Probabilistic graphical techniques for automated ice-bottom tracking and comparison between state-of-the-art solutions

by

Victor Berger da Silva

Submitted to the graduate degree program in Electrical Engineering and Computer Science and the Graduate Faculty of the University of Kansas in partial fulfillment of the requirements for the degree of Master of Science.

_____

Chair: Carl Leuschen

_____

Co–Chair: John Paden

_____

Guanghui Wang

Date Defended: May 10th, 2019

The thesis committee for Victor Berger da Silva

certifies that this is the approved version of the following thesis:

Probabilistic graphical techniques for automated ice-bottom tracking and comparison between state-of-the-art solutions

_____

Chair: Carl Leuschen

_____

Co–Chair: John Paden

Date Approved: May 17th, 2019

Abstract

Multichannel radar depth sounding systems are able to produce two-dimensional and three-dimensional imagery of the internal structure of polar ice sheets. One of the relevant features typically present in this imagery is the ice-bedrock interface, which is the boundary between the bottom of the ice sheet and the bedrock underneath. Crucial information regarding the current state of the ice sheets, such as the thickness of the ice, can be derived if the location of the ice-bedrock interface is extracted from the imagery. Due to the large amount of data collected by the radar systems employed, we seek to automate the extraction of the ice-bedrock interface and allow for efficient manual corrections when errors occur in the automated method.

We present improvements made to previously proposed solutions which pose feature extraction in polar radar imagery as an inference problem on a probabilistic graphical model. The improvements proposed here are in the form of novel image pre-processing steps and empirically-derived cost functions that allow for the integration of further domain-specific knowledge into the models employed. Along with an explanation of our modifications, we demonstrate the results obtained by our proposed models and algorithms, including significantly decreased mean error measurements such as a 43% reduction in average tracking error in the case of three-dimensional imagery.

We also present the results obtained by several state-of-the-art ice-interface tracking solutions, and compare all automated results with manually-corrected ground-truth data. Furthermore, we perform a self-assessment of tracking results by analyzing the differences found between the automatically extracted ice-layers in cases where two separate radar measurements have been made at the same location.

Acknowledgements

The radar data used in this work were collected as a part of NASA Operation IceBridge. The radar system and SAR processing software were developed with support from the University of Kansas, NSF grant ANT-0424589, and NASA Operation IceBridge grant NNX16AH54G. The layer tracking work was supported on NSF Big Data grant 1838236 and NSF grant ACI-1443054.

ArcticDEM was provided by the Polar Geospatial Center under NSF OPP awards 1043681, 1559691 and 1542736.

We also acknowledge the contributions of many CReSIS faculty, staff, and students who contributed to the radar system development and to NASA aircraft and instrument teams, especially ATM who provided GPS data, for flying and supporting the mission.

Para meu pai e minha mãe.

Table of Contents

Chapter 1 - INTRODUCTION AND LITERATURE REVIEW

1.1 - Background and motivation

Experts in climatological, geophysical and polar research have reached full agreement that the large glacial bodies of the planet, such as Antarctica and Greenland, have undergone a process of accelerated melting over the past several decades [1 - 4]. This is of extraordinary concern to the international scientific community due to the significant social and environmental impacts of sea level rise generated by melting ice, many of which can already be observed today.

Estimates of the annual polar ice mass discharge and contribution to sea level vary according to the methodology and data acquisition processes employed in different studies; however, researchers agree that both have significantly increased in the 20th century. Based on combined satellite altimetry, interferometry, and gravimetry datasets, some scientists argue that since 1992 polar ice melting has caused global sea level to rise, on average, by 0.59 ± 0.20 millimeters per year [2], whereas other studies indicate this number may be as high as 2 millimeters per year [3]. Furthermore, recent research indicates that the Antarctic ice sheet discharge increased from 40 ± 9 gigatonnes/year in 1979-1990 to 252 ± 26 gigatonnes/year in 2009-2017 [4].

Several models and simulations that allow scientists to understand and predict the current and future ice sheet dynamics have been proposed. These models often involve ice-surface mass balance and conservation, ice flow, and discharge rates [5, 6, 7]. In order for these models to be formulated, large amounts of data must be directly acquired from polar regions. Ice sheet data are often repeatedly acquired from the same locations over the years so that direct calculations of the variations in ice sheet volume and ice velocity can be performed [8] and so that the long-term simulations can be evaluated. One of the main measurements required for assessing the mass

balance of the ice sheets of a particular region is the ice thickness [8], which we focus on throughout this work.

Several techniques have been employed over the years for the purpose of mapping and imaging the polar ice sheets. Ground-penetrating radar depth sounding systems have been developed for the purpose of imaging both the exposed and the subterranean structure of the polar ice sheets in an efficient manner. Since the 1980s, radar depth sounding (RDS) systems have been developed at the University of Kansas, as part of research efforts by the Center for Remote Sensing of Ice Sheets (CReSIS) and previously the Radar Systems and Remote Sensing Laboratory (RSL). Gradual improvement in RDS instrumentation led to the development of the Multichannel Coherent Radar Depth Sounder (MCoRDS) [9, 10], which is able to provide wide-coverage sounding and imaging of polar ice sheets with fine spatial resolution.

Using this sensor, CReSIS is able to image the internal structure of polar ice sheets to a depth of several kilometers. MCoRDS collects polar data from airborne platforms such as the Lockheed P-3 Orion and the Douglas DC-8 aircraft operated by the National Aeronautics and Space Administration (NASA).

1.2 - Radar imagery

We use MCoRDS data to form images of the subterranean ice topography. The radar system employs a cross-track antenna array where each antenna element is individually sampled. Data from each element are independently pulse-compressed to resolve targets in the range dimension and synthetic aperture radar (SAR) processed to resolve targets in the along-track dimension. Finally, the individual SAR images from each element are combined using array processing to resolve targets in the elevation angle dimension [11]. The reader is referred to [11,

12, 13] for further information regarding the data acquisition and signal processing techniques employed. All radar data presented in this document were acquired with MCoRDS and processed in the manner briefly presented above. Note that minor differences exist in the radar systems and data processing parameters used in different CReSIS deployments, which lead to minor differences in the imagery generated from each deployment season.

Data products from this processing include two-dimensional SAR images where only the nadir elevation angle is resolved and three-dimensional SAR tomographic images where targets from all cross-track angles are resolved. The *nadir* direction is defined as pointing straight down, and therefore it is orthogonal to the surveyed terrain if the terrain is perfectly flat.

In the nadir-looking two-dimensional echograms generated by CReSIS, the ice structures are captured along the flight profile. Fig. 1.1 illustrates the image axes with respect to the aircraft, and Fig. 1.3 shows a two-dimensional echogram generated by CReSIS from data collected as part of the 2017 Operation IceBridge deployment to Greenland. The horizontal axis represents the along-track dimension, where each column is a range-line. The vertical axis corresponds to the fast-time dimension, where each row is a range bin. The vertical dimension is directly related to the depth of the subterranean ice structure. The pixel intensity of the radar image is proportional to radar scattering intensity, with darker representing a stronger scattering signal. The array processing algorithm for 2D imagery uses a filter to estimate the intensity of targets in the nadir direction while suppressing targets from off nadir. The Minimum Variance Distortionless Response (MVDR) algorithm is used for all 2D images in this work [12].

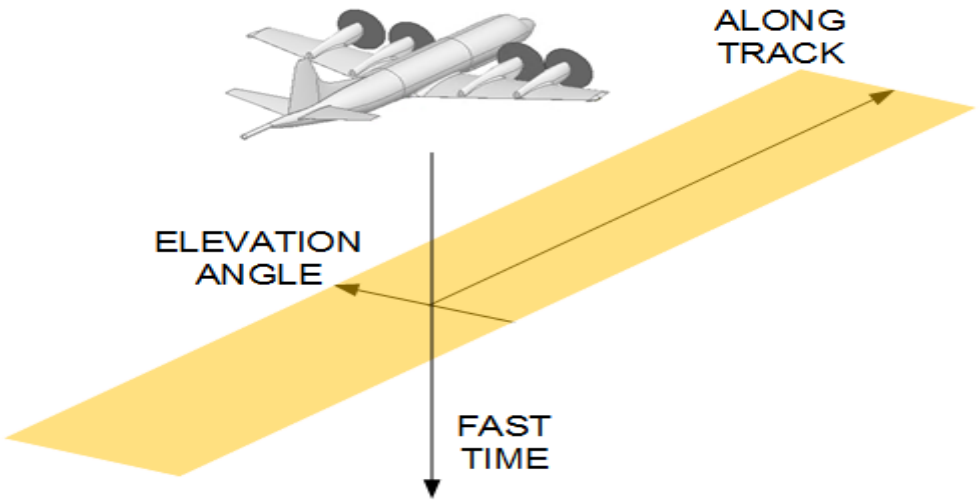

Fig. 1.1. Illustration depicting the axes of the radar imagery relative to the radar platform. Nadir-looking two-dimensional echograms are vertically oriented and lie in the plane formed by the fast time and along-track axes.

The three-dimensional data products consist of sequences of two-dimensional cross-track images (or "slices") of the terrain. In each slice, the horizontal axis displays the radar elevation angle discretized into direction-of-arrival (DoA) bins, and the vertical axis depicts the fast-time dimension in the same manner as a two-dimensional echogram where each row corresponds to a range-bin. In this work, the terms "DoA bin" and "elevation angle bin" are synonymous. The MUltiple SIgnal Classification (MUSIC) algorithm is used to generate the 3D images [13]. The color of each pixel represents the MUSIC cepstrum which is loosely related to the scattering intensity from a particular DoA bin. See Fig. 1.5 for a single two-dimensional slice of the three-dimensional imagery generated by CReSIS from data collected as part of the 2014 Operation IceBridge deployment to Greenland. In that figure, yellow indicates a larger cepstrum value which is associated with increased scattering intensity, and blue indicates a lower cepstrum value. Each slice of the 3D imagery represents a single range-line of SAR imagery, and thus corresponds to one column of the two-dimensional echogram depicting the same surveyed terrain. The 2D image

is a proper subset of the 3D image and does not have an elevation angle dimension; it corresponds to just the nadir elevation angle bin from each 3D slice except that the array processing method to obtain the image is different. See Fig. 1.2 for a diagram demonstrating how an array of consecutive cross-sectional 2D slices forms a three-dimensional image.

In CReSIS data products, a data segment typically corresponds to the data collected during entire flights. Data segments are divided into data frames, which usually correspond to roughly 50 kilometers (measured in the along-track dimension) of surveyed terrain each. Most examples of 2D imagery presented in this document depict entire data frames, such as Fig. 1.3.

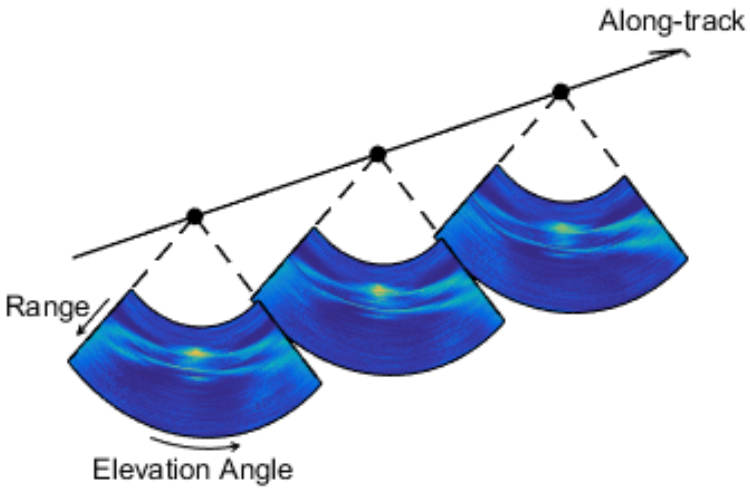

Fig. 1.2. A sequence of cross-track "slices" generates a 3D image of the surveyed terrain.

1.3 - Ice layers

In both two-dimensional and three-dimensional formats of imagery generated by CReSIS, the most prominent features are the ice-surface and ice-bottom layers. The ice-surface layer is the interface between the air and the top surface of the ice sheet. The ice-bottom layer is the interface between the bottom of the ice-sheet and the bedrock or liquid water underneath it.

No ice-bottom layer is visible in portions of the flights that are not over ice. Because the radar radio waves suffer increased power attenuation while traveling through the ice, the ice-bottom boundary, where it exists, may be relatively faint and difficult to detect. See Fig. 1.5 and Fig. 1.7 for examples of this, along with the correct labeling of the ice-bottom layers in Fig. 1.6 and Fig. 1.8 respectively.

In slices of 3D imagery, the response from the ice-bottom interface is typically of lower quality toward the grazing angles (leftmost and rightmost edges of individual slices). This is because the radar signal must travel farther to reach the ice in cases where the surface topography is relatively flat, the beam pattern is steered straight down at nadir so that less energy is sent toward grazing angles, the angular resolution of the horizontally oriented antenna array worsens toward the end-fire case, and the backscatter tends to be less for larger angles of incidence.

In contrast, a strong radar response is typically received from the ice-surface. The terrain underneath the ice is rich and heterogeneous, and can present varied topographies including flat regions, mountains, and valleys. In contrast, a much smoother interface is usually expected for the ice-surface, due to the gradual accumulation of ice and snow over thousands of years which generally fills in any low spots in the terrain.

For many glaciological research purposes, the locations of both ice-surface and ice-bottom layers must be obtained from the imagery, which allows for the slope, thickness, and other

properties of the ice to be numerically computed. This information can be used in generating digital elevation models (DEMs) of the imaged terrain.

Labeling of the layers in two-dimensional imagery has traditionally been performed by hand, since the ice boundaries are usually easily identifiable to a trained annotator. Manual tracking normally consists of hand-selecting sparse points corresponding to the ice boundaries, followed by a linear interpolation of the selected points in order to form a continuous layer. In some cases where the ice-bottom scattering is a distinct layer, a simple active contour model, also known as a *snake* model, is also used. This routine starts at the manually labeled points and tracks the highest intensity value within a manually selected range-bin window around the labeled point's range-bin for each of the neighboring range-lines on each side of the manual point. This process is repeated with the newly tracked range-lines, and then again with those range-lines, until all the range-lines have a tracked value.

However, due to the large amount of data acquired by modern ice imaging systems, precise manual tracking of two-dimensional echograms is a highly time-consuming process even with the aid of the simple active contour model mentioned above. In the case of three-dimensional imagery, manual labeling of thousands of sequential slices to generate a three-dimensional model of the ice-bed is not feasible. For this reason, significant effort is being spent to develop automated ice layer tracking systems for both 2D and 3D data formats.

Between the two aforementioned data products, there exists a fundamental difference that impacts the design and architecture of automated layer trackers: a strong correlation between consecutive slices of 3D imagery is expected, as they represent adjacent cross-sections of the surveyed terrain. For this reason, in most state-of-the-art software for three-dimensional ice reconstruction, tracking is performed in batches of consecutive slices, in order to exploit the along-

track correlation between them. On the other hand, no similar "third dimension" exists in the case of two-dimensional imagery, and therefore each echogram is processed individually.

In this work, we propose several improvements and adaptations to two existing automated layer tracking solutions. The first, initially proposed by Crandall *et al*. [14], is based on the construction of a Hidden Markov Model (HMM) framework that defines a probabilistic cost minimization model conditioned on 2D images, with the Viterbi algorithm [15, 16] being employed to perform exact inference on the HMM. The second, applied only to three-dimensional imagery, is based on a Markov Random Field (MRF) framework conditioned on a 3D image. This ice-bottom tracking method was initially proposed by Xu *et al.* [17]. While performing exact inference on MRFs is NP-hard in the general case, a number of *approximate* algorithms have been developed for this purpose. The sequential tree-reweighted message passing (TRW-S) algorithm [18] is employed for approximate inference on the MRF. The result of the inference step performed by either of the aforementioned algorithms is the answer that will maximize the probability of correctly labeling the ice-bottom interface conditioned on the input imagery; such probabilities are calculated from a combination of single-pixel and transition cost functions. The majority of the improvements proposed in this work are in the form of refinements to the existing cost functions, with the purpose of integrating further domain-specific knowledge and *a priori* information about the ice-sheets into the calculations. A detailed explanation about the theory and implementation of these methods is provided in Chapter 2.

Tracking of the ice-surface boundary is generally a trivial task, especially because accurate estimates exist for most regions surveyed by CReSIS instrumentation. Because of this, we make the assumption that the ice-surface is labeled *a priori* and only the ice-bottom layer must be tracked in both 2D and 3D echogram formats. For ice-surface information in 2D echograms, we use a

threshold tracker occasionally guided by the ArcticDEM [19] and Reference Elevation Model of Antarctica (REMA) [20] DEMs. For the 3D imagery, we rely entirely on the ArcticDEM and REMA DEMs for ice-surface information. The available ice-surface layer information is used by the automated trackers proposed in this work as an additional source of evidence and as a constraint for the location of the ice-bottom boundary.

Several existing automated layer tracking techniques attempt to determine the locations of both ice-surface and ice-bottom layers simultaneously, rather than using the data available for the ice-surface layer. Thus, these approaches do not require the location of the ice-surface layer to be known *a priori*. Existing solutions also widely differ regarding which inputs *must* be passed in to the layer tracking software, and which inputs *may* be optionally passed in if available. A review of existing layer tracking approaches, as well as their inputs and outputs, is provided in the next section.

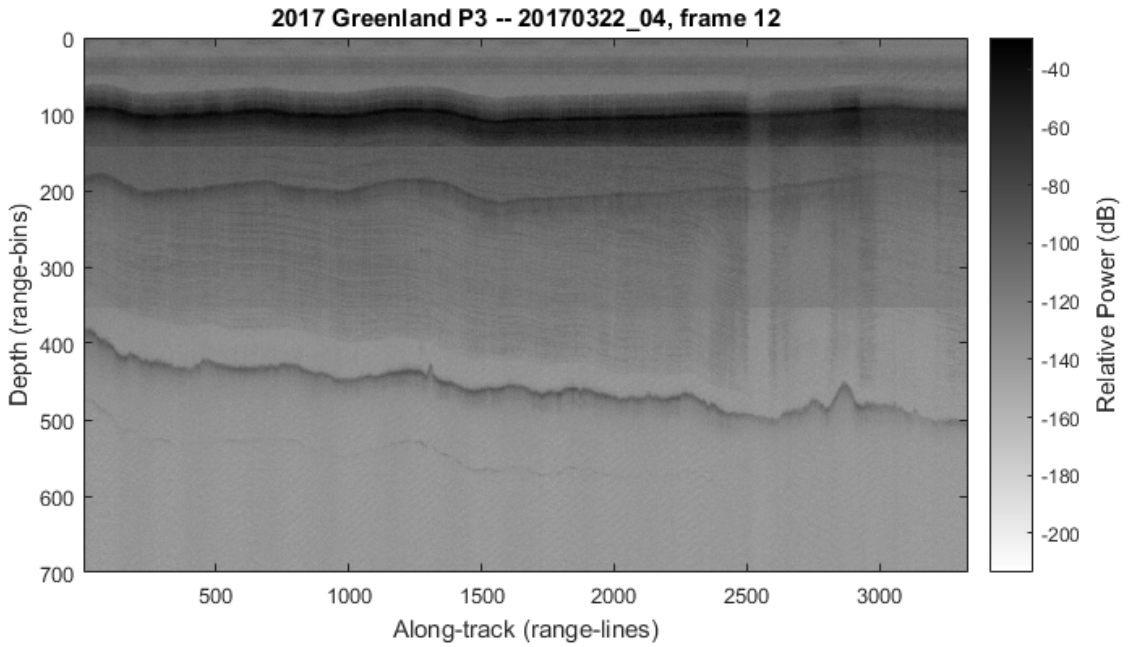

Fig 1.3. An example of a two-dimensional radar echogram generated from data acquired from MCoRDS as part of the NASA Operation IceBridge deployment to Greenland in 2017.

