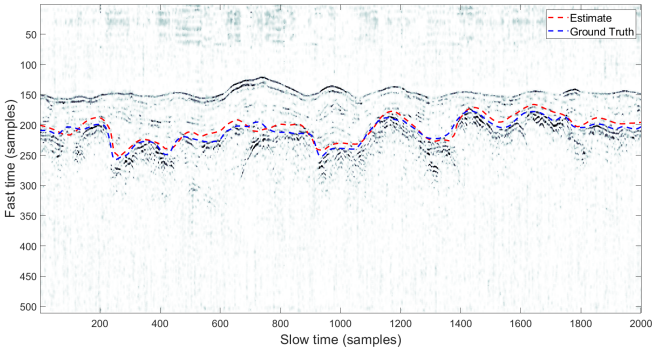
(a) Subset 1, 16'000-18'000.



(b) Subset 2, 13'000-15'000.



(c) Subset 3, 950-2950.



(d) Subset 4, 8800-10'800.

Figure 8.1: Bottom results from dataset 1 with their given ground truth and estimate from the method.

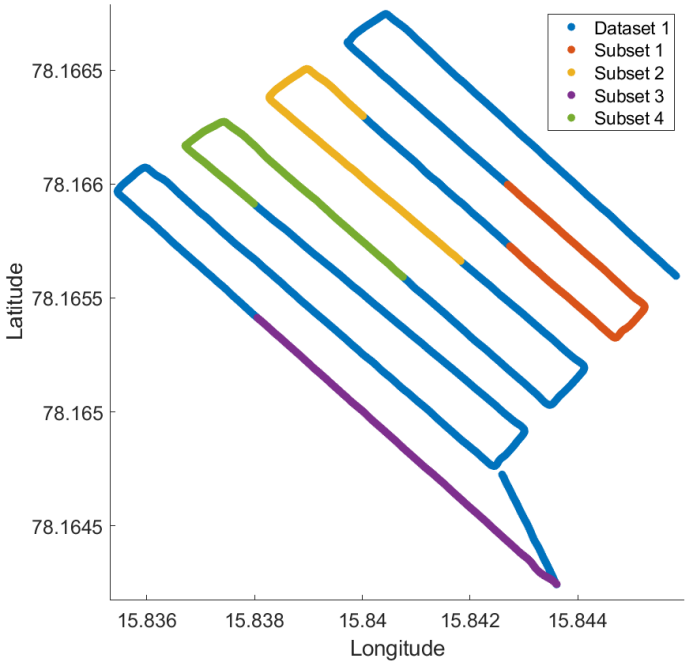
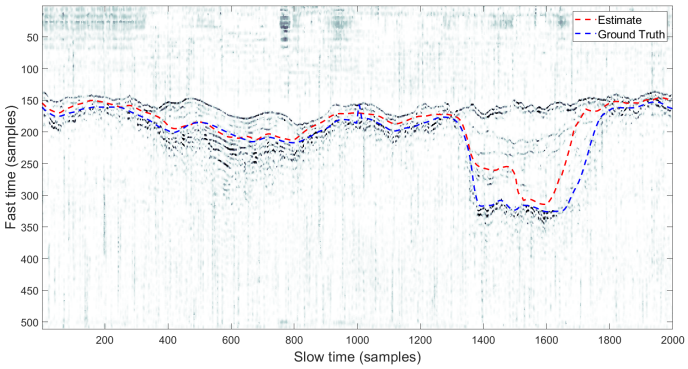


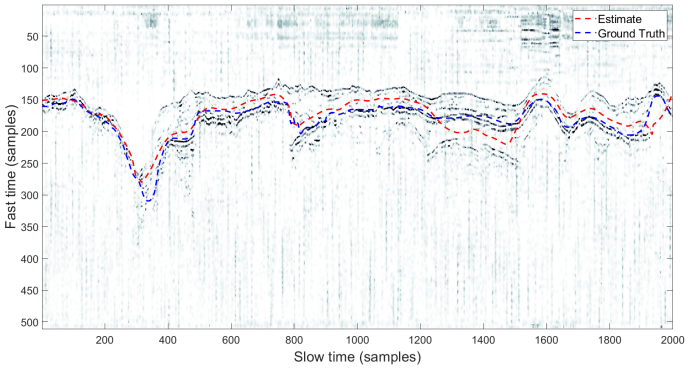
Figure 8.2: Flight pattern for Dataset 1 with the different subsets marked.