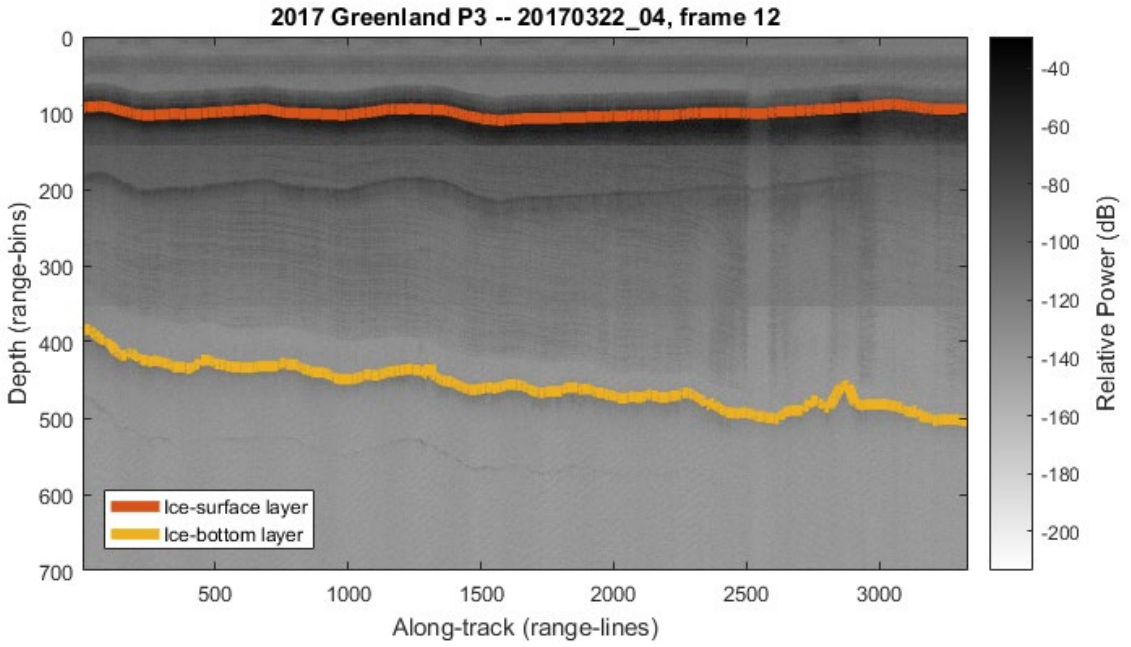

Fig 1.4. The same echogram as presented in Fig 1.3 above. The ice-surface and ice-bottom layers have been overlaid on the echogram, and are indicated by the continuous orange and yellow lines, respectively. This color coding for the correct ice-surface and ice-bottom layers is standard throughout this document.

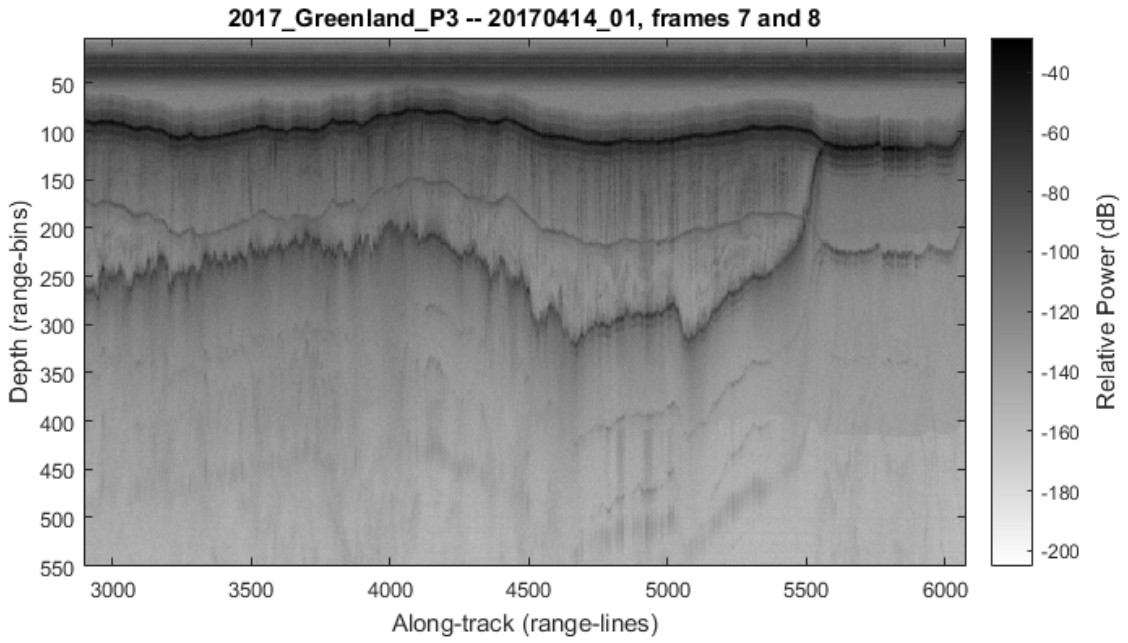

Fig 1.5. An example of a two-dimensional radar echogram generated from two adjacent data frames acquired from MCoRDS as part of the NASA Operation IceBridge deployment to Greenland in 2017. Notice how this example displays a more rugged ice-bottom layer than the previous.

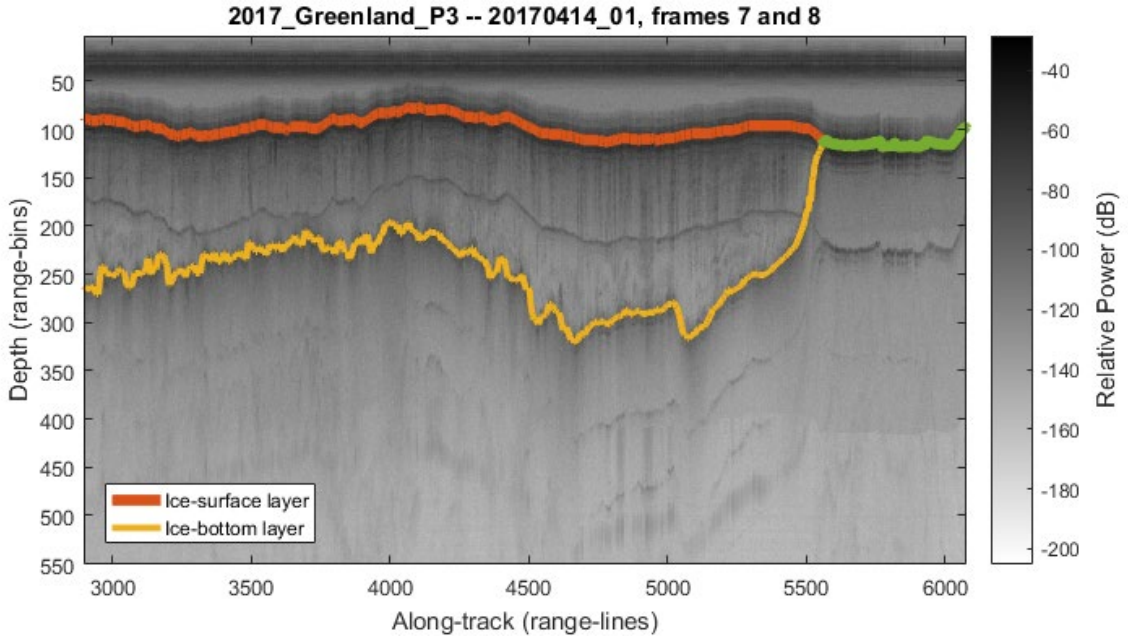

Fig 1.6. The same echogram as presented in Fig 1.5 above. The ice-surface and ice-bottom layers have been overlaid on the echogram. Notice that the two layers merge when no ice is present around range-line 5,500. This segment is indicated by the green line.

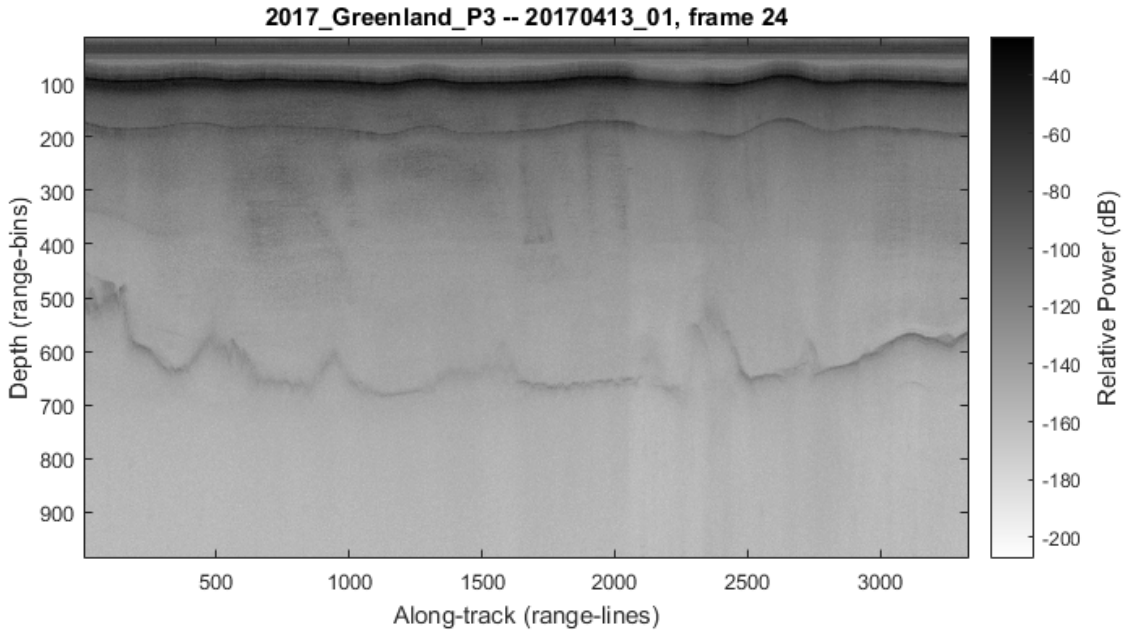

Fig 1.7. An example of a two-dimensional radar echogram generated from data acquired from MCoRDS as part of the NASA Operation IceBridge deployment to Greenland in 2017. Notice that the ice-bottom interface in this image is relatively faint and blurry in several areas.

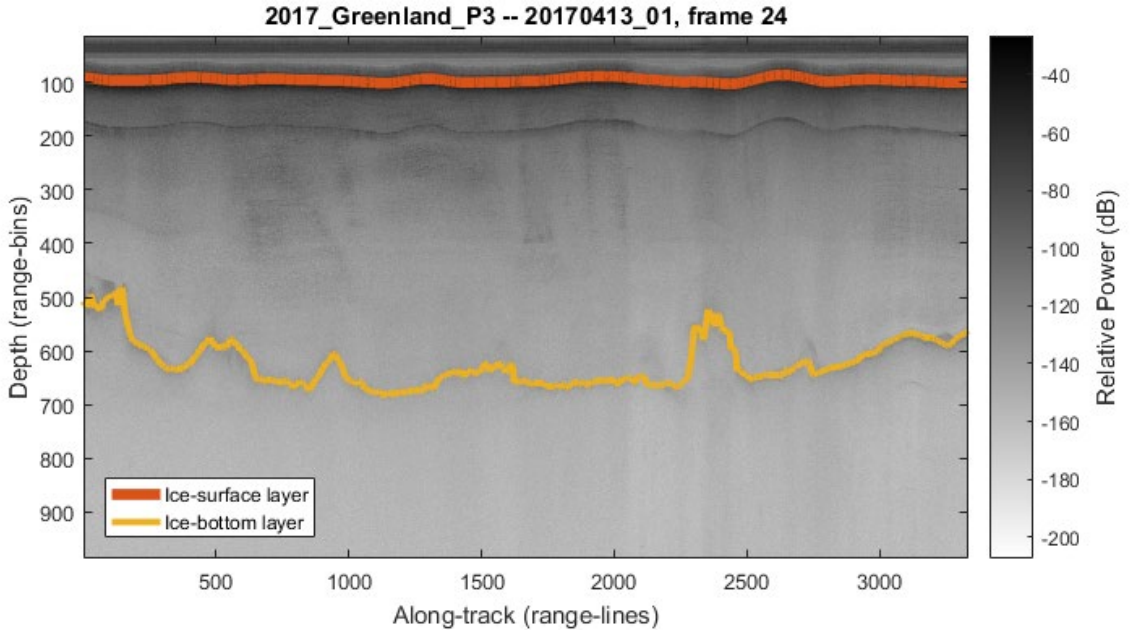

Fig 1.8. The same echogram as presented in Fig 1.7 above. The ice-surface and ice-bottom layers have been overlaid on the echogram. The improvements we propose to the automated layer trackers aim to improve the ice-bottom detection rate even in unfavorable situations such as the one presented above.

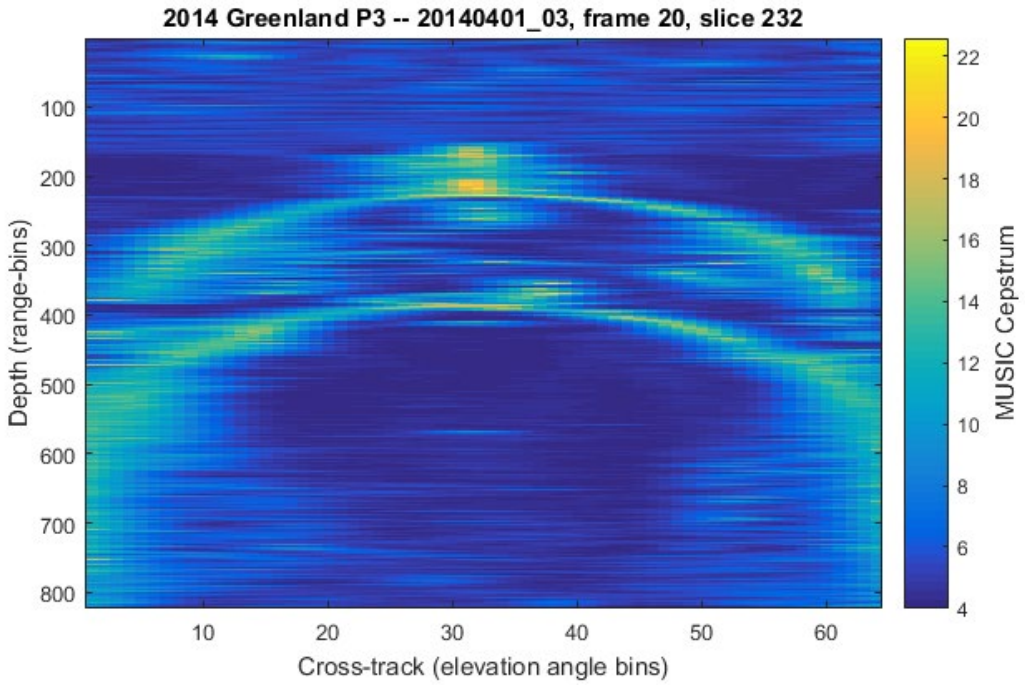

Fig 1.8. Example of a single slice of three-dimensional imagery generated from data acquired from MCoRDS as part of the NASA Operation IceBridge deployment to Greenland and the Canadian Arctic Archipelago in 2014.

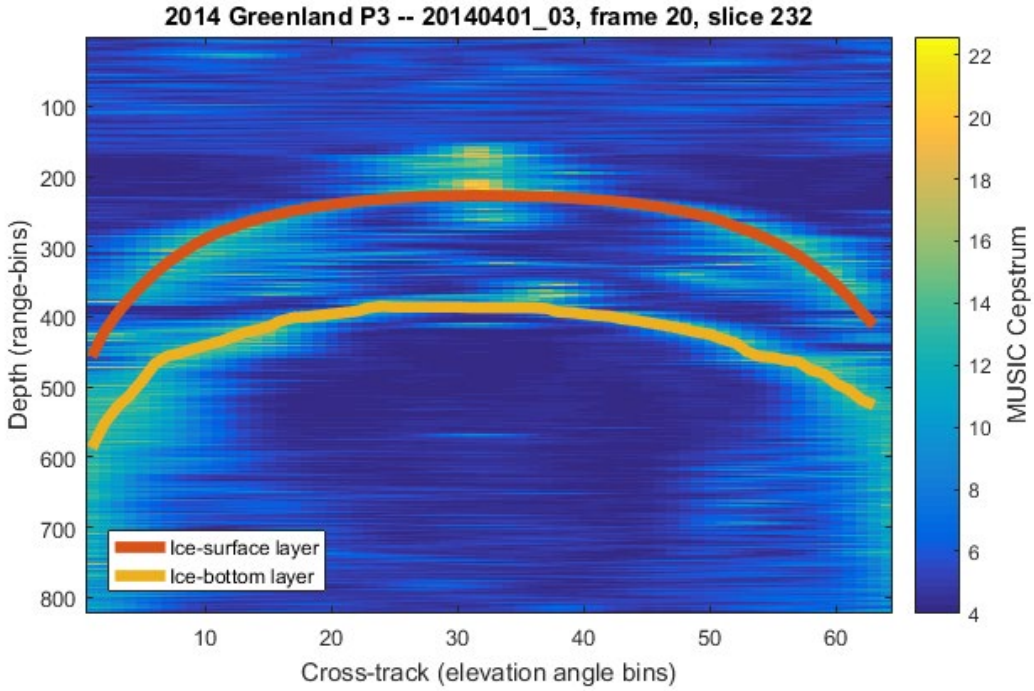

Fig 1.9. The same slice as presented in Fig 1.8 above. The ice-surface and ice-bottom layers have been overlaid on the image and are depicted with the standard color scheme.

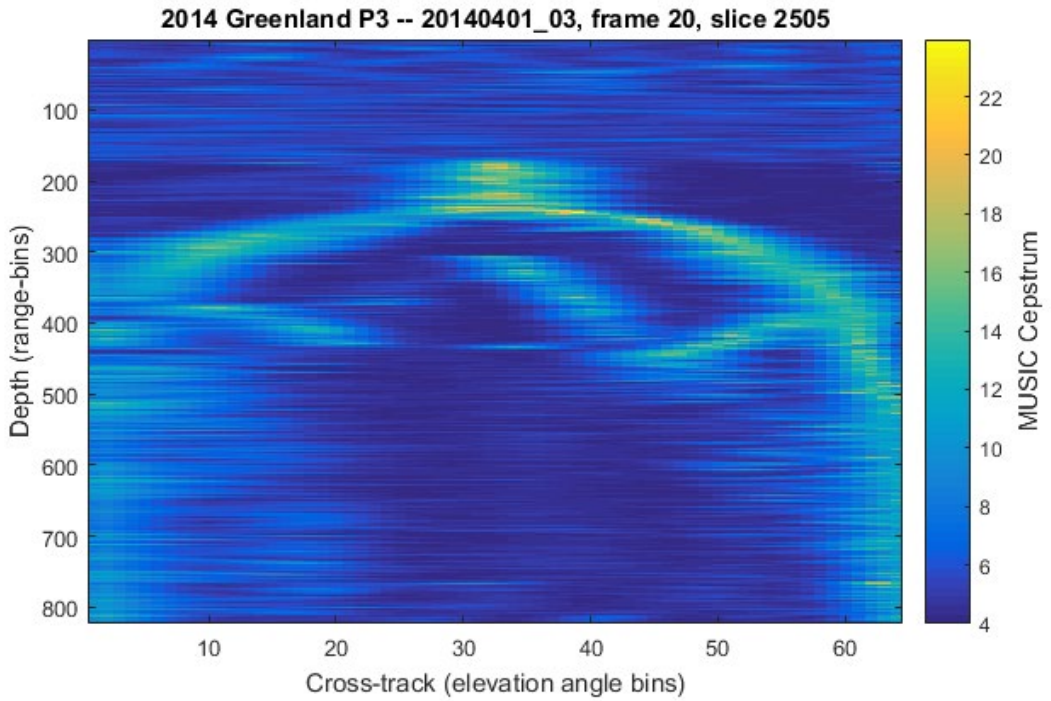

Fig 1.10. Example of a single slice of three-dimensional imagery generated from data acquired from MCoRDS as part of the NASA Operation IceBridge deployment to Greenland and the Canadian Arctic Archipelago in 2014. Notice also how this example displays a more rugged ice-bottom layer than the previous.

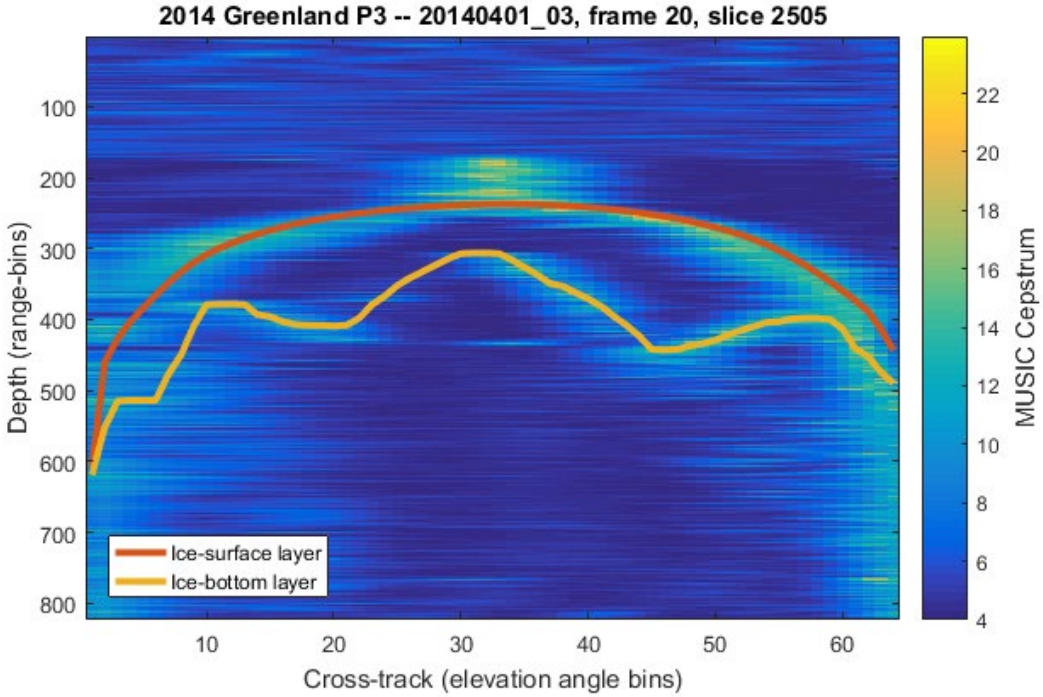

Fig 1.11. The same slice as presented in Fig 1.10 above. The ice-surface and ice-bottom layers have been overlaid on the image and are depicted with the standard color scheme.

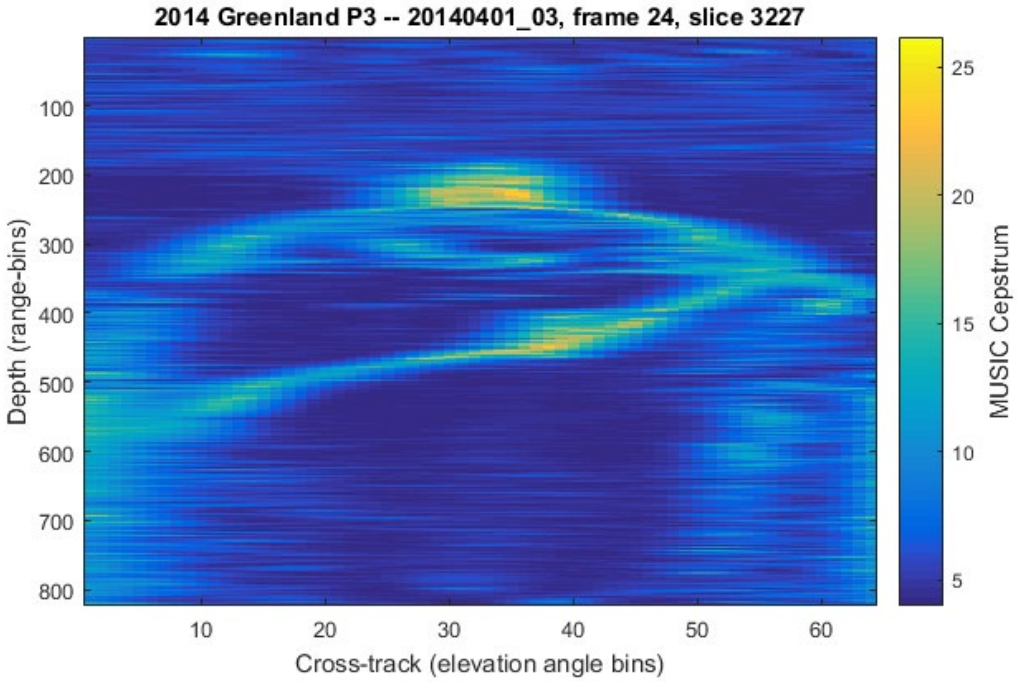

Fig 1.12. Example of a single slice of three-dimensional imagery generated from data acquired from MCoRDS as part of the NASA Operation IceBridge deployment to Greenland and the Canadian Arctic Archipelago in 2014.

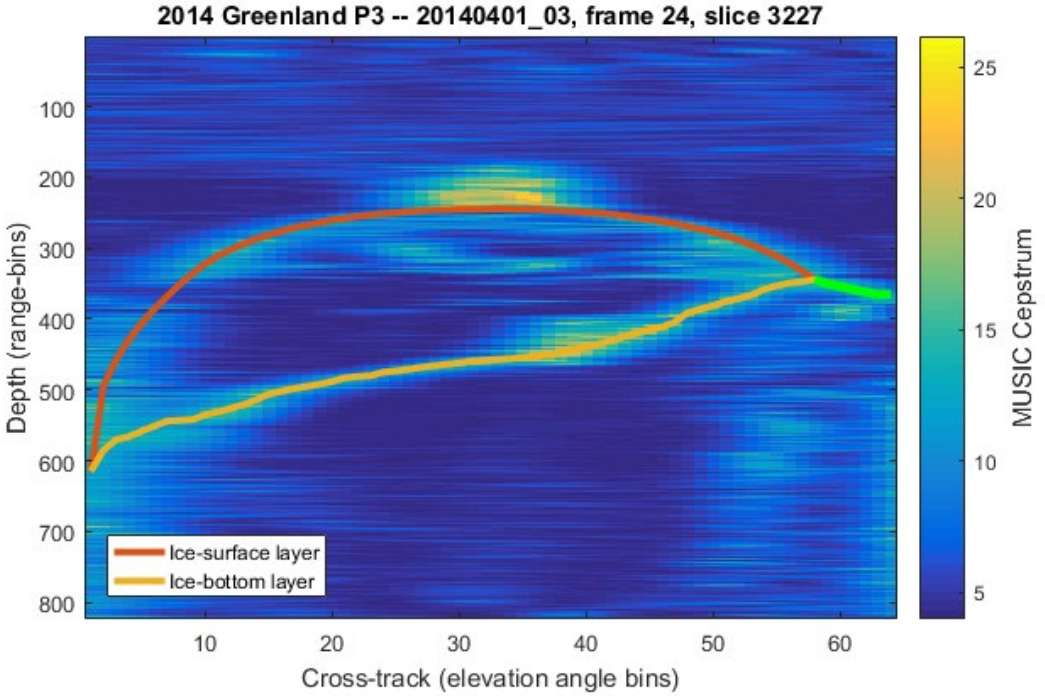

Fig 1.13. The same slice as presented in Fig 1.12 above. The ice-surface and ice-bottom layers have been overlaid on the image and are depicted with the standard color scheme. Notice that the two layers merge when no ice is present around direction-of-arrival bin 57. This segment is indicated by the green line.

1.4 - Review of existing approaches for automated ice layer tracking

The large majority of all layer tracking solutions presented to date focus on two-dimensional data, and are not designed to take advantage of the additional information available in the 3D case.

One of the first efforts regarding automated layer tracking solutions specifically designed for radar echograms of polar ice sheets, Gifford *et al*. [21] propose two methods for detecting both ice boundaries from two-dimensional imagery while receiving no input other than the radar echogram itself. The fist solution is a simple gradient-based edge detection approach employing detrending routines and image filtering for noise suppression. A more robust approach, based on the active contour technique, is also suggested.

As a successor to the aforementioned active contour technique, a distance-regularized level-sets approach was proposed by Rahnemoonfar *et al*. [22], in which both ice layers are simultaneously tracked by evolving an initial contour in the radar imagery until the ice interfaces are properly represented by the contour boundaries. This method was developed for tracking of two-dimensional echograms, and requires no additional sources of evidence regarding the ice sheets. Ilisei *et al*. [23] propose a technique that exploits the statistical properties of the radar response and generates a subsurface map that is segmented into regions corresponding to various features. This solution was applied to 2D echograms acquired from MCoRDS.

In another solution, Crandall *et al*. [14] pose layer-tracking as an inference problem on a probabilistic graphical model (PGM). An HMM is formulated from the constraints and assumptions made about the data, and divided into a set of non-loopy graphs on which the Viterbi algorithm can be utilized to solve for the highest-probability sequence of hidden states. The hidden states, in this case, are the rows that correctly label the ice-bottom layer in each column of the

input image matrix. In this document we propose improvements to this solution, in the form of novel cost functions that allow for further domain-specific knowledge and additional sources of evidence to be integrated into the PGM.

The original Viterbi algorithm-based tracking technique was designed to track both ice layers in 2D imagery. The ice-surface was detected first, and then the ice-bottom layer with the restriction that it should be located underneath the ice-surface. This method has no additional input requirements other than the echogram itself and a set of optional scalar parameters, which are factored in the calculation of the pixel and transition probabilities that compose the model. In this work, an additional image processing step based on the image gradient was used before tracking.

Building on the probabilistic graphical approach proposed by [14], Lee *et al*. [24] construct an MRF to represent the model and use the Gibbs sampling algorithm to sample from the joint distribution of all possible layers conditioned on a 2D radar echogram image. The joint distribution of the layer boundaries is decomposed according to how well the echogram of interest can be explained by a given labeling, and how well the labeling satisfies the assumptions made about the ice layers, such that they should be smooth, continuous, and located around areas of high pixel intensity. The Gibbs Sampler is a Markov Chain Monte Carlo (MCMC) algorithm capable of producing samples from a given target distribution without directly sampling from it or even knowing its form, as long as the conditional distribution of each of the variables is known and tractable to compute. The Gibbs sampling-based approach has no additional input requirements, and detects both ice layers simultaneously by solving for the joint probability of the two layers.

Research has also been conducted with the purpose of developing automated layer tracking software able to detect multiple internal layers from a two-dimensional radar image, rather than only tracking the ice-bottom layer or both the ice-surface and ice-bottom boundaries. Examples

include the solution proposed by Mitchell *et al.* [25] which uses an active contour model to find high-intensity edges likely to correspond to layer boundaries. Mitchell [26] also demonstrates the application of MCMC techniques to extract internal ice layers from nadir-looking 2D echograms. Panton [27] proposes an active contour solution which requires a human operator to manually provide seed points as initial layer estimates. Carrer and Bruzzone [28] use a local scale HMM solved with the Viterbi algorithm to detect internal layers after a series of image pre-processing steps.

Research efforts in the fields of remote sensing, computer vision, and pattern recognition have also been performed with the purpose of identifying roads, vegetation, buried objects, and other features from ground-penetrating radar imagery, which is similar to our problem. As an example, a technique using a PGM solved using the Viterbi algorithm is proposed by Smock and Wilson [29].

For ice-bottom extraction from three-dimensional imagery, a solution was proposed by Xu *et al*. [17] using the TRW-S algorithm to iteratively perform energy minimization on a first-order MRF in a PGM framework similar to that proposed by Crandall *et al*. [14]. In this document we propose improvements to this solution, in the form of novel cost functions that allow for further domain-specific knowledge and additional sources of evidences to be integrated into the PGM. Berger *et al.* [30] presents improvements to the existing HMM/Viterbi algorithm framework of [14] and MRF/TRW-S algorithm framework of [17], many of which are reviewed in this document.

Furthermore, Xu *et al.* [31] recently proposed a deep-learning approach using a combination of a three-dimensional convolutional neural network (C3D) and a recurrent neural network (RNN) for three-dimensional imagery. Convolutional and recurrent neural networks have

revolutionized computer vision and pattern recognition tasks in recent years, and this is the first application of neural networks for the purpose of three-dimensional ice-layer tracking that we are aware of. In this solution, the C3D network performs local feature extraction in both intra- and inter-slice dimensions, while the RNN is used to capture longer-range features both within individual images and across the entire sequence of slices that compose the 3D dataset.

1.5 - Outline of thesis

The remainder of this document is organized as follows: in Chapter 2, a theoretical review of the HMM and MRF frameworks and the Viterbi and TRW-S algorithms is provided, along with an explanation regarding their implementation for ice layer tracking purposes. Chapter 3 presents an empirical method of providing the automated layer tracking software with additional information about the ice layers based on a geostatistical analysis of ground truth data. Chapter 4 discusses the results obtained from the proposed algorithms, and presents a comparison between these and the results generated by other proposed layer tracking solutions. Chapter 5 presents concluding remarks about the work performed.

Chapter 2 - THE HMM AND MRF GRAPHICAL MODELS FOR ICE LAYER TRACKING

2.1 - Introduction

In this section, we provide an overview of the automated ice-bottom layer tracking methods based on the construction of an HMM and a MRF for two-dimensional and three-dimensional imagery, respectively. A theoretical review of the Viterbi and TRW-S algorithms is first provided, followed by an explanation of the HMM and MRF graphical models and their implementation for ice layer tracking purposes. Furthermore, a summary of the inputs to and outputs from both HMM and MRF trackers is provided, along with a brief characterization of each.

Several improvements to both layer tracking applications are also proposed, mostly in the form of improved image pre-processing steps and novel cost functions that integrate further domain-specific knowledge into the probabilistic graphical models constructed. These modified cost functions were designed to match empirically-discovered properties of the surveyed ice based on a geostatistical analysis of manually-corrected ground-truth data, in a procedure described in detail in Chapter 3 of this document. We also propose a modified approach regarding the implementation of the TRW-S algorithm which helps improve tracking results compared to the original solution of [17].

In the case of individual two-dimensional images, we construct an HMM conditioned on the input image, as proposed by Crandall *et al*. [14]. The most likely sequence of hidden states of an HMM can be found efficiently and exactly by performing inference with the Viterbi algorithm. Here, the definition of "two-dimensional images" is applied to both nadir-looking 2D echograms and individual slices of 3D imagery. On the other hand, an MRF is constructed conditioned on the entire three-dimensional image (in other words, on the entire set of consecutive slices that compose

a data frame). Performing exact inference on an MRF is an NP-hard problem and thus computationally intractable in the general case. However, efficient approximate techniques for this purpose exist, including the TRW-S algorithm used for this purpose by Xu *et al.* [17]. TRW-S is the *de facto* state-of-the-art in approximate MRF inference. Regardless of the model or algorithm used, the result of the inference step is the vector or matrix that describes the path with the highest calculated probability of correctly labeling the ice-bottom boundary conditioned on the input image and ancillary data used by the specific implementation.

Both HMM and MRF models are based on a unary cost function which assigns the cost for the ice-bottom layer to pass through each and every valid pixel. The formulation of this unary cost function is similar in both the HMM and MRF models. These models are also based on a binary (or *pairwise*) cost function, which is defined differently between the HMM and MRF models and generates a major difference between the two. In the case of a two-dimensional image, the binary cost function assigns costs to all valid transitions between adjacent columns, regardless of whether the columns represent range-lines (if the image is a nadir-looking 2D echogram), or represent direction-of-arrival bins (if the image is a single slice from 3D imagery). This combination of a unary cost function and a one-dimensional binary cost function defines an HMM.

However, in the case of three-dimensional imagery, two binary cost functions are used: one that enforces smoothness between adjacent columns (direction-of-arrival bins), and another that enforces smoothness on adjacent slices (range-lines). This combination of a unary cost function and a two-dimensional binary cost function generates a MRF, which, unlike an HMM, cannot be solved using the Viterbi algorithm. An MRF is similar to an HMM in that it consists of unary and pairwise terms, but an important difference is that the graphical structure of pairwise terms in an HMM forms a chain whereas an MRF may have arbitrary graph structure.

In both HMM and MRF models, each valid pixel is a node of the corresponding graph. In Fig. 2.1, we present a diagram depicting a two-dimensional graph of size 4-by-2, as well as the unary cost and one-dimensional binary cost assignments which define an HMM. In this diagram, each node is represented as a square and is assigned an individual unary cost value. The binary cost assignments are represented as arrows, and only the assignments corresponding to one specific node are shown in the figure for ease of visualization. In reality, binary costs are assigned between all valid pixels of all pairs of consecutive columns of the graph. Since these binary costs are only assigned to transitions between adjacent columns, this is a one-dimensional binary cost function. However, in the MRF there exists a third dimension corresponding to different slices of the 3D imagery, which can be visualized as an extension of the graph depicted below to include multiple *pages*. Therefore, the binary costs in the MRF are assigned both column-to-column (cross-track) and page-to-page (along-track), and thus this is a two-dimensional binary cost function.

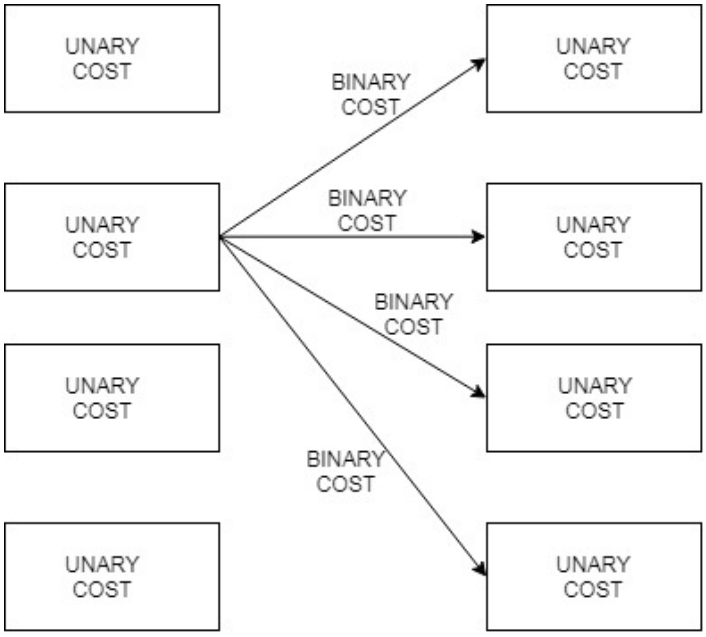

Fig. 2.1. Diagram depicting a two-dimensional graph, in which the definition of a unary cost for each individual node and a one-dimensional transition (binary) cost function between adjacent nodes in consecutive columns defines an HMM. Not shown are the binary cost assignments between every pixel of the first column and every pixel of the second column.

The HMM-based tracker was originally developed for the purpose of layer tracking on nadir-looking 2D echograms only; however, it can be applied to 3D data with no modifications to its implementation if every slice of the three-dimensional imagery is individually processed by the software – after all, each individual slice is a two-dimensional image. As the Viterbi algorithm operates on non-loopy chains only, there seems to be no possibility of applying it on the entire 3D dataset at once, but inter-slice message passing may be simulated by providing the algorithm with the results from the previously tracked slice in the form of additional ground-truth data points. This possibility is not explored in this work and each slice is tracked individually with no prior information from other slices. Note that such an approach would still be significantly less robust than TRW-S, because it would make final decisions in a sequential manner since it would consider slices sequentially instead of simultaneously. For example, an incorrect decision in the first slice would negatively impact inference on subsequent slices because that bad estimate would be fed in as ground truth. In contrast, TRW-S considers all evidence simultaneously, meaning that ambiguity in the first slice may be resolved by evidence in later slices.

Without the capability of inherently propagating layer information between adjacent slices, the Viterbi tracker often presents discontinuities in the along-track dimension when applied to three-dimensional datasets – this is only partly mitigated, but not completely solved, if ground-truth from the previous slice is provided to the tracker. On the other hand, the TRW-S algorithm is well-suited for 3D data tracking due to its capabilities of performing both intra-slice (cross-track) and inter-slice (along-track) message passing when performing global energy minimization. The inter-slice message propagation is performed both forwards and backwards, which allows for the tracking result of a given slice to be iteratively influenced by the partial tracking of slices both

ahead of and before it. This method is therefore capable of preventing discontinuities in both along-track and cross-track dimensions.