Table 8.1: Bottom RMSE in air equivalent cm and samples, and Cross correlation for dataset 1.

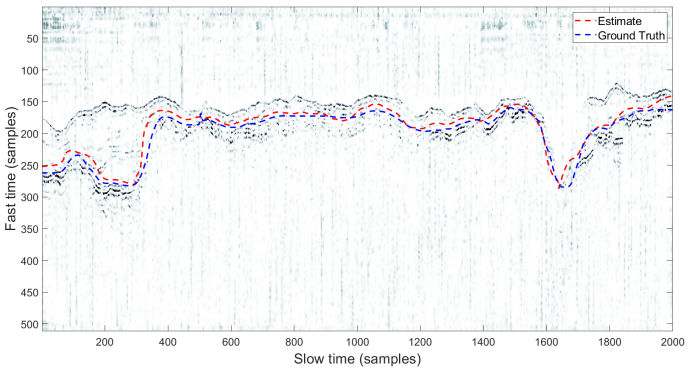
Subset	RMSE (cm)	RMSE (samples)	Cross correlation
Subset 1	12.3504	10.9425	0.9514
Subset 2	14.5877	12.9160	0.9848
Subset 3	45.0193	39.9044	0.8020
Subset 4	9.9372	8.7880	0.9396



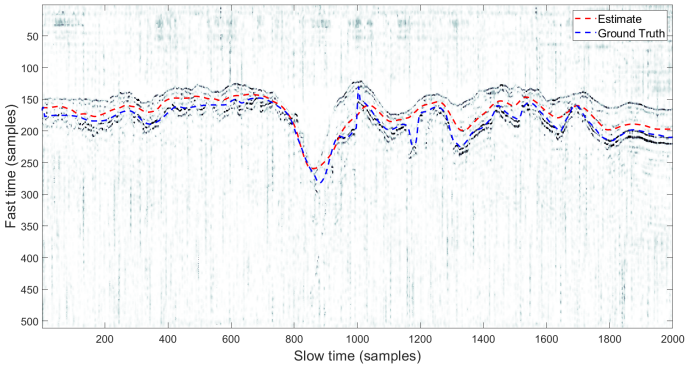
(a) Subset 1, 6000-8000.



(b) Subset 2, 8100-10'100.



(c) Subset 3, 12'500-14'500.



(d) Subset 4, 25'001-27'000.

Figure 8.3: Bottom results from dataset 2 with their given ground truth and estimate from the method.

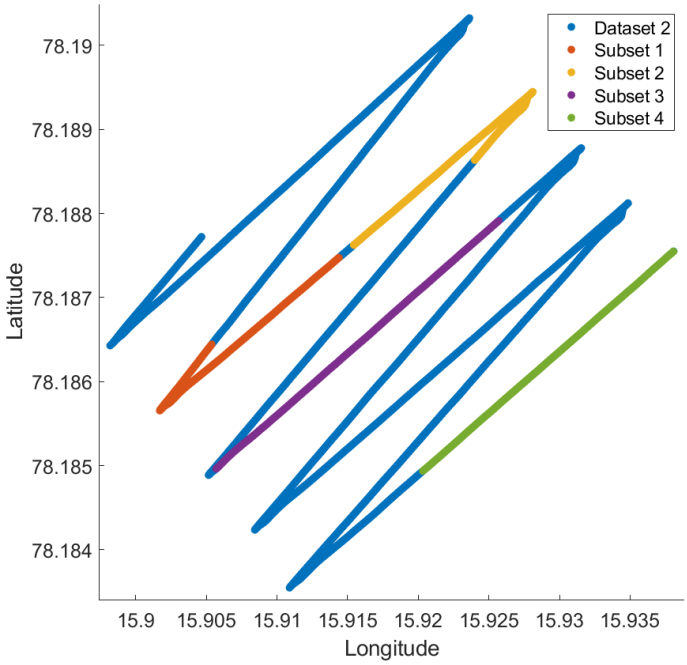
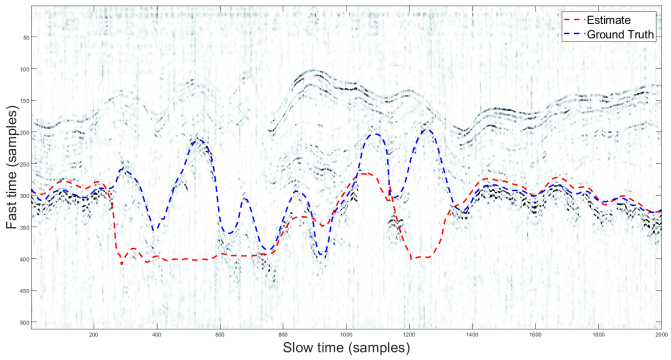


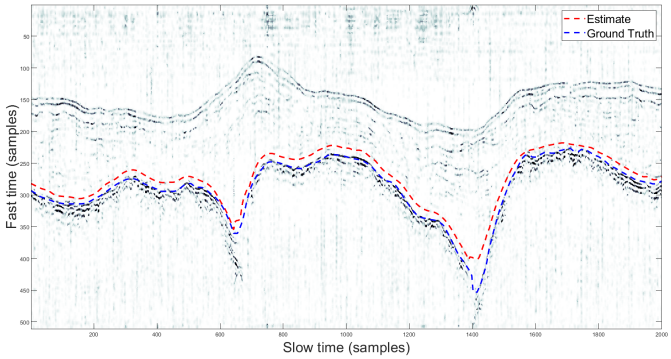
Figure 8.4: Flight pattern for Dataset 2 with the different subsets marked.

Table 8.2: Bottom RMSE in air equivalent cm and samples, and Cross correlation for dataset 2.

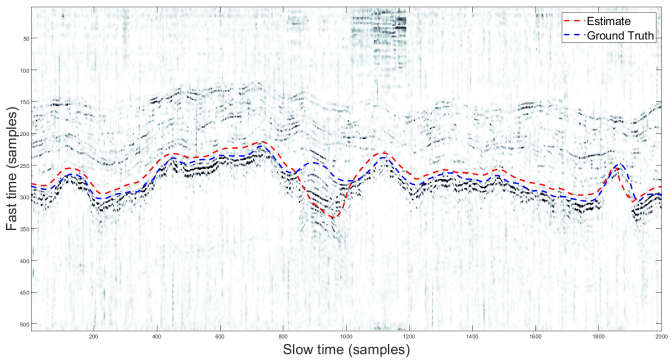
Subset	RMSE (cm)	RMSE (samples)	Cross correlation
Subset 1	28.7721	25.4965	0.9297
Subset 2	17.9384	15.8762	0.8655
Subset 3	12.9376	11.4329	0.9736
Subset 4	16.1574	14.2994	0.9484



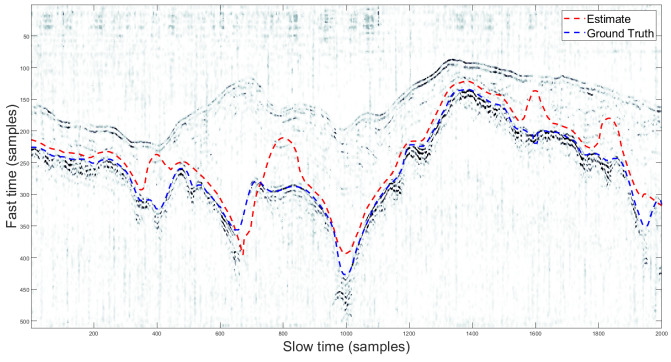
(a) Subset 1, 7600-9600.



(b) Subset 2, 17'600-19'600.



(c) Subset 3, 23'400-25'400.



(d) Subset 4, 26'200-28-200.

Figure 8.5: Bottom results from dataset 3 with their given ground truth and estimate from the method.