2.2 - The Viterbi algorithm

The Viterbi algorithm is an efficient dynamic programming method for finding the highest-probability sequence of hidden states in a finite-state discrete-time HMM. The path which presents the highest-probability sequence of hidden states is the path with the highest total *path metric* between the initial and final states [32]. In the context of a Markov process, any non-zero probability of a single transition between two states is defined as a *branch metric*. The *branch metric* of the transition from state i to state j is defined as m_{ij}. This and other probabilities are usually computed as log-likelihood probabilities to reduce computational complexity and rounding errors by converting multiplications of probabilities to summations of log-probabilities. The Viterbi algorithm finds the path which maximizes the *path metric*, which equals the sum of all *branch metrics* between the initial and final states. Mathematically, the procedure applied is defined as the recursive optimization problem of

$$M_j(k) = MAX_i \{ M_i(k-1) + m_{ij}(k) \}, \quad (1)$$

which states that the maximum-likelihood path up to state j after k transitions equals the maximum-likelihood path up to the previous state i, plus the *branch metric* between states i and j. Any unallowed transitions between two states are given the value of $m_{ij} = -\infty$, to ensure that they will not be selected as part of the final path after maximization. The final path discovered by the algorithm is guaranteed to be the one which maximizes the global likelihood of transitions between the initial and final states.

2.3 - The TRW-S algorithm

The TRW-S algorithm is a successor to the tree-reweighted max product message passing (TRW) algorithm [33] for computing the maximum a posteriori (MAP) configuration of an MRF in which the cost function may be defined as a cyclic (or *loopy*) graph-structured probability distribution. For non-cyclic graphs (also known as *trees*), simpler message-passing methods such as the max-sum (or max-product) algorithm may be applied instead. The Viterbi algorithm is a special case of the max-sum (or max-product) algorithm.

Both TRW and TRW-S are iterative message-passing algorithms in which real-valued functions called *messages* are exchanged among the nodes of the MRF. The messages contain likelihood distributions that reflect each variable's beliefs about the labels of its neighbors, based on its own evidence and the messages it has received in past iterations. In order to propagate the messages, both algorithms break down the MRF into several monotonic chains and transmit the messages along each chain. Local evidence thus iteratively propagates from neighbor to neighbor, spreading across the entire graph given enough iterations. After all iterations are performed, each node chooses its label based on its local evidence and the last set of messages it received from its neighbors. The main difference between the two algorithms is that the original TRW performs this message propagation on a bi-directional parallel schedule, whereas TRW-S does so in a specific sequential schedule. Because of this modification, TRW-S requires only one message, rather than two, to be passed on each edge of the graph – thus, it converges faster and requires only half the memory of TRW and other traditional message passing methods. However, neither of these algorithms are guaranteed to find the optimal cost minimization sequence within the graph. In the context of the CReSIS three-dimensional data, each pixel in the image passes a cost message to its

four neighboring pixels to the left and right (direction-of-arrival dimension) and forward and backward (along-track dimension).

TRW and TRW-S may also be applied on a non-cyclic graph. In this case, the exact marginal will be calculated and the algorithm is guaranteed to converge to the optimal MAP configuration. At this point, these algorithms would be similar, both in execution and performance, to the Viterbi algorithm. Since the Viterbi algorithm was already initially implemented for performing inference on non-cyclical graphs in the context of ice layer tracking and is guaranteed to efficiently find the optimal solution, the possibility of using TRW-S or other loopy-BP techniques for this purpose has not been explored.

While both Viterbi and TRW-S algorithms were originally devised to discover the maximum-likelihood configuration conditioned on the HMM or MRF of interest, the exact same frameworks may be used to perform inference on a graph in which edges are described in terms of a discrete *cost*, rather than probabilities. The cost represents a penalty associated with selecting a certain sequence of edges, rather than an increased likelihood, and thus needs to be minimized rather than maximized. In several publications, this cost is referred to as *energy*, and thus the procedure receives the name of *energy minimization.* The Viterbi and TRW-S are tasked with discovering the lowest-cost solution conditioned on the given radar echograms. Costs are assigned in a very similar way for both 2D and 3D imagery, according to a cost model explained in detail in Section 2.4.

The original TRW-S ice-bottom tracker proposed in [17] has been modified in several ways to improve its performance [11, 30]. In the original implementation, in both direction-of-arrival and along-track dimensions there was a preferential direction where the current iteration message is propagated, while a stale cost message from the previous iteration was propagated in the opposite

direction. This created a bias towards the side of the image from which the preferential direction started from. For example, when the preferential direction is left to right, the left-most side of the image has a stronger effect on the result than all other columns because its message is passed all the way across the image in a single iteration. The original solution dealt with this by alternatively propagating messages from the left to right and then right to left and from up to down and then down to up. The issue with this solution was that the most extreme directions of arrival (far-left and far-right), where the signal quality is worst, were being given too much influence.

In the case of 3D imagery, we have high-confidence ground-truth points at the nadir direction-of-arrival bin of every slice which is the *center* of the image. The ground-truth points are from the corresponding previously tracked and corrected two-dimensional data. Therefore, the ground-truth point at nadir should exert the largest influence on the tracking, and for this reason the preferential direction was changed to be outward from nadir. So on the left side of the image, the preferential direction is toward the left and on the right side of the image, the preferential direction is toward the right. In this way, the nadir column asserts the greatest influence. See Fig. 2.2 for a graphical representation of the message propagation in the along-track and cross-track dimensions performed by this new implementation of the TRW-S algorithm.

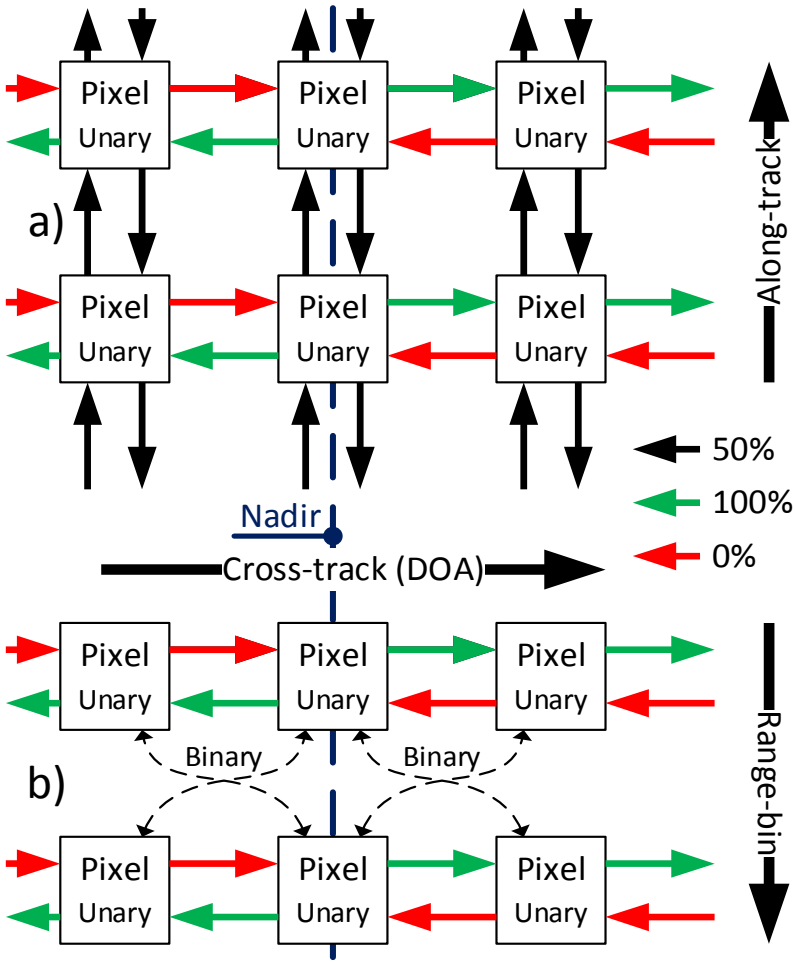

Fig. 2.2. The diagram above depicts the unary cost assignment to individual pixels (represented by squares) and two-dimensional binary cost propagation in the along-track and cross-track dimensions in part (a) and in the cross-track dimension in part (b). Note from (a) that binary cost is entirely propagated outwards from nadir with the proposed modification, and no cost is propagated from the extreme elevation angles to nadir. The along-track (slice-to-slice) propagation is performed evenly (i.e. 50% weight is assigned to the incoming and outgoing messages both forwards and backwards).

2.4 - A probabilistic graphical model for ice layer tracking: unary cost function

The aforementioned HMM and MRF cost-minimization graphical models are formulated based on two cost functions: one unary, which assigns unary costs to individual pixels of the given radar image, and one binary, which assigns costs to all consecutive range-line-to-range-line transitions (in both 2D and 3D cases) and to all consecutive DoA-bin-to-DoA-bin transitions (in

the 3D case only). In both 2D and 3D labeling problems, the goal is to trace the location of the ice bed; in other words, the goal is to find the set of labels that correctly delineate the ice-bottom interface.

In the case of nadir-looking two-dimensional imagery, the HMM-based automated tracker is given an intensity image I of dimension N_{RB}-by-N_{RL}, where N_{RB} corresponds to the number of range-bins and N_{RL} corresponds to the number of range-lines that compose the echogram. Let us also define $s_{SURF}(c)$ as the range-bin index of the known ice-surface layer at column c of the image, therefore with the restriction that $1 \leq c \leq N_{RL}$ and $1 \leq s_{SURF}(c) \leq N_{RB}$. Concretely, the goal is to identify, for each column c of I, the corresponding row coordinate s_c through which the ice-bottom layer passes. The tracking output $S = (s_1, s_2, \dots, s_{N_{RL}})$ is therefore in the form of a N_{RL}-dimensional vector with the restriction that $s_{SURF}(c) \leq s_c \leq N_{RB}$, and thus s_c may be found in any of the valid range-bins (i.e. at or below the given ice-surface location). The format of the tracking output vector assures that exactly one row (range-bin) will be selected as the ice-bottom layer for each column (range-line) of the image. As previously mentioned, the HMM-based tracker is also able to perform tracking on individual slices of three-dimensional imagery. Each slice is an intensity image I of dimension N_{RB}-by-N_{DoA}, where N_{DoA} is the number of direction-of-arrival bins in the input slice. In this case, the output obtained from the HMM tracker is in the form of a N_{DoA}-dimensional vector. In the three-dimensional case, the MRF-based automated tracker is given a three-dimensional matrix of dimension N_{RB}-by-N_{DoA}-by-N_{RL}, which therefore corresponds to a sequence of N_{RL} slices, each with dimension N_{RB}-by-N_{DoA}. The tracking output in this case is a N_{DoA}-by-N_{RL} matrix where each element represents the output found for each column of each slice that composed the three-dimensional input space.

In both 2D and 3D radar imagery cases, the unary cost function is tasked with integrating into the probabilistic model several constraints and assumptions regarding the shape and location of the ice-bottom. It assigns to every *valid* pixel a unary cost ψ_U which represents the cost for the ice-bottom layer to pass through that pixel. A pixel of any column is considered to be *invalid* for the ice-bottom layer if, for example, it is located above the given ice-surface layer for that column.

Furthermore, our proposed ice-bottom trackers make use of existing ice-mask datasets to invalidate additional pixels and thus further *sparsify* the calculation of unary costs. An ice-mask dataset is a binary raster used to determine whether or not ice is present at a given geographical location. These data are available for most regions surveyed by CReSIS (e.g. Randolph Glacier Inventory [34]). In both the HMM and MRF models proposed here, every pixel in the same range-bin as the ice-surface layer is considered invalid if the ice-mask points to the existence of ice. This is sensible since the layer tracker should find a non-zero ice thickness for every column in which ice exists. For all pixels in columns to which the ice-mask points to the inexistence of ice, however, the unary cost calculation is skipped and the ice-bottom location is immediately set to match the location of the ice-surface. All invalid pixels are assigned a unary cost of $\psi_U = \infty$, which guarantees they will not be selected by the algorithms as part of the path with minimum total cost.

Other than the aforementioned invalidation of certain pixels, the leading assumption made in our proposed framework is that the ice-bottom boundary is located along strong radar reflection peaks since the dielectric contrast between the ice and bedrock or water below creates a strong reflection. For this reason, the first unary term – and the one with the greatest overall impact on unary cost – increases the unary cost of pixels with low intensity. This term is ψ_{SINC}, named in this manner due to the sinc function used in its calculation.

Previous work [17] measured the sum squared difference in the image pixel intensity relative to a template of an ideal layer return. The template was found through an automated training sequence using the *a priori* surface information. Although it is data dependent, the template invariably has a peak in the center with decreasing values towards the edges of the template. Because the term measured the squared distance to the template, a peak response in the imagery with exactly the same intensity values produces the lowest cost. We modified the previously proposed template term in order to better use the dynamic range of the imagery. A problem with using the squared distance to the template is that the peak intensity from the ice-bottom layer varies and larger intensities generally indicate a better measurement (since these correspond to greater signal to noise ratios). The square distance metric meant that values with a larger peak intensity than the template would actually be penalized. To better handle peak intensity variability, we now use a correlation operation that multiplies the template with the image:

$$\psi_{\mathrm{SINC}}(\mathrm{s},\mathrm{c}) = -\sum_{\mathrm{p}\in\mathrm{T}} \mathrm{I}(\mathrm{s}+\mathrm{p},\mathrm{c})\mu(\mathrm{p}), \tag{2}$$

where I is the input image and $\mathrm{p} \in \mathrm{T} = \{-5,-4,-3,\ldots,5\}$ refers to the pixel index of the correlation function, and $\mu(\mathrm{p})$ is the correlation function which is now fixed to $\mathrm{sinc}\,(\mathrm{p}/3.33)$ which for ± 5 pixels approximately corresponds to the midpoint in the first minimum on either side of the sinc function peak at $\mathrm{p} = 0$ as shown in Fig. 2.3. The truncated sinc waveform was chosen because it is symmetric and has a single peak in the center. Improved performance is likely possible by tuning the waveform shape.

To allow the algorithms to account for the inclusion of ground-truth data points, two cost terms are proposed: ψ_{GT} and ψ_{EXTRA}. In the case of nadir-looking 2D echograms, ground-truth points are 2-tuples which associate a certain range-bin with a certain range-line and thus have a format of $(\mathrm{s}_{\mathrm{GT}}, \mathrm{c}_{\mathrm{GT}})$ where s_{GT} is the range-bin index and c_{GT} is the range-line index of the ground-

truth point. In the case of three-dimensional imagery, ground-truth points are 3-tuples which associate a certain range-bin with a certain slice and a certain DoA bin of the 3D dataset, and thus have the format of (s_{GT}, c_{GT}, d_{GT}) where d_{GT} is the index of the slice with which the ground-truth point is associated.

The purpose of the ψ_{GT} and ψ_{EXTRA} unary cost terms is to encourage the ice-bottom layer to be drawn towards ground-truth points, if they exist. Both terms modify the unary cost of all valid pixels located in any column (range-line in 2D imagery, DoA bin in a slice of 3D imagery) which contains a ground-truth point. Pixels in columns with no ground-truth points are unaffected. The term ψ_{GT} accounts only for high-confidence ground-truth points, whereas ψ_{EXTRA} accounts only for low-confidence ground-truth points. To account for potential small inaccuracies in the ground-truth, the algorithms are not forced to return an answer in which the ice-bottom is found to exactly match the location of that point, but encouraged to approximately match it by a cost term

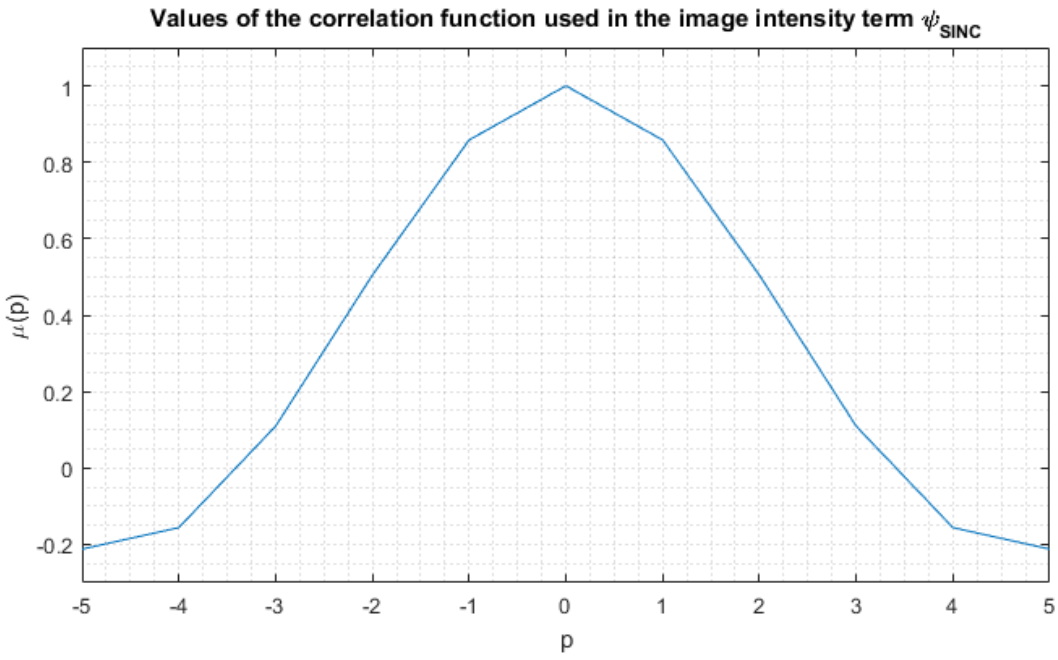

Fig. 2.3. Values of the correlation function $\mu(p)$ used in the image intensity term ψ_{SINC}.

that is positively associated with the squared vertical distance to the ground-truth point, as in

$$\psi_{GT}(s, c) = (s - s_{HCGT}(c))^2 \text{ and } \psi_{EXTRA}(s, c) = (s - s_{LCGT}(c))^2 \quad (3)$$

where s represents the row index and c represents the column index of the pixel of interest. s_{HCGT} and s_{LCGT} represent the row (range-bin) index of high-confidence and low-confidence ground-truth points, respectively. Effectively, the difference between the influence of ψ_{GT} and ψ_{EXTRA} in the total unary cost arises from the fact that they are given different weighting in the final unary cost calculation, with ψ_{GT} being assigned significantly higher weight. Fig. 2.4 demonstrates the cost, before any weighting is applied, added by the two ground-truth terms.

High-confidence ground-truth can be manually added by a human operator. For both 2D and 3D imagery, this is not done before the automated tracking is run. However, high-confidence ground-truth points are automatically acquired for 2D data by intersecting the flight path of the

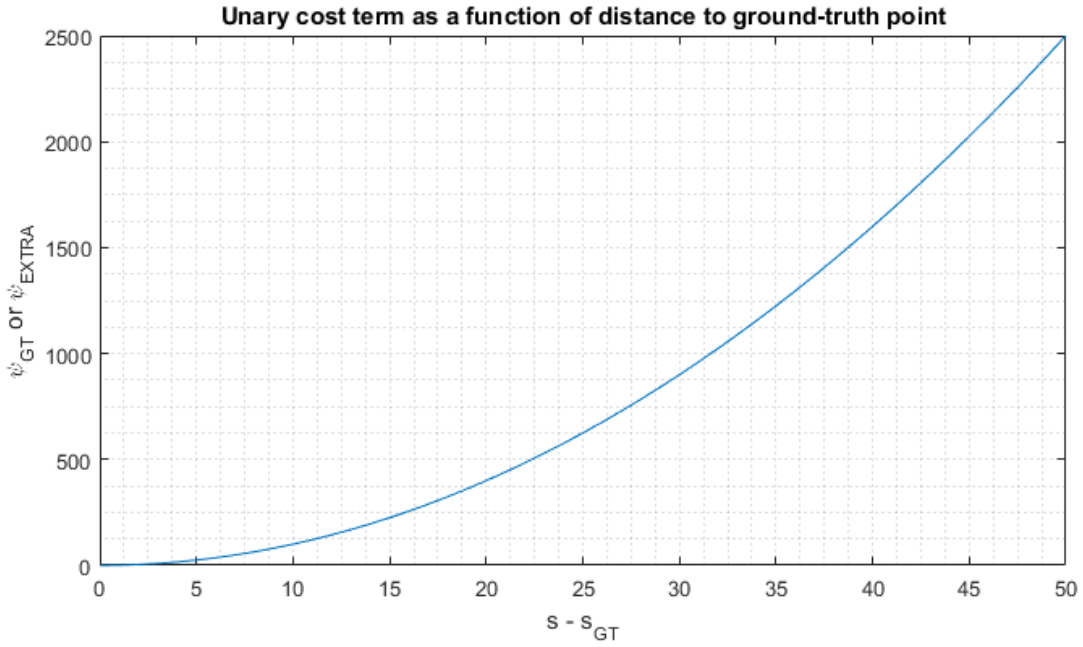

Fig. 2.4. The ψ_{GT} and ψ_{EXTRA} terms of the unary cost function of both Viterbi and TRW-S algorithms increase unary pixel cost according to the squared vertical distance between the pixel of interest and the ground-truth point, as demonstrated by Eq. (3). The ψ_{GT} term accounts for high-confidence ground-truth points and is given higher weight in the final unary cost calculation.

data frame of interest with flight paths of previous surveys of that geographical region. Frequently, a given location will have been imaged and labeled before, and the point at which the new flightline crosses the old will already have ice-bottom depth information associated with it, which can then be used to help the tracker. These flightline intersections are commonly known as *crossovers*, and can also be used as part of a self-assessment of the error associated with layer tracking results as demonstrated in Chapter 4 of this document.