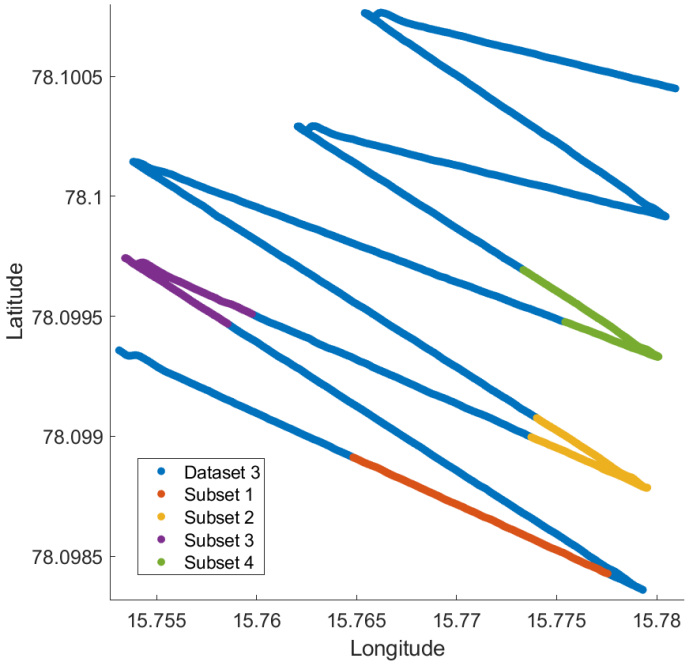


Figure 8.6: Flight pattern for Dataset 3 with the different subsets marked.

Table 8.3: Bottom RMSE in air equivalent cm and samples, and Cross correlation for dataset 3.

Subset	RMSE (cm)	RMSE (samples)	Cross correlation
Subset 1	81.3504	72.0804	0.0445
Subset 2	17.2242	15.2628	0.9916
Subset 3	21.6267	19.1556	0.7188
Subset 4	34.2525	30.3408	0.9389

9 Discussion

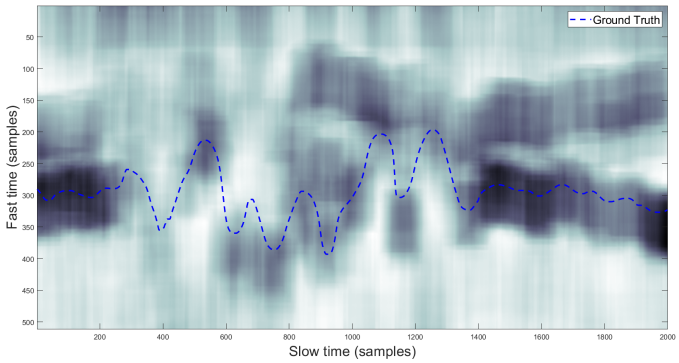
In the detection method used to detect the bottom layer, no other new techniques were introduced compared to the top layer. However, the techniques were implemented in different ways, order and values, because of the different nature of the layers. The problematic elements of the detection of the bottom layer, which were the same for the top detection, were the tuning of parameters made by human judgement and the mean removal. Mean removal, which could remove intensity values of the layers in an attempt to reduce the noise, also reduced portion of the bottom layer, as could be seen from the histogram in Figure 4.2. However, the mean removal did not impact the method equally because the bottom method heavily blurred the layer before use.

9.1 Method Strengths and Limitations

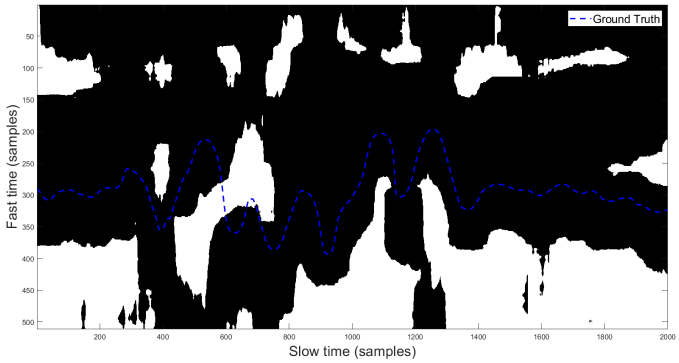
The method's main advantage is that it can follow the bottom layer and detect variation in the ground, although not that accurately. On the other hand, the results from the method were able to follow the bottom layers with a high cross-correlation, which could be useful where the accuracy of up to 20 cm (air equivalent) is tolerated. It should also be noted that there are some areas in most of the datasets which show a high RMSE and cross-correlation where the error rate disproportionately arise from single events in the subsets. For example, we can see this in dataset 1 subset 3, dataset 2 subset 1 and dataset 3 subset 1 and 4. Looking at one of them, dataset 3 subset 1, we can try to

understand where in the method it goes wrong and what could be done about it.

By running through the method step by step, we can see what causes such a wrong estimate of the bottom layer. In the first step, after the dataset's preparation, we find the problem. Here we do extensive mean smoothing to find the tail end of the hyperbolas for the bottom layer. Unfortunately, because of the size of the smoothing, the considerable amount of noise, and the complexity of the subset, the smoothing fails to do its intended purpose. In Figure **??** we can see the images of the smoothing and where the thresholding includes a lot of the noise. For reference, both the images are depicted with the ground truth. In Figure 9.2 we do correct the step by lowering the mean smoothing, resulting in a reasonable bottom estimate in Figure 9.2a. Making this correction the new standard in the method would be tempting, but using a low mean smoothing on the other subsections results in worse outcomes. By looking at the other subsection from the last paragraph, it was observed that the low estimate accuracy resulted from the thresholding and the standard deviation shift in the trimming. Therefore, tuning both these parameters would be necessary for achieving the best result using this method.

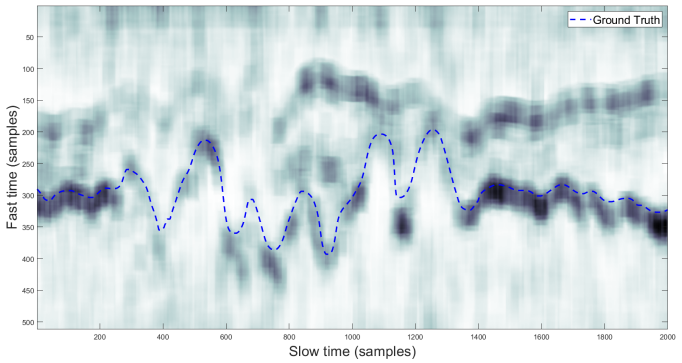


(a) Smoothing by 50×50.

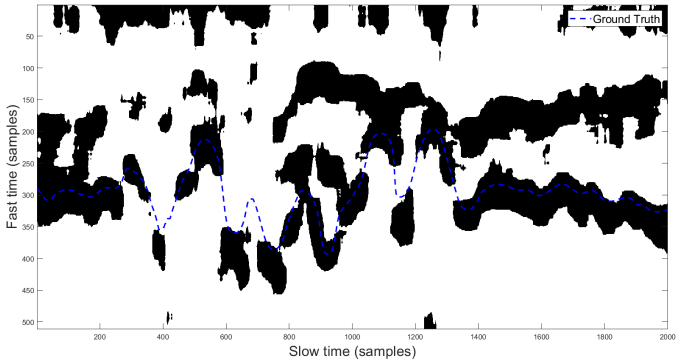


(b) Thresholding using Otsu's method.

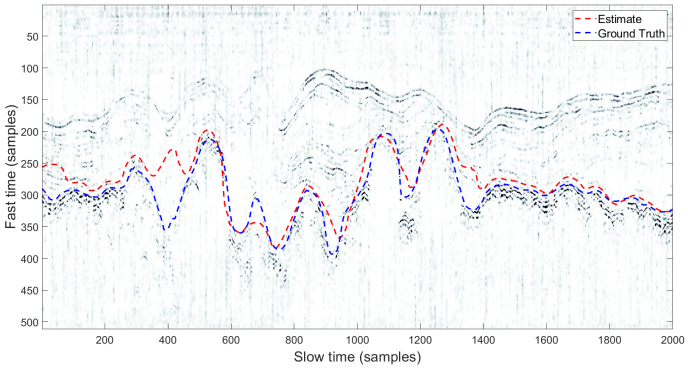
Figure 9.1: Steps in the bottom method where the estimate is shifted in dataset 3 subset 1.



(a) Smoothing by 20×20.



(b) Thresholding using Otsu's method.



(c) New estimate.

Figure 9.2: Fixing the bottom estimate by lowering the mean smoothing at the start.

The main problems unique to the bottom layer detection lie in both its assumptions: the top estimate is accurate, and the hyperbolas' tail extent follows the top of the hyperbolas. Both of these assumptions are used when trimming the

data. The thought of using trimming based on the top estimate was due to it being reasonably accurate, preventing the situation where the bottom layer could be above the top layer and the removal of the strong intensity values coming from the top layer. This assumption worked well in the datasets tested in this thesis, with some exemptions. If there were some areas of deviation, it was usually because the top estimate vent above the top layer, resulting in a less trimmed top layer for the bottom estimate. The other assumption was that the tail extent of the returning waves hyperbola followed the top of the hyperbolas. This assumption worked well for most places but struggled in sections where the bottom layer had a more complex snow-ground boundary. Even though it is not confirmed or certain, the boundaries it did well were probably where the transmitted waves had a strong reflection boundary, e.g. snow to ground or bedrock.