For 3D imagery, high-confidence ground-truth points are taken from the result of the 2D tracking process by using the ice-bottom label as ground-truth in the nadir elevation angle bin of each slice of the 3D imagery. Although this is not strictly required for constructing the MRF model or applying the TRW-S algorithm, in all the results presented in this work the tracked nadir bin from 2D imagery is used as ground-truth to the 3D imagery and we did not evaluate the performance without this ground-truth added in for 3D imagery.

A potential source of low-confidence ground-truth points is existing third-party estimates of the ice-bottom obtained, for example, from ice thickness models based on ice flow dynamics and mass conservation [6]. While these methods provide wide-coverage estimates of the ice-bottom, they offer relatively low-resolution or smoothed estimates of the ice-bottom. Although somewhat imprecise, we may take advantage of these ice-flow or mass-conservation estimates as a weak source of evidence since they rarely diverge very far from the true result. See Fig. 2.5 for an example of a nadir-looking 2D echogram including an estimate of the ice-bottom obtained through the mass conservation method. Since these *a priori* estimates are interpolated to cover all range-lines, every range-line receives a low-confidence ground-truth point in this case.

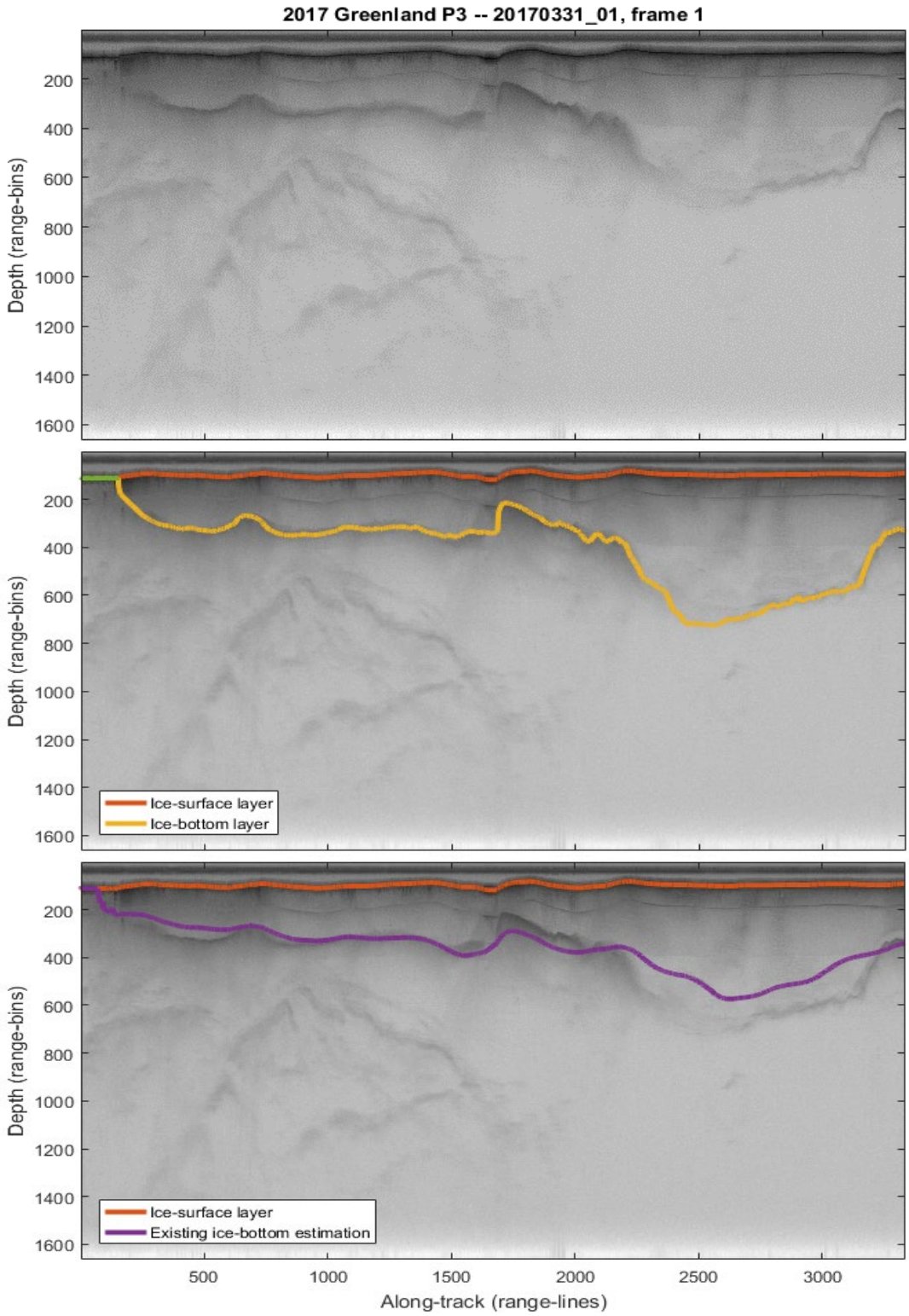

Fig. 2.5. A two-dimensional echogram from the 2017 OIB deployment to Greenland. From top to bottom: the original echogram with no layers depicted; the same echogram with the existing ice-surface layer (in orange) and the true ice-bottom layer (in yellow) overlaid; the same echogram with the existing ice-surface layer (in orange) and the existing a priori ice-bottom estimate (in magenta) obtained from the mass conservation method, which is used in the HMM model as a weak source of evidence of the ice-bottom.

In addition, it is beneficial to add some sort of repulsion factor to the unary cost in order to prevent the ice-bottom trackers from mistakenly labeling part of the strong ice-surface return as the ice-bottom. The original HMM-based tracker [14] did not include a term with this function into the unary cost calculation, but a fixed unary cost term ψ_{REP} was used for this purpose in the original MRF/TRW-S framework [17]. Let us define Δ_{Y} to be the positive vertical distance, in range-bins, between the known ice-surface layer and the pixel of interest:

$$\Delta_{\mathrm{Y}}(\mathrm{s},\mathrm{c}) = \mathrm{s} - \mathrm{s}_{\mathrm{SURF}}(\mathrm{c}) \tag{4}$$

where s represents the row index of the pixel of interest, and $\mathrm{s}_{\mathrm{SURF}}(\mathrm{c})$ is the range-bin index of the ice-surface layer at the same column c as the pixel of interest. The unary cost term ψ_{REP} proposed in [17] is defined as

$$\begin{gathered}\text{if } \Delta_{\mathrm{Y}}(\mathrm{s},\mathrm{c}) < 20\text{:} \\ \psi_{\mathrm{REP}}(\mathrm{s},\mathrm{c}) \coloneqq 200\end{gathered} \tag{5}$$

which therefore assigns an equal repulsion cost to all pixels within 19 or fewer range-bins of the ice-surface layer. However, this cost formulation tends to generate results in which a disproportionally large percentage of the ice-bottom layer is located exactly 20 range-bins away from the ice-surface, due to the abrupt increase in unary costs for all pixels between that range-bin and the ice-surface layer.

To combat this, we modified ψ_{REP} to introduce a gradual increase in the repulsion cost term, in which a pixel immediately under the ice-surface ($\Delta_{\mathrm{Y}} = 1$) is assigned a ψ_{REP} value approximately equal to 180, whereas a pixel with 50 or more range-bins of vertical separation to the ice-surface would have no unary cost added from this term. A smooth transition between these two extreme values was found, calculated according to a fixed shifted exponential decay function for pixels with $1 \leq \Delta_{\mathrm{Y}}(\mathrm{s},\mathrm{c}) \leq 50$. Note that a value of $\Delta_{\mathrm{Y}} = 0$, which would correspond to a pixel on the same range-bin as the ice-surface, is meaningless due to the aforementioned invalidation of

such pixels. This proposed fixed repulsion term ψ_{REP} is mathematically defined as

$$\psi_{REP}(s,c) = \begin{cases} 0, & \Delta_Y > \alpha_{MSD} \\ \alpha_{MC} * e^{-\lambda * \Delta_Y} - \alpha_{MC} * e^{-\lambda * \alpha_{MSD}}, & \text{otherwise} \end{cases} \quad (6)$$

where α_{MSD} defines the maximum sensory distance from the ice-surface, α_{MC} defines the maximum cost applied by this term, and λ is a manually chosen exponential decay constant. In the previously proposed solution that made use of this fixed repulsion term, the selected values for these parameters were $\alpha_{MSD} = 50$, $\alpha_{MC} = 200$, and $\lambda = 0.075$. The value of α_{MC}, which in theory would be assigned to pixels in the same range-bin as the ice-surface layer ($\Delta_Y = 0$), was specifically selected to match the original fixed repulsion cost value of the MRF framework, presented in Eq. 5. Fig. 2.6 illustrates this proposed term.

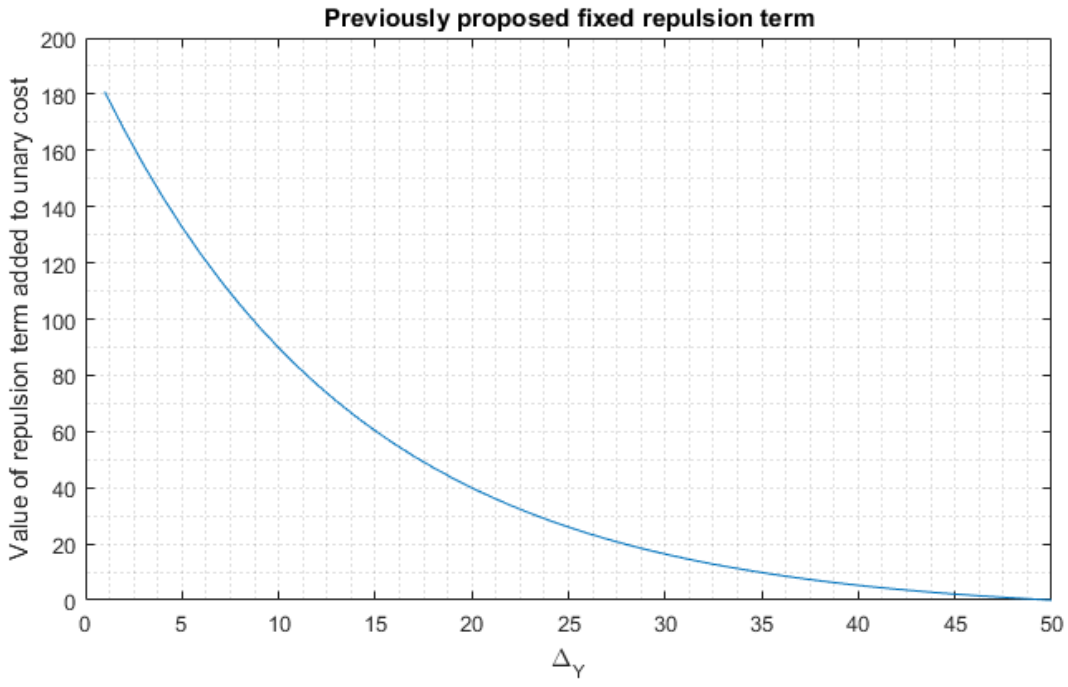

Fig. 2.6. The previously proposed fixed repulsion term of the unary cost function, which increased the cost of all valid pixels according to how close they were to the ice-surface. Note that this was formulated as an exponential decay function capped at a value of approximately 180 for pixels immediately underneath the ice-surface ($\Delta_Y = 1$). As an improvement to this term, we propose an empirically-derived term based on a geostatistical analysis of the data, which substitutes the function demonstrated above for a dynamically-calculated term.

Furthermore, it is helpful to add to the PGM a unary cost term ψ_{SURF} with the intention of enforcing a surface and ice thickness constraint on all valid pixels. The original versions [14, 17] of the HMM and MRF frameworks did not employ a cost term with this purpose in their unary cost functions. In previous versions of the work presented here, this cost term was set to force the ice-bottom to lie in a certain range relative to the ice-surface if close to the ice-margin (i.e. near the transition between no-ice and ice), or have no effect ($\psi_{\mathrm{SURF}} = 0$) when more than a certain distance away from the ice-margin. In this previously proposed model, the difference between the ψ_{REP} and ψ_{SURF} terms lies in the fact that ψ_{REP} will gradually increase the unary cost of pixels as they vertically approach the given ice-surface layer regardless of the ice-mask value of the column of interest, whereas ψ_{SURF} is a hard constraint that is set to either ∞ or zero, and effectively restricts the range of allowed values for s. The term "ice-margin" refers to the meeting point between icy and non-icy regions.

However, both the aforementioned ψ_{REP} and ψ_{SURF} methods present major drawbacks. Defining the ψ_{REP} term as presented above leads to every pixel – regardless, for example, of whether the surveyed terrain presents sea ice/icebergs or a thick continental ice sheet – being equally penalized for generating a thin ice layer. Furthermore, the aforementioned ψ_{SURF} term, while useful in generating smoother transitions between icy and non-icy areas, was in part arbitrarily defined and not specifically tuned to match the realistic constraints of the ice sheets.

We propose a dynamic, empirically-discovered term able to simultaneously replace both ψ_{REP} and ψ_{SURF} terms of the unary cost function described above. The main purpose of this improvement is to assign a single cost term ψ_{DIM} which accounts for the expected ice thickness of a point according to the distance between that point and the ice-margin nearest to it. In other words, this novel cost term assigns a single unary cost term as a function of both the thickness of the ice

and the distance to the ice-margin, effectively combining the ψ_{REP} and ψ_{SURF} terms into one. We henceforth refer to the distance between a given coordinate and the ice-margin nearest to it as "distance to ice-margin", or DIM. Furthermore, this novel cost term was empirically discovered rather than being arbitrarily defined or tuned, which likely increases the overall robustness and favorable impact of this term. The method through which this novel cost term is calculated is detailed in Chapter 3 of this document.

Intuitively, areas very near the ice-margin – such as sea ice, icebergs, or the edges of the ice sheets – tend to present thinner ice, whereas points with large DIM may present a variety of ice thickness measurements but it is unlikely that these points will contain very thin ice.

After the four unary cost terms (including ψ_{DIM}, presented in detail and mathematically defined in Chapter 3) are calculated for every valid pixel, the unary cost function performs a weighted summation to assign the final unary cost, as follows:

$$\psi_{\mathrm{U}}(\mathrm{s},\mathrm{c}) = \mathrm{w}_{\mathrm{SINC}}\psi_{\mathrm{SINC}}(\mathrm{s},\mathrm{c}) + \mathrm{w}_{\mathrm{GT}}\psi_{\mathrm{GT}}(\mathrm{s},\mathrm{c}) + \mathrm{w}_{\mathrm{EXTRA}}\psi_{\mathrm{EXTRA}}(\mathrm{s},\mathrm{c}) + \mathrm{w}_{\mathrm{DIM}}\psi_{\mathrm{DIM}}(\mathrm{s},\mathrm{c}) \tag{7}$$

where the w variables control the weights of each term in the total calculation. These weights may be manually or automatically tuned to improve the tracking performance of the algorithms. A potential method of tuning these scalars in an automated manner, as long as manually-corrected ground-truth layer data are available, is to perform a naïve (exhaustive) grid-search hyper-parameter optimization routine in which the imagery is tracked using several different combinations of these parameters and the combination that yields the lowest average error when compared to the ground-truth layers is selected as optimal. A more robust approach is to perform random search [35], a recent hyper-parameter global optimization technique that has been shown to outperform naïve grid-search methods, particularly in large parameter spaces where not all

variables have equal impact on the final error measurement and therefore are not equally important to tune. Regardless of the optimization method, it is possible that the optimal combination found based on the labeled dataset of a given deployment season may no longer be optimal or perform as well when applied to a different season, due to differences in image characteristics or in the ice itself (e.g. tuning on a mountainous regions and then applying the results to a flat region).

2.5 - A probabilistic graphical model for ice layer tracking: binary cost function

In both of the proposed HMM and MRF frameworks, a binary cost ψ_B is assigned to every transition between consecutive range-lines (in both 2D and 3D imagery) and to every transition between consecutive direction-of-arrival bins (in 3D imagery). In the case of 2D echogram tracking, handled only by the HMM/Viterbi algorithm in the presented framework, the binary function defines an along-track transition model by assigning costs to all transitions between consecutive columns (range-lines) of the image. When the HMM/Viterbi algorithm framework is applied to individual slices of 3D imagery, this pairwise function defines a cross-track transition model by assigning costs to all transitions between consecutive columns (DoA bins) of each slice. Due to the single-chain (non-cyclic) restrictions on the graph structure in the HMM framework, it is not possible to also assign an along-track transition model in this case.

In the MRF/TRW-S algorithm framework, applied only to three-dimensional spaces, the binary cost function defines costs for transitions in both the along-track and cross-track dimensions – this is possible because an MRF may have arbitrary graph structure. Note that the along-track transition model assigns costs to consecutive slice-to-slice transitions of the MRF whereas the cross-track transition model assigns costs to consecutive DoA-bin-to-DoA-bin transitions of the MRF; this is not to be confused with the message-passing system employed by the TRW-S

algorithm which is simply a method of iteratively solving the MRF defined by the unary and binary costs.

The main purpose of the binary cost is to enforce a smoothness constraint on the model, increasing the likelihood that transitions which generate smoother layers will be selected by assigning to these a lower cost. While the ice-bottom boundaries are often found to be rough and sloped, abrupt vertical discontinuities in the ice are not expected. Thus, a locally-smooth interface, in which no large *jumps* occur between any two consecutive range-lines or direction-of-arrival bins, is generally a reasonable assumption for the bottom of the ice sheet. However, more sophisticated transition models – such as discussed in this section and in Chapter 3 – may employ empirical probabilistic models that more accurately model the ice-bottom smoothness.

A simulated example of how binary costs may be assigned to a transition between consecutive range-lines can be seen in Fig. 2.7.

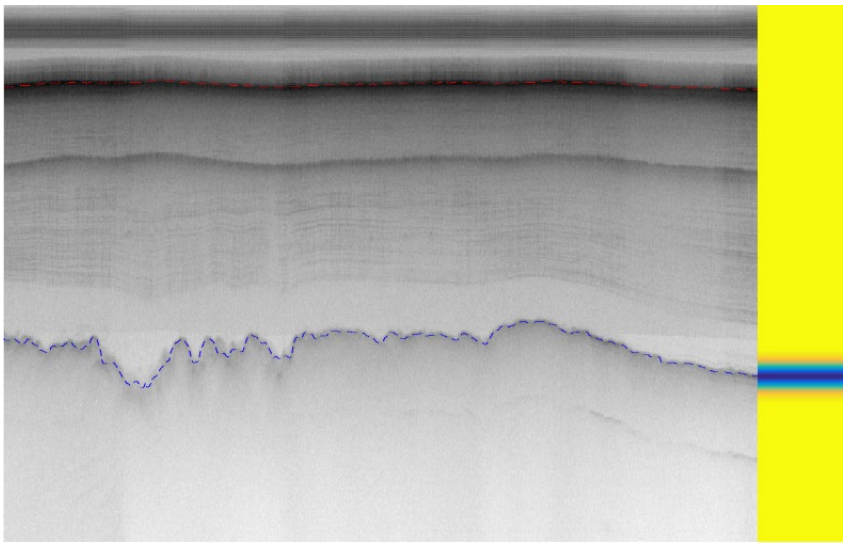

Fig. 2.7. Simulated example of the range-line-to-range-line transition model costs. The vertical bar on the right represents the binary cost assigned to each range-bin of the upcoming column at the right side of the image: darker blue represents a lower binary cost, and yellow represents a higher binary cost.