In the top estimate, the choice of using dataset 1 in the method development could be one of the reasons why it did poorer in the two other datasets because it had a stronger intensity in the top layer. On the other hand, in detecting the bottom layer, the choice of dataset did not matter on the grounds of intensity value. However, it did have a difference because of altitude changes between the datasets and the complexity of the layers. The bottom layer is more susceptible to sharp altitude changes when it comes to the topography variation. In contrast, the top layer gets a smoothed representation of the topography because of the snow. Therefore, an increased bottom layer altitude variation which we could see in dataset 3, could be one of the reasons why the bottom method did poorer in the dataset contra dataset 1.

10

Final Conclusion

In this thesis, an automatic layer detection method was developed to detect the top and bottom layers in snow-packed terrain. The method developed used, amongst others, Canny edge detection and morphological operators as its main elements in detecting the layers. The two layers were developed separately from each other because they had different characteristics, but they did follow the same primary principles.

The detection method of the top layer was accomplished by reducing the noise in the image and then utilising the distinct intensity increase from air to the snow layer. Using this method accomplished a RMSE accuracy below the range resolution of the radar system of about 5cm. It was achieved by representative testing subsections, excluding outliers and signal dropouts. A quality estimate was also given to the top layer, but it did have limitations due to the implementation of the normalisation. Therefore, it should only be used as a quick assessment of the quality and not a concluding estimate. For future improvement, one way to fix this is to put all the quality values in a global vector before normalisation.

The detection method of the bottom layer was performed by going after two assumptions: the top estimate is accurate, and the hyperbolas' tail extent follows the bottom layer. The top estimate is necessary because it is used in trimming the data, and tracking is the primary method used in combination with heavy smoothing. This resulted in a high RMSE value of about 20 samples but a high cross-correlation value of 0.9 in most subsets. For that reason, a

quality estimate was not given to the bottom estimate since it was considered inaccurate.

The overarching goal of the thesis was to find a method that could detect the primary interfaces from GPR data collected from an airborne drone over a snow-pack. This thesis automatically accomplished the detection, with predictable minor variable adjustments needed for the best result. Furthermore, the method presented could now be applied routinely to estimate the primary interfaces in other GPR data, where previous no method existed.

10.1 Future Improvements

As stated before, the goal of this thesis was to find an automatic layer detection method that could detect the primary interfaces in a snow-packed terrain. Furthermore, as a subgoal, we wanted to find a solution to the main problem where a focus would be held on the early model-based method, and an expansion into a more complex method would be explored if needed. This thesis showed that detecting the primary interfaces was possible to a varying degree of success using simple hands-on and model-based methods. The main problem with the solution was that certain variables needed to be changed to get a good result. The changing of variables goes against the complete automatisation of the interpretation of GPR data, but the method could be a prototype were to build upon its concepts.

To further increase the accuracy in future improvements, an automatisation process of the known variables which need change could be implemented. For example, it could look into where the problem originates, from the snow density, drone speed or other parameters in the original data, and take them into account. On the other hand, an expansion into a more complex method could be conducted. As discussed in the introduction, areas worth exploring are, e.g., machine learning, which has been shown in other areas to be fast and accurate. In addition, it could also be possible to use the CFAR adaptive algorithm, which could improve the base data used for detection based on statistical properties in the data. Using CFAR could potentially make the method presented in this thesis more viable since the noise was a significant factor in the result of the outcome.

Bibliography

[1] Rolf-Ole Rydeng Jenssen. Radar system development for drone borne applications with focus on snowpack parameters. 2021.

[2] D Schneider. Small drone probes antarctic ice with radar. *IEEE Spectrum, Tech Talk*, 2014.

[3] Emily Arnold, Carl Leuschen, Fernando Rodriguez-Morales, Jilu Li, John Paden, Richard Hale, and Shawn Keshmiri. Cresis airborne radars and platforms for ice and snow sounding. *Annals of Glaciology*, 61(81):58–67, 2020.

[4] Mrinal K Sen, Paul L Stoffa, Roustam K Seifoullaev, and Jacob T Fokkema. Numerical and field investigations of gpr: toward an airborne gpr. *Subsurface Sensing Technologies and Applications*, 4(1):41–60, 2003.