In the original solutions of Crandall *et al.* [14] and Xu *et al.* [17], the implementation of the binary cost term was set to prioritize flat surfaces in the coordinate systems of the 2D and 3D imagery, by simply assigning no additional cost to the range-bin which generates a perfectly flat transition and increasing the cost of transitions to range-bins above or below the current one, therefore equally penalizing all vertical *jumps* (regardless of *jumping* up or down) between consecutive range-lines or direction-of-arrival bins. The binary costs assigned to along-track and cross-track transitions were equal and calculated in the same manner.

This term was defined as a zero-mean Gaussian probability distribution transformed into a quadratic function by using a negative logarithm. This is done both for computational efficiency and to reduce rounding errors associated with taking products of many probability distribution functions since the logarithm converts the probability distribution products into summations of the log-probabilities. This term was mathematically defined as

$$\psi_B(s_{c_i}, s_{c_j}) = w_B * -\ln(\mathcal{N}(s_{c_i} - s_{c_j}, 0, \sigma_B^2) \tag{8}$$

where s_{c_i} and s_{c_j} are respectively the row index (range-bin) of the consecutive source and destination range-lines (in the 2D and 3D cases) or direction-of-arrival bins (in the 3D case) and therefore $(s_{c_i} - s_{c_j})$ is the range-bin offset between the source and destination, and $\mathcal{N}$ is the Gaussian (normal) probability density function with the usual definition of

$$\mathcal{N}(x, \mu, \sigma^2) = \frac{1}{\sqrt{2 * \pi * \sigma^2}} e^{-\frac{(x-\mu)^2}{2\sigma^2}} \tag{9}$$

where σ_B^2 is the desired variance of the Gaussian distribution, and w_B is a scalar weighting factor that determines the impact of the binary cost ψ_B in the total cost calculation. The binary cost ψ_B of all invalid transitions (i.e., between non-consecutive range-lines or DoA bins) was set to ∞,

ensuring that these would never be selected by the algorithms. See Fig. 2.8 for an example of this previous ψ_B term.

However, since the 3D data are in the native cylindrical coordinate system native of the radar sounding processing, this original binary cost function which assigned minimum cost to a flat surface in a cylindrical coordinate system did not assign minimum cost to a flat topography in the Cartesian coordinate system which we believe is the more likely scenario for most scenes. Specifically, a flat surface is not represented as a flat (i.e. perfectly horizontal) layer as a function of the directions of arrival. A flat surface in Cartesian coordinates curves downward towards the edges of the 3D imagery (see Fig. 1.8 for an example of this effect). Also, if the aircraft altitude changes, both the ice-surface and ice-bottom will change together with altitude.

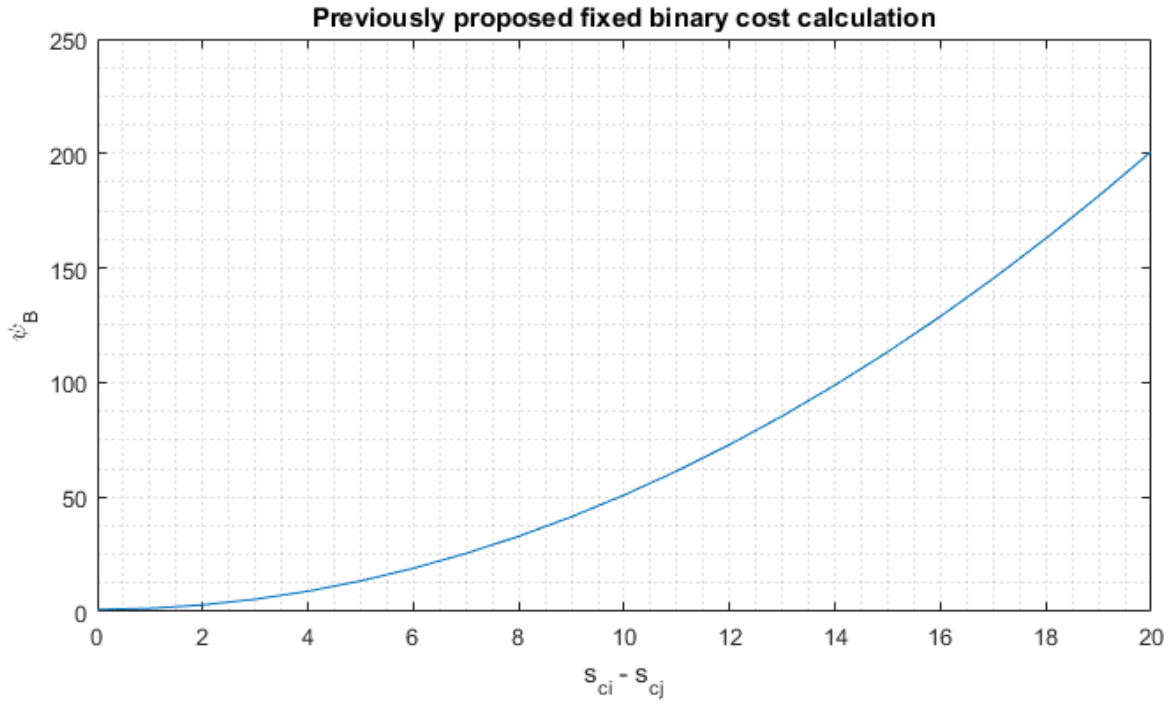

Fig. 2.8. The previously proposed *fixed* binary cost function as defined by Eq. (7), which increased the binary cost of both along-track and cross-track transitions according to a simple quadratic function. The values demonstrated above are presented before the application of the weighting factor w_B and vary depending on the σ_B^2 parameter, here shown with unit variance ($\sigma_B^2 = 1$).

For this reason, one potential improvement to this term involves setting the lowest transition costs to range-bins that follow the range-slope of the ice-surface. In the 3D imagery scenario, although this is still not a flat surface in the Cartesian coordinate system, this is a flatter and more realistic approximation of the expected shape of the ice-bottom, and computationally simpler than calculating the shape of a flat ice-bottom in Cartesian space which must account for ice refraction from a non-flat ice-surface layer. According to this proposed modification, the binary cost may then be calculated as

$$\psi_B(c_i, c_j) = w_B * [s_{c_i} - s_{c_j} - \left(s_{SURF}(c_i) - s_{SURF}(c_j)\right)]^2 \quad (10)$$

where s_{c_i} and s_{c_j} are the row-indices assigned to adjacent range-lines or direction-of-arrival bins, and therefore $\left(s_{SURF}(c_i) - s_{SURF}(c_j)\right)$ corresponds to the range-slope of the ice-surface between the coordinates of interest.

In the along-track dimension, this modification is not as helpful since a flat (i.e. perfectly horizontal) ice boundary in the along-track dimension does approximately depict flat topography; approximately because it does not account for the change in radar wave propagation speed in ice versus air or changes in platform altitude. Additionally, although it is a low correlation relationship, there is often correlation between the range-slope of the ice-surface and the shape of the ice-bottom layer [36, 37]; thus, this cost term is used for both 2D and 3D data and for both the Viterbi and TRW-S algorithms in the along-track dimension.

As an improvement to both of the aforementioned binary cost calculations based on fixed layer smoothness and following the range-slope of the surface, we propose an empirically-derived binary cost function similar to what was described for the distance-to-ice-margin term of the unary cost function. Based on manually-corrected ground-truth data, we are able to compute more

accurate transition models for both along-track and cross-track dimensions. Both of the new transition models calculate binary costs based on the variances of the distributions of vertical difference, in units of range-bins, observed when an along-track or cross-track transition is performed.

Finally, after unary costs are assigned to all valid pixels and binary costs are assigned to all valid along-track (in the 2D and 3D cases) and cross-track (in the 3D case) transitions, the cost-minimization framework is formulated as

$$E(S) = \sum_{c=1}^{N} \psi_U(s_c, c) + \sum_{p=(c_i, c_j)}^{p \in P} \psi_B\left(s_{c_i}, s_{c_j}\right) \tag{11}$$

where s_c is the row index through which the ice-bottom layer passes at column c, and N is the number of columns in the input image, which is equal to N_{RL} in a nadir-looking 2D echogram and N_{DoA} in a slice of three-dimensional imagery. In addition, P is the set of all 2-tuples that represent neighboring range-lines or DoA bins, and $E(S)$ is the energy function which the Viterbi and TRW-S algorithms are tasked with minimizing. The notation of s_{c_i} and s_{c_j} is presented in Eq. 10.

2.6 - Data pre-processing steps

Along with novel unary and binary cost terms, we propose an improved combination of pre-processing steps applied to nadir-looking 2D radar imagery, which we have experimentally found to improve tracking performance. We now list and describe these pre-processing routines.

The 2D image intensity exhibits a strong dependence on depth in ice due to ice loss and spherical spreading loss. We apply a simple detrending routine that normalizes the mean intensity of each row. This helps the tracker in areas where the bed echo is weak. Without detrending, clutter

near the ice-surface is often so strong that the ice-bottom layer tracker may jump to this signal despite the layer smoothness and surface-repulsion constraints enforced by the cost functions of the models. See Fig. 2.9 for a demonstration of the effects of the detrending process.

The first proposed Viterbi solution [14], which dealt with 2D images only, used image gradients and cumulative max gradients to handle the dynamic range. While the 2D images are estimates of scatterer intensity from MVDR, the 3D images are generated from the MUSIC cepstrum and a similar detrending procedure is not necessary because MUSIC produces a muted dynamic range. In [17], which dealt with the 3D MUSIC images, a simple thresholding technique was used to reduce the dynamic range. With this approach, every pixel of the input image with value greater than a certain manually tuned threshold was made equal to the threshold value to reduce the difference between the pixel values of the ice-surface and ice-bottom layers to prevent the automated tracker from selecting the (often) stronger ice-surface return as the ice-bottom. The problem with thresholding is that the shape and strength of the return is distorted by the thresholding and occasionally the ice bottom would exceed this threshold and be clipped as well. The thresholding step was removed for the 3D images in our modified approach.

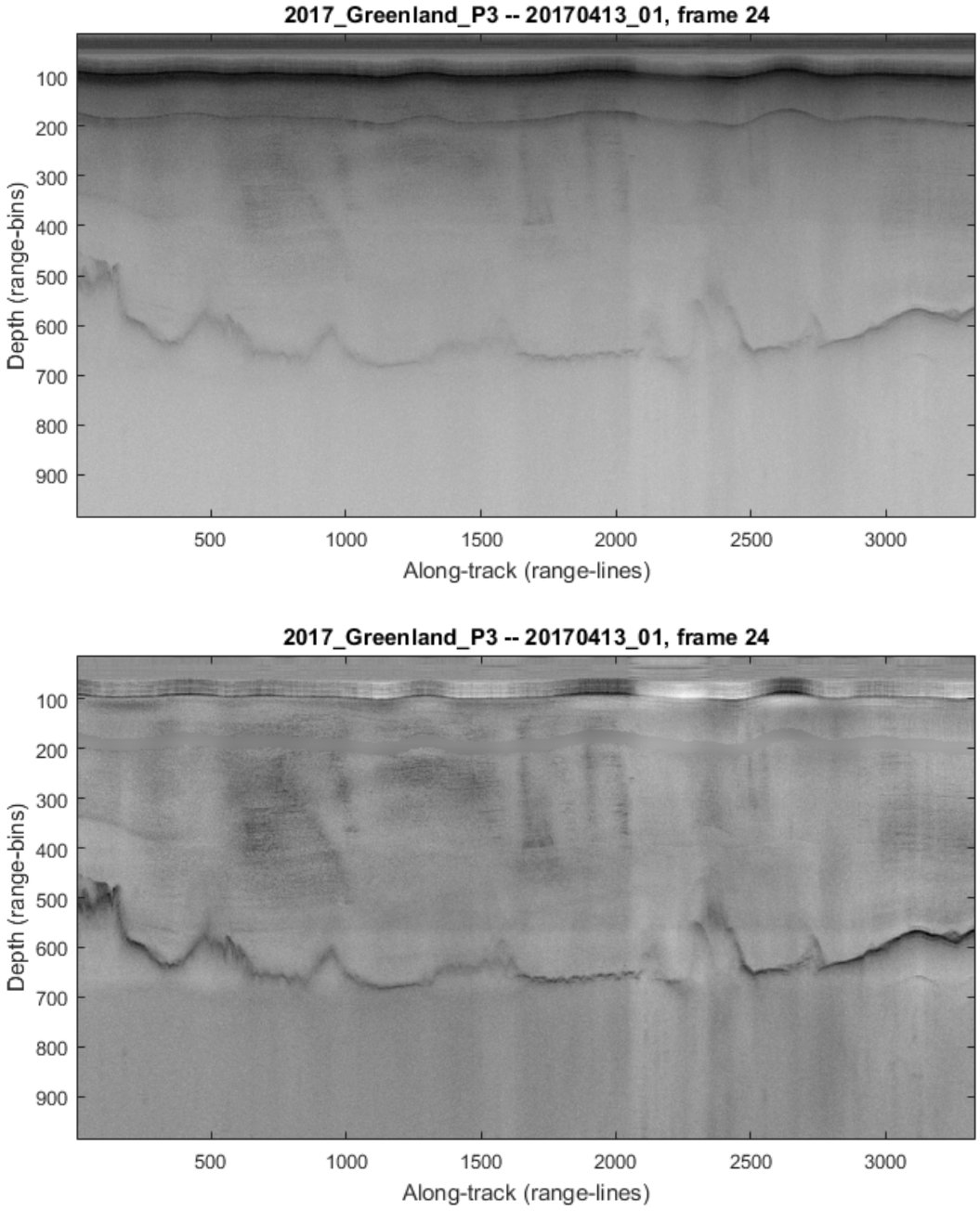

Fig. 2.9. A demonstration of the effect of the detrending pre-processing step applied to all nadir-looking 2D echograms in this tracking framework. The top image displays the original echogram; note how the ice-bottom interface is faint and blurry in several areas. These become significantly clearer and more apparent in the second image, which displays the same echogram after the detrending process. This is the same echogram displayed in Figs. 1.7 and 1.8.

An undesirable feature present in the 2D imagery is the surface multiple, which is caused by a ringing of the radar signal between the ice-surface and the aircraft. This is also seen in the 3D imagery, but the tracking is not as affected. To mitigate the effect of the surface multiple in 2D imagery as a false positive to the algorithms, we employed a simple method of smoothing the input image around the areas in which the surface multiple is located. This was achieved by replacing from 20 rows above to 20 rows below the surface multiple with a blurred version of the image. The image is blurred with a 2D Gaussian filter with standard deviation equal to 50 pixels and kernel size equal to 201 pixels. It is possible to estimate the location of the first surface multiple by doubling the two-way travel time of the ice-surface. We have experimentally found that this multiple suppression approach does not significantly slow down the tracking or decrease tracking accuracy significantly even if the ice-bottom layer is located within the rows to which the filtering was applied, due to the smoothness and continuity constraints enforced by the unary cost function. We did, however, find the rate of mislabeling the ice-bottom due to the surface multiple to have been largely decreased. See Fig. 2.10 for a demonstration of multiple suppression, and Fig. 2.11 for an example of tracking results with and without it.

Additionally, previous 2D tracking efforts performed layer tracking on the 2D echograms of individual data frames. This sometimes resulted in lower quality results near the edges of the data frames, because evidence from contiguous frames was not being considered. For this reason, we horizontally concatenated the two-dimensional data frames before passing them to the Viterbi algorithm so that all adjacent data frames (usually entire flights) are tracked at once. This also increases the probability that the image being processed will include automated high-confidence ground truth from crossovers, although this tends to have a relatively local effect on improving performance.

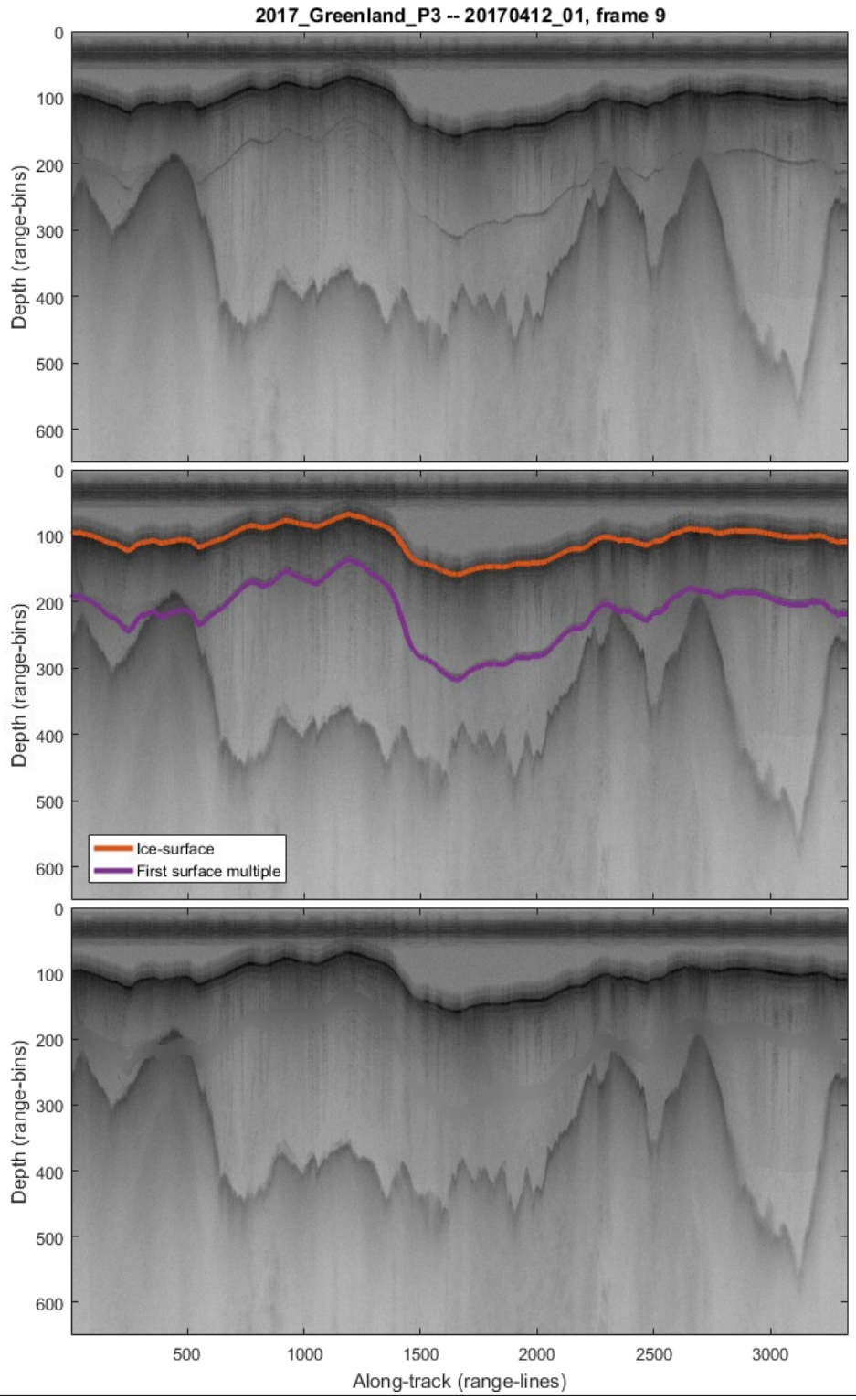

Fig. 2.10. Demonstration of the multiple suppression technique. From top to bottom: the original echogram with no layers depicted; the same echogram with the existing ice-surface layer (in orange) and the first surface multiple (in magenta) calculated as twice the two-way travel time of the ice-surface; the echogram after application of the multiple suppression step. Note how the first surface multiple, explicitly depicted in the second image, has been blurred out in the result shown in the third image. See next page for an example including the tracking results.