[5] Irwin Sobel. An isotropic 3x3 image gradient operator. *Presentation at Stanford A.I. Project 1968*, 02 2014.

[6] David Marr and Ellen Hildreth. Theory of edge detection. *Proceedings of the Royal Society of London. Series B. Biological Sciences*, 207(1167):187–217, 1980.

[7] John Canny. A computational approach to edge detection. *IEEE Transactions on pattern analysis and machine intelligence*, (6):679–698, 1986.

[8] Daniel Rueckert and Julia A Schnabel. Model-based and data-driven strategies in medical image computing. *Proceedings of the IEEE*, 108(1):110–124, 2019.

[9] Scott Konishi, Alan L. Yuille, James M. Coughlan, and Song Chun Zhu. Statistical edge detection: Learning and evaluating edge cues. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25(1):57–74, 2003.

[10] David R Martin, Charless C Fowlkes, and Jitendra Malik. Learning to detect natural image boundaries using local brightness, color, and texture cues. *IEEE transactions on pattern analysis and machine intelligence*, 26(5):530–549, 2004.

[11] Pablo Arbelaez, Michael Maire, Charless Fowlkes, and Jitendra Malik. Contour detection and hierarchical image segmentation. *IEEE transactions on pattern analysis and machine intelligence*, 33(5):898–916, 2010.

[12] Saining Xie and Zhuowen Tu. Holistically-nested edge detection. In *Proceedings of the IEEE international conference on computer vision*, pages 1395–1403, 2015.

[13] Augusto Cunha, Axelle Pochet, Hélio Lopes, and Marcelo Gattass. Seismic fault detection in real data using transfer learning from a convolutional neural network pre-trained with synthetic seismic data. *Computers & Geosciences*, 135:104344, 2020.

[14] Sinno Jialin Pan and Qiang Yang. A survey on transfer learning. *IEEE Transactions on knowledge and data engineering*, 22(10):1345–1359, 2009.

[15] Samer Lahouar and Imad L Al-Qadi. Automatic detection of multiple pavement layers from gpr data. *NDT & e International*, 41(2):69–81, 2008.

[16] Jun Zhang, Xing Yang, Weiguang Li, Shaobo Zhang, and Yunyi Jia. Automatic detection of moisture damages in asphalt pavements from gpr data with deep cnn and irs method. *Automation in Construction*, 113:103119, 2020.

[17] Kien Dinh, Nenad Gucunski, and Trung H Duong. An algorithm for automatic localization and detection of rebars from gpr data of concrete bridge decks. *Automation in Construction*, 89:292–298, 2018.

[18] Zhongming Xiang, Abbas Rashidi, and Ge Ou. An improved convolutional neural network system for automatically detecting rebar in gpr data. In *Computing in Civil Engineering 2019: Data, Sensing, and Analytics*, pages 422–429. American Society of Civil Engineers Reston, VA, 2019.

[19] P.P. Gandhi and S.A. Kassam. Analysis of cfar processors in nonhomogeneous background. *IEEE Transactions on Aerospace and Electronic Systems*, 24(4):427–445, 1988.

[20] Louis L Scharf. Detection, estimation and time series analysis. *Statistical Signal Processing*, 1991.

[21] Rafael C Gonzalez, Richard E Woods, and Barry R Masters. Digital image processing third edition. *Journal of Biomedical Optics*, 14(2):029901, 2008.

[22] Norbert Wiener. A heuristic exposition of wiener's mathematical theory of prediction and filtering. 1964.

[23] Deepika Sharma and Pawanesh Abrol. Svd based noise removal technique: An experimental analysis. *International Journal of Advanced Research in Computer Science*, 3(5), 2012.

[24] Nobuyuki Otsu. A threshold selection method from gray-level histograms. *IEEE transactions on systems, man, and cybernetics*, 9(1):62–66, 1979.

[25] Rolf Ole Rydeng Jenssen, Markus Eckerstorfer, and Svein Jacobsen. Drone-mounted ultrawideband radar for retrieval of snowpack properties. *IEEE Transactions on Instrumentation and Measurement*, 69(1):221–230, 2019.

[26] Rolf Ole R Jenssen and Svein Jacobsen. Drone-mounted uwb snow radar: technical improvements and field results. *Journal of Electromagnetic Waves and Applications*, 34(14):1930–1954, 2020.