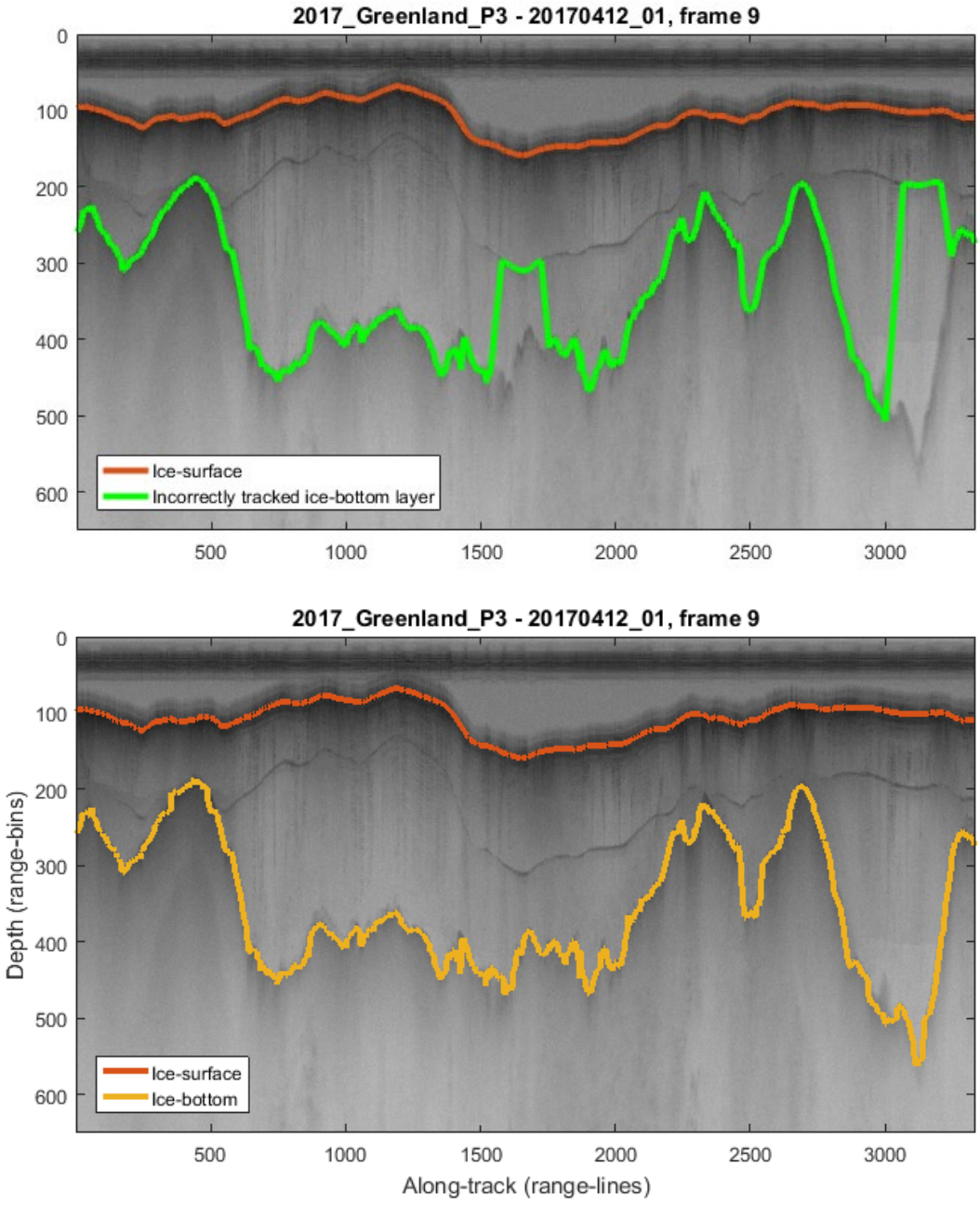

Fig. 2.11. The same 2D nadir-looking echogram as presented in Fig. 2.9, with the existing ice-surface layer (in orange) and the ice-bottom layers found by the Viterbi algorithm (in yellow) overlaid on the image. The result on top was generated with no multiple suppression applied to the original image. Notice how on two separate occasions, starting around range-lines 1,500 and 3,000 the automated tracker incorrectly *jumps* up and selects the first surface multiple as the result for the ice-bottom layer, even with the layer smoothness constraints included in the tracking models. The result on the bottom was obtained with the multiple suppression process of filtering the original image as described in the text. With this pre-processing step, the tracker correctly selects the ice-bottom as can be seen in the second image above. No other pre-processing steps or input parameters (including ground-truth points) were modified in the generation of these results.

2.7 - Inputs provided to the HMM/Viterbi and MRF/TRW-S trackers

The purpose of this section is to list and describe the inputs provided to the HMM/Viterbi and MRF/TRW-S ice-bottom trackers and to describe the technical details of the specific implementations to support use of the tools. All presented inputs are common to both trackers, except when stated otherwise, but often have different dimensions and interpretations between the two. The most essential difference between the inputs is that the HMM/Viterbi tracker receives only a single two-dimensional image (nadir-looking echogram or a single slice of 3D imagery) at a time, while TRW-S receives a 3D image composed of two or more slices. While the MRF model and the TRW-S algorithm in general may be applied to tracking of a single slice at a time, our implementation of this framework requires that two or more slices be passed in.

Several inputs are required, in the sense that the algorithms need some value to use in the calculations, but need not necessarily be passed in because a certain fixed (hard-coded) value is used if no value is provided for that variable, and thus these inputs may be left empty. In several cases, using pre-defined values has an overall tendency of decreasing the performance of the algorithms due to the values being somewhat generic or not employing the cost function improvements proposed in this work such as the empirically-derived calculations obtained from geostatistical analysis. The applications and effects of the inputs are described for both algorithms in Sections 2.3 and 2.4 of this document.

The graphical interface for browsing, verifying, and correcting CReSIS radar echograms is written in Matlab. Both tracking algorithms are implemented as routines developed in C++ and invoked by the Matlab interpreter as a function via the Matlab Executable (MEX) framework. Both the interface and the tracking functions are cross-platform and have been verified to perform identically on Windows and Linux systems.

In the order they were integrated into the cost functions and are passed in to the tracking software, the inputs are as follows:

1. Image data. This input is the matrix containing the radar imagery itself. In the case of 2D radar echograms, handled only by the HMM/Viterbi algorithm framework, this is a two-dimensional matrix of dimensions N_{RB} rows and N_{RL} columns, where N_{RB} is the number of range-bins and N_{RL} is the number of range-lines of the echogram. In the case of 3D radar imagery, the Viterbi algorithm can perform tracking of individual slices, as previously mentioned; in this case, the input is a two-dimensional matrix of size N_{RB} rows and N_{DoA} columns, where N_{DoA} is the number of direction-of-arrival bins of the given slice. The TRW-S algorithm, however, must be provided with a sequence of slices at a time, in the form of a three-dimensional matrix of size N_{RB}-by-N_{DoA}-by-N_{RL}. Note that N_{RL} corresponds to the number of slices that are simultaneously provided to TRW-S, with the requirement that $N_{RL} > 1$ in our implementation of the algorithm. In order to enhance contrast, the image input is provided in the form of $20 * \log_{10}(I)$ in all cases, where I is the input image matrix. Other pre-processing steps applied to the 2D and 3D radar echograms are detailed in Section 2.5.
2. Location of the ice-surface layer. This input consists of the row (range-bin) values that label the ice-surface layer, which can be acquired from existing third-party airborne or spaceborne laser altimetry instrumentation [19, 20]. It is provided to the Viterbi algorithm as a row-vector of the same size as the number of columns in the image: a N_{RL}-dimensional vector in the nadir-looking 2D echogram case or a N_{DoA}-dimensional vector in the case of an individual slice. It is provided as a N_{DoA}-by-N_{RL} matrix to the TRW-S algorithm, therefore corresponding to one value for each DoA of every slice being processed.

3. High-confidence ground-truth points. This input corresponds to ψ_{GT} in Eq. 3 and Eq. 7, and consists of the set of high-confidence ground-truth points used in the unary cost calculation. The Viterbi algorithm receives these as 2-by-N_{HCGT} matrix which corresponds to a sequence of range-bin/range-line pairs in the 2D case, or range-bin/DoA-bin pairs in the 3D case. The equivalent input to the TRW-S algorithm is a 3-by-N_{HCGT} matrix which corresponds to a sequence of range-bin/slice/DoA-bin 3-tuples. In both cases, N_{HCGT} is the number of high-confidence ground-truth points available for the dataset of interest. May be left empty and no fixed values are used in that case.
4. Low-confidence ground-truth points. This input corresponds to ψ_{EXTRA} in Eq. 3 and Eq. 7, and consists of the set of low-confidence ground-truth points used in the unary cost calculation. The Viterbi algorithm receives these as a 2-by-N_{LCGT} matrix which corresponds to a sequence of range-bin/range-line pairs in the 2D case, or range-bin/DoA-bin pairs in the 3D case. The equivalent input to the TRW-S algorithm is a 3-by-N_{LCGT} matrix which corresponds to a sequence of range-bin/slice/DoA-bin 3-tuples. In both cases, N_{LCGT} is the number of low-confidence ground-truth points available for the dataset of interest. May be left empty and no fixed values are used in that case.
5. Ice-mask. This input is a binary raster interpolated to describe every range-line and direction-of-arrival bin of data displayed by the echogram of interest. Each element of the raster determines the existence of ice at a certain coordinate when equal to 1, and the inexistence of ice if equal to 0. Typically generated by third-party researchers from satellite imagery, the ice-mask is available for most regions surveyed by CReSIS. See, for example, the Randolph Glacier Inventory [34] for more information. In our tracking solutions, the ice-mask has the same size as input #2 (location of the ice-surface layer) in all cases.

6. Values of the correlation function $\mu(p)$. This input defines the values of the correlation function $\mu(p)$ used in the image intensity term of the unary cost function, as presented in Eq. (2). In all cases, this input is of size 1-by-$N_{\mu(p)}$, where $N_{\mu(p)}$ is an arbitrary positive natural number. In this work, $N_{\mu(p)} = 11$ and the values of the $\mu(p)$ function used are shown in Fig. 2.2.
7. High-confidence ground truth weight. This input corresponds to w_{GT} in Eq. 7, which scales the effect of the high-confidence ground-truth in the unary cost function. May be left empty and a pre-defined value is used in that case.
8. Smoothness weight. Used as a scaling factor used in the binary cost function. Provided to the Viterbi algorithm as a scalar, and as a 2-by-1 vector to TRW-S. In the Viterbi algorithm, controls column-to-column transition weights; in TRW-S, the first element controls cross-track transition weights and the second element controls along-track transition weights. A pre-defined value is used if left empty.
9. Slope of the ice-surface. Represents the range-slope of the ice-surface, and has the same size as input #2 (location of the ice-surface layer) in all cases. Calculated as the first derivative of the ice-surface vector (in the 2D case) and as the first derivative of ice-surface of every slice (in the 3D case). Used in the binary cost function. For example, if a portion of a 2D ice-surface is defined in terms of range-bins as

   $$\text{ICE-SURFACE} = [\cdots,100,101,99,100,100,101,\cdots]$$

   then the range-slope corresponding to that portion is equal to

   $$\text{RANGE-SLOPE} = [\cdots,1,-2,1,0,1,\cdots]$$

   and the same calculation is performed in the 3D case where the ice-surface is passed in as matrix corresponding to the value for every direction-of-arrival bin of every slice.

10. Bounds. Used whenever the automated tracking should ignore certain extreme DoA bins (far left or right edges) of the input image. While typically not applicable to nadir-looking 2D echograms, this is often useful in the tracking of slices of 3D imagery since the extreme direction-of-arrival bins have the lowest quality and may be ignored. In the case of Viterbi tracking of 3D slices, it is passed in as a 2-vector representing the starting and ending valid DoA bins; for example, setting this to equal $[3, 60]$ will cause the tracker to ignore the 3 leftmost and 4 righmost DoA bins in a standard 64 DoA bin slice, and no output will be provided for the ignored DoA bins. A similar 4-vector is passed in to TRW-S, in which the 4 elements represent the starting and ending valid DoA bins and the starting and ending valid range-lines for tracking, respectively. A constraint on the valid range-lines is typically not used. A pre-defined value in which no DoA bins or range-lines are ignored is used if left empty.
11. Low-confidence ground-truth weight. This scalar input corresponds to w_{EXTRA} in Eq. 7, which scales the effect of the low-confidence ground-truth in the unary cost function. May be left empty and a pre-defined value is used in that case.
12. Distance to nearest ice-margin. A vector or matrix indicating the shortest distance between every column of the input image (in every range-line, in the 3D case) to an ice-margin. Used in the calculation of ψ_{DIM} in the unary cost function; see Eq. 7 and Chapter 3 for more information on this topic. This input has the same size as input #2 (location of the ice-surface layer) in all cases. May be left empty and the previously proposed ψ_{REP} and ψ_{SINC} terms are used in that case.

13. Distance to nearest ice-margin cost matrix. This matrix defines the unary cost term

$$\psi_{\mathrm{DIM}} = f(\mathrm{T}, \mathrm{D}) \tag{12}$$

as a function of the calculated ice-thickness T of the pixel of interest and as a function of its distance to the nearest ice-margin D. The values of D for every pixel are directly taken from the previous input. An explanation on how these values are obtained from the geostatistical analysis is provided in Chapter 3. For both 2D and 3D imagery and for both HMM/Viterbi and MFR/TRW-S solutions, this matrix has arbitrary size in both dimensions, but must contain at least one element. The DIM is provided in units of meters and the ice-thickness is provided in units of range-bins. Pixels with DIM or thickness greater than the maximum values defined in the matrix are thresholded to the maximum defined in the matrix. May be left empty and the previously proposed ψ_{REP} and ψ_{SINC} terms are used in that case, and may not be used if input #12 (distance to ice-margin matrix) has been left empty.

14. Variances for the along-track transition model. This one-dimensional vector defines the variances of the probability distributions used in the along-track transition model, which is part of the binary cost function of both tracking models. This input is passed in as a scalar value to the HMM/Viterbi tracking technique in the case of a 2D nadir-looking echogram. This scalar corresponds to the variance, in units of range-bins, discovered for the nadir DoA bin. This input is not provided to the HMM/Viterbi tracker when processing individual slices of 3D imagery. It is passed in as a 1-by-$\mathrm{N}_{\mathrm{DoA}}$ vector to the MRF/TRW-S framework, with each element corresponding to the variance discovered for each direction-of-arrival bin in the input image. May be left empty and the previously proposed fixed along-track binary term is used in that case.

15. Variances for the cross-track transition model. This one-dimensional vector defines the variances of the probability distributions used in the cross-track transition model, which is part of the binary cost function of both tracking models. Passed in as a 1-by-N_{DoA} vector to both tracking models in the case of tracking three-dimensional imagery. This input is not provided to the HMM/Viterbi tracker when processing two-dimensional nadir-looking imagery. May be left empty and the previously proposed fixed cross-track binary term is used in that case.

See Table 2.1 for a summary of the inputs described above. A list of the symbols used is provided below.

N_{RB}	Number of range-bins in the input image
N_{RL}	Number of range-lines in the input image
N_{DoA}	Number of direction-of-arrival bins in the input image
N_{HCGT}	Number of high-confidence ground-truth points
N_{LCGT}	Number of low-confidence ground-truth points
$N_{\mu(p)}$	Length of the correlation function $\mu(p)$ used in the ψ_{SINC} unary cost term
T_{MAX}	Maximum ice-thickness T defined by $f(T, D)$ of the ψ_{DIM} unary cost term
D_{MAX}	Maximum DIM D defined by $f(T, D)$ of the ψ_{DIM} unary cost term

Input type	Dimension (HMM, 2D data)	Dimension (HMM, 3D data)	Dimension (MRF)	Required	May be left empty
Image data	N_{RB}-by-N_{RL}	N_{RB}-by-N_{DoA}	N_{RB}-by-N_{DoA}-by-N_{RL}	✓	⊗
Ice-surface layer	1-by-N_{RL}	1-by-N_{DoA}	N_{DoA}-by-N_{RL}	✓	⊗
High-confidence GT	2-by-N_{HCGT}	2-by-N_{HCGT}	3-by-N_{HCGT}	⊗	✓
Low-confidence GT	2-by-N_{LCGT}	2-by-N_{LCGT}	3-by-N_{LCGT}	⊗	✓
Ice-mask	1-by-N_{RL}	1-by-N_{DoA}	N_{DoA}-by-N_{RL}	✓	⊗
Correlation function $\mu(p)$	1-by-$N_{\mu(p)}$	1-by-$N_{\mu(p)}$	1-by-$N_{\mu(p)}$	✓	⊗
High-confidence GT weight	Scalar	Scalar	Scalar	✓	✓
Smoothness weight	Scalar	Scalar	2-by-1	✓	✓
Ice-surface range-slope	1-by-N_{RL}	1-by-N_{DoA}	N_{DoA}-by-N_{RL}	✓	⊗
Bounds	Not applicable	2-by-1	4-by-1	⊗	✓
Low-confidence GT weight	Scalar	Scalar	Scalar	✓	✓
DIM	1-by-N_{RL}	1-by-N_{DoA}	N_{DoA}-by-N_{RL}	⊗	✓
DIM cost values	T_{MAX}-by-D_{MAX}	T_{MAX}-by-D_{MAX}	T_{MAX}-by-D_{MAX}	⊗	✓
Variances for along-track transition model	Scalar	Not applicable	1-by-N_{DoA}	⊗	✓
Variances for cross-track transition model	Not applicable	1-by-N_{DoA}	1-by-N_{DoA}	⊗	✓

Table 2.1. Inputs provided to the HMM/Viterbi and MRF/TRW-S trackers.

Chapter 3 - GEOSTATISTICAL ANALYSIS

3.1 - Theory and motivation

As mentioned throughout Chapter 2, the unary and binary cost functions employed in all previous iterations of the HMM and MRF-based ice-bottom tracking solutions [14, 17, 30] were composed of fixed calculations that attempted to match the assumptions and constraints of the ice sheets. While favorable results were obtained by the previous models, they were limited by the difficulty of discovering proper cost formulations and tuning the numerous parameters that numerically define them. Intuitive manual estimation of these parameters is challenging because of the complexity of the models. Automated tuning is computationally expensive because it requires tracking a large dataset many times using different parameter combinations and comparing the results with the ground-truth.

Previous iterations of the HMM and MRF models also presented other problems. For instance, the previously-proposed ψ_{REP} term of the unary cost function equally penalized all pixels for being near the ice-surface, regardless of the expected ice thickness. However, it is sensible to assume that icy areas very near the ice-margin are more likely to have very thin ice and thus pixels that are much deeper than the ice-surface should be penalized in this case. Similarly, the parameters used in the previous ψ_{SURF} term, while useful in generating smooth transitions between icy and non-icy areas, were initial educated guesses and not tuned. This unary cost term was defined to force the ice-bottom layer to lie within a certain distance (in the range dimension) to the ice-surface in all cases where the pixel of interest was within a certain distance (in the along-track or cross-track dimensions) to a point with no ice (in other words, to the nearest ice-margin). The proper values for these two distances, however, are difficult to intuitively estimate or automatically tune because of the aforementioned problem with parameter tuning dimensionality. This obstacle is

found in both HMM and MRF models and thus affects all ice-bottom tracking in the previously presented framework. Furthermore, as detailed in Section 2.5 of this document, the proposed binary cost models reward transitions that adhere to the constraints of smoothness and matching the range-slope of the ice-surface layer. For this reason, in past iterations of the HMM/MRF frameworks, areas of rugged bedrock topography show a degradation in tracking performance. Furthermore, the cross-track transition model applied by the original binary cost function presented in Eq. 8 did not allow for any variation in the expected distribution of the transitions along the elevation angle dimension even though the ice-bottom layer tends to be flatter around nadir and to curve downward towards the extreme elevation angles. The reformulated binary cost function, presented in Eq. 10, mitigates this problem by accounting for the range-slope of the ice-surface, but the range-slope of the ice-surface is unlikely to be perfectly matched to the range-slope of the ice-bottom layer.

As a partial solution to the problems presented above, we have conducted a geostatistical analysis using the existing manually-corrected ground-truth two-dimensional and three-dimensional layer data, from which three probability distributions were calculated. These distributions are used to define empirically-derived cost terms for both the unary and binary cost functions. These consist of a novel ψ_{DIM} term in the unary cost function, which replaces the previous ψ_{SURF} and ψ_{REP} terms, and more sophisticated along-track and cross-track transition models for the binary cost function.

The final values for the ψ_{DIM} term for 2D and 3D data were derived separately, using two-dimensional and three-dimensional ground-truth layer data, respectively. The ψ_{DIM} term resulting from each was applied only to tracking of the respective type of imagery. Similarly, a geostatistical analysis was separately conducted using the 2D and 3D training dataset to discover the along-track

transition models appropriate to each type of imagery. As the cross-track transition model is exclusive to three-dimensional imagery tracking, only the 3D training set was used in the derivation of this term.

The new cost terms are based on parametric models of the histograms from ground-truth layer data. Modifications with the purpose of generalizing beyond the training set's limitations were made to the ψ_{DIM} unary cost term discovered from the three-dimensional data, since the three-dimensional training set used does not include very thick ice or ice far from the ice margin.

In all cases, basing the unary and binary cost calculations on actual first and second order statistics of ground-truth data should provide more realistic models assuming the geostatistics are sufficiently stationary and the Gaussian approximations used are close enough matches. Each of the three modifications proposed here is presented as a separate section in this chapter.

The datasets used for all geostatistical analysis calculations are the training subset of the two-dimensional and three-dimensional datasets presented in Chapter 4. The two-dimensional dataset was acquired over Greenland and the Canadian Arctic Archipelago (CAA) during the 2014 NASA Operation IceBridge arctic deployment. The three-dimensional dataset is a proper subset of the 2D dataset, containing only data acquired over the CAA during the same deployment. The two-dimensional ground-truth layer dataset was manually tracked by experienced ice layer analysts, using the simple automated tools briefly described in Section 1.3 of this document. The ground-truth 3D layer dataset was initially tracked with the original MRF/TRW-S algorithm framework and then manually corrected by human annotators. More information about the datasets used in this work is provided in Chapter 4 of this document.

3.2 - Unary cost term as a function of ice thickness and distance to ice-margin

The novel probabilistic cost term ψ_{DIM} assigns to every valid pixel a cost value based on the ice thickness generated by that pixel and the distance between that pixel and the ice-margin nearest to it. In other words, this unary cost term is calculated as a function of the thickness T and distance-to-ice-margin D of the pixel of interest. This is represented as $\psi_{\mathrm{DIM}} = f(\mathrm{T}, \mathrm{D})$, which has been previously presented in Eq. 12. This cost term applies a unary cost to pixels based on where the current pixel's ice thickness falls in the ice thickness histogram for a similar DIM. The higher the estimated probability based on the histogram, the lower the cost will be. As mentioned before, the conceptual expectation is that areas very near the ice-margin will tend to have thin ice, whereas points with large DIM may have a large variety of ice thickness measurements but it is unlikely that these points will contain very thin ice.

The purpose of this probabilistic term is two-fold. First, it generates ice-bottom layers with smooth transitions between icy and non-icy areas based on the test set's measured geostatistics, and therefore it serves as a replacement for the previously-proposed ψ_{SURF} term. Because of the discovered geostatistical values, the ψ_{DIM} cost term assigns low costs to pixels that generate thin ice measurements in areas with low DIM, and the cost for generating thin ice measurements steadily rises as the distance-to-ice-margin increases. This cost distribution generates a smooth approximation of the ice-bottom layer towards the ice-surface as the coordinate of interest approaches the ice-margin.

The second purpose of this probabilistic term is that it serves as an empirically-derived substitute for the previous ψ_{REP} term of the unary cost function, which prevents the layer tracker from mistakenly labeling the strong ice-surface return as the ice-bottom layer. This is due to the fact that pixels with large DIM and low ice thickness will most likely not be selected by the

algorithms due to their increased ψ_{DIM} cost, unless the other cost terms are so favorable they compensate for the increased cost assigned by this term. In other words, ice in areas with relatively high distance-to-ice-margin (such as greater than 1,000 meters) will not be tracked as being thin, since this is physically unlikely, unless the other unary cost terms (ψ_{SINC}, ψ_{GT}, and ψ_{EXTRA}) are low enough that they compensate for the increased cost of going against the expected high ice thickness.

The total distance between a given coordinate and the ice margin nearest to it is calculated as the rounded Euclidean distance transform of the binary ice-mask raster. For each element in the ice-mask, the distance transform assigns a number equal to the distance between that element and the nearest zero-element (which therefore indicates an ice-margin). Table 3.1 provides an example of a two-dimensional binary ice-mask raster with the convention that a value of 1 indicates the presence of ice and a value of 0 indicates the inexistence of it at each coordinate. Table 3.2 shows the results of the Euclidean distance calculations on the initial matrix. Table 3.3 presents the final results after rounding the distance measurements.

DoA_{k+2}	1	1	1	1	1
DoA_{k+1}	1	0	1	1	1
DoA_{k}	0	0	1	1	1
DoA_{k-1}	0	0	0	1	1
DoA_{k-2}	0	0	1	1	0
	RL_{k-2}	RL_{k-1}	RL_{k}	RL_{k+1}	RL_{k+2}

Table 3.1. Example of a two-dimensional ice-mask raster.

DoA_{k+2}	1.4142	1	1.4142	2.2361	3.1623
DoA_{k+1}	1	0	1	2	2.8284
DoA_{k}	0	0	1	1.4142	2
DoA_{k-1}	0	0	0	1	1
DoA_{k-2}	0	0	1	1	0
	RL_{k-2}	RL_{k-1}	RL_{k}	RL_{k+1}	RL_{k+2}

Table 3.2. Result of the Euclidean distance transform applied to the raster in Table 3.1.

DoA_{k+2}	1	1	1	2	3
DoA_{k+1}	1	0	1	2	3
DoA_{k}	0	0	1	1	2
DoA_{k-1}	0	0	0	1	1
DoA_{k-2}	0	0	1	1	0
	RL_{k-2}	RL_{k-1}	RL_{k}	RL_{k+1}	RL_{k+2}

Table 3.3. Result of rounding the matrix in Table 3.2. This is the final distance-to-ice-margin input passed in to the tracking algorithms. This is input #12 presented in Section 2.6 of this document.

In the previous tables, DoA_i and RL_j represent arbitrary adjacent direction-of-arrival bins and range-lines, respectively. The DIM is calculated for every range-line in the case of 2D data. In the 3D case, which is shown in the tables above, the DIM is calculated for every range-line and direction-of-arrival bin. In order to normalize the resolution of range-lines and direction-of-arrival bins, all DIM measurements are converted to meters. Furthermore, the thickness measurement $\mathrm{T}(s, c)$ generated by a certain pixel is equal to the vertical distance, measured in range-bins, to the known ice-surface layer at that coordinate. This is mathematically described by

$$\mathrm{T}(s, c) = s - s_{\mathrm{SURF}}(c), \tag{13}$$

following the same notation used in Eq. 4.

We now focus on describing the computation of the ψ_{DIM} cost term itself. As previously mentioned, separate geostatistical analyses using the 2D and 3D ground-truth layer datasets were performed. In both cases, the first step is to generate the histogram of the ice thickness measurements for every DIM from the existing ground-truth datasets. Note that the 3D dataset used presents statistical limitations due to being mostly composed of narrow fjord glaciers in the relatively small ice caps of the Canadian Arctic Archipelago. This dataset presents points with distance to ice margin up to only approximately $2{,}500$ meters. Furthermore, ice thickness distributions in areas with high DIM (such as greater than 1,000 meters) are very sparsely defined due to the low number of occurrences of such points in the 3D dataset used. For this reason, we consider only the distributions found for DIM $\leq 1{,}000$ meters as statistically significant. Fig. 3.1 presents the two-dimensional histogram of ice thickness for all DIM measurements up to $2{,}500$ meters, obtained from the ground-truth three-dimensional layer dataset.

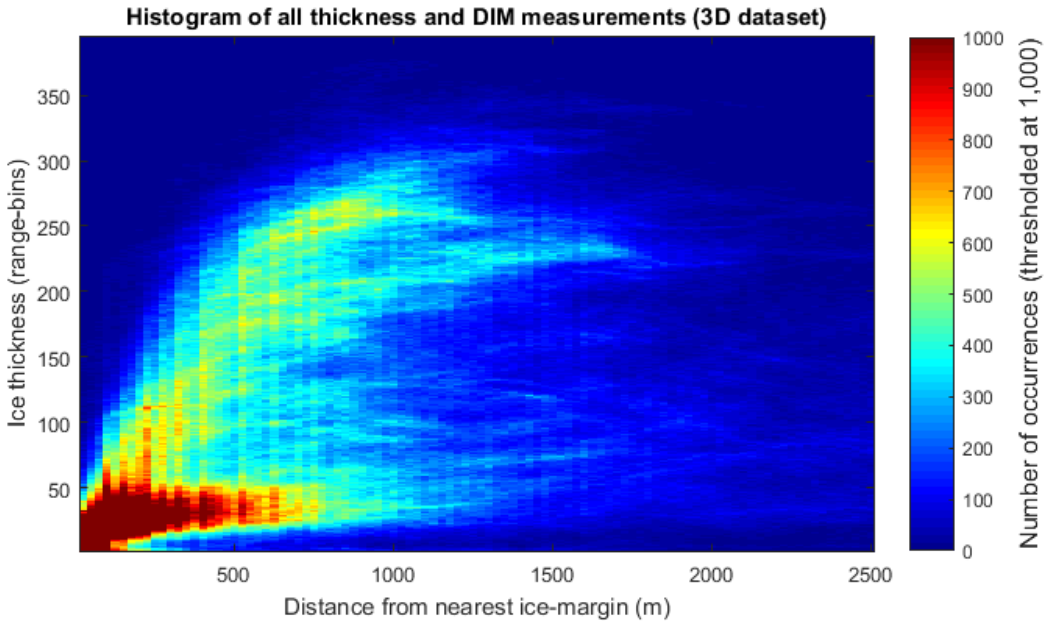

Fig. 3.1. Histogram of the ice-thickness measurements (vertical axis) found for all distances-to-ice-margin (horizontal axis) from the ground-truth three-dimensional dataset. The color of each pixel in the matrix above represents the number of occurrences for the corresponding combination of ice-thickness and DIM (see color bar on the right hand side). We thresholded the values depicted here to 1,000 for ease of visualization, but the values used in the actual cost calculations were not thresholded.

On the other hand, no similar limitation exists for the geostatistical analysis performed using the two-dimensional training dataset, which includes imagery acquired from the center of the Greenland continental ice sheet and presents data points with distance to ice margin in excess of 300 kilometers. In this case, we consider the distributions found for DIM $\leq 350{,}000$ meters as statistically significant. Fig. 3.2 presents the two-dimensional histogram of ice thickness for all DIM measurements up to 375 kilometers. For ease of visualization of this histogram, we have normalized each column of the matrix so that all values of each column add up to exactly 1. In the resulting image, the darker pixels represent a low occurrence of the DIM and ice thickness combination of interest, whereas the brighter pixels represent the frequently occurring combinations. Fig. 3.3 presents the same histogram but with the low-DIM/low-thickness data points in detail.

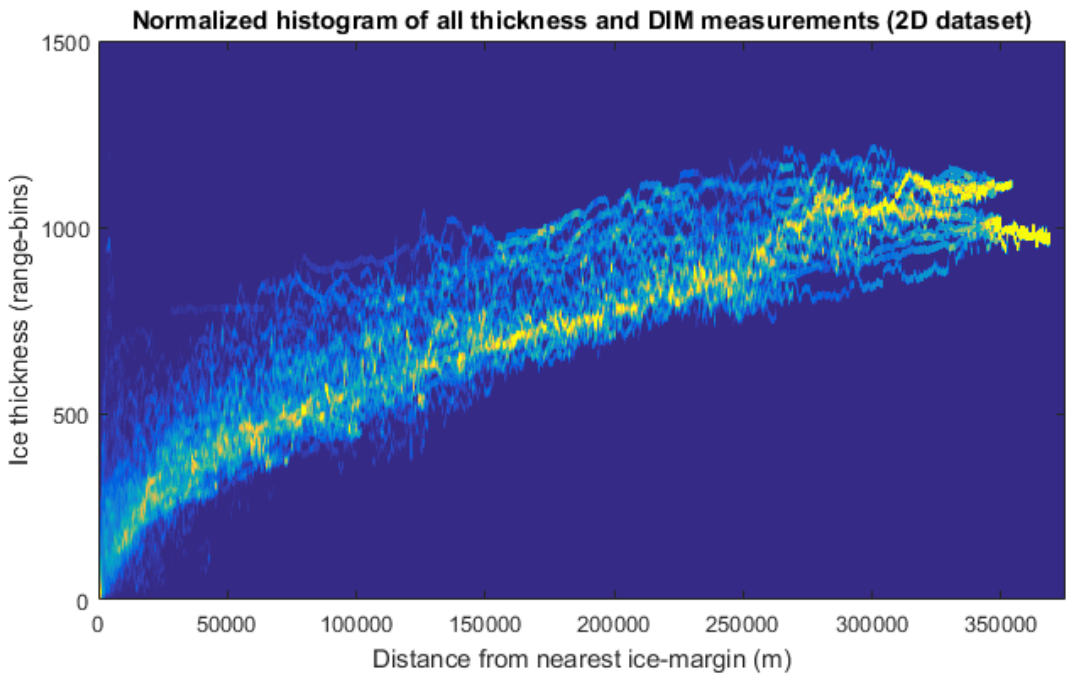

Fig. 3.2. Histogram of the ice-thickness measurements (vertical axis) found for all distances-to-ice-margin (horizontal axis) from the ground-truth two-dimensional dataset. For ease of visualization, we have normalized the histogram above so that the sum of the values in each column is equal to 1. The darker pixels represent a low occurrence of the DIM and ice thickness combination of interest, whereas the brighter pixels represent the frequently occurring combinations.

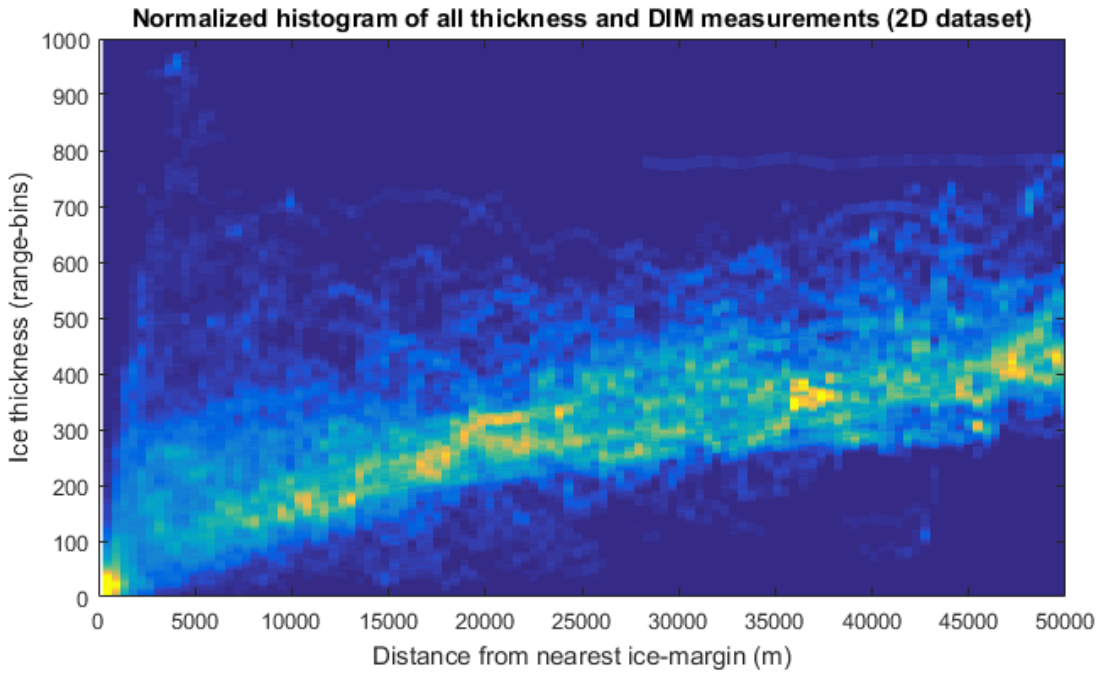

Fig. 3.3. Histogram presented in Fig. 3.2, with the low-DIM/low-thickness data points in detail.

We demarcate boundaries in the histograms presented above such that 90% of all data points in each column in the histogram matrix fall within the main interval, 5% fall over and 5% fall under the calculated limits. This allows for the distinction between a high probability area, defined by the points within the interval, and lower probability areas outside the interval. These intervals are defined up to the maximum distance to ice margin considered statistically significant in each case. In Fig 3.4, we present the histogram obtained from the three-dimensional dataset, up to DIM $\leq 1{,}000$ meters, with the red lines representing the boundaries of the main interval in which 90% of all data occurrences take place. We refer to these boundaries as the upper 5% bound, which defines for each DIM the ice thickness for which 5% of all data points present greater ice thickness, and similarly the lower 5% bound, which defines for each DIM the ice thickness for which 5% of all data points present lower ice thickness.

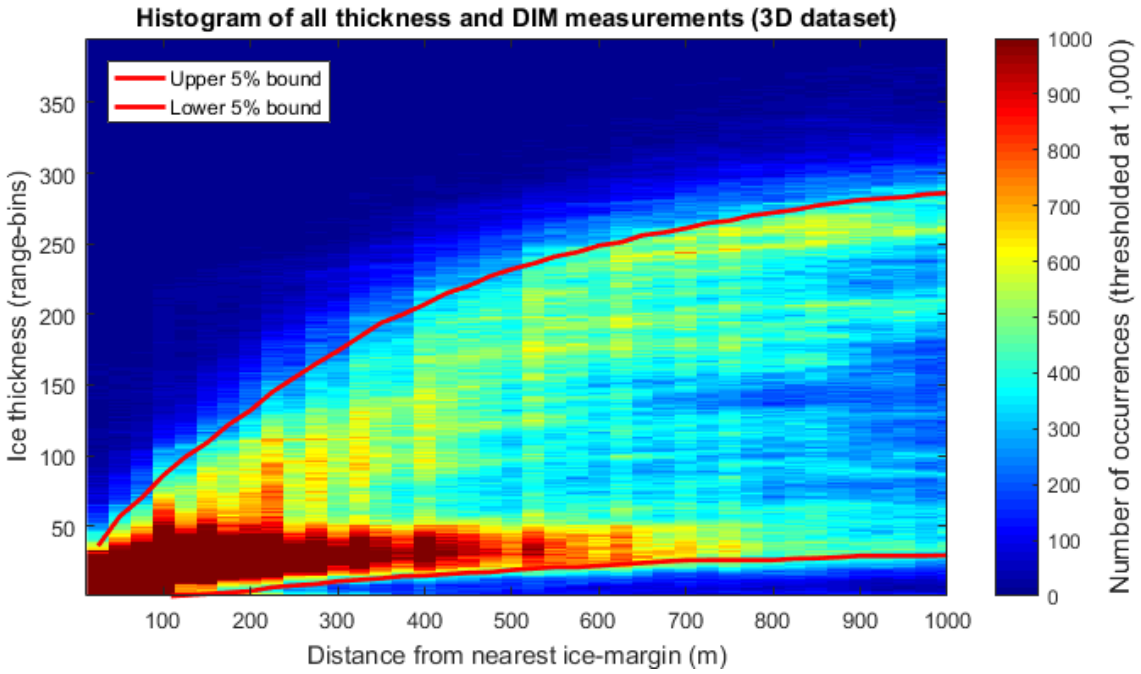

Fig. 3.4. Histogram of the ice-thickness measurements (vertical axis) found for all distances to ice margin (horizontal axis) from the ground-truth three-dimensional dataset, displayed up to the maximum distance to ice margin considered statistically significant (1,000 meters). The red lines represent the interval in which 90% of all data points are located.

In the case of the histogram obtained from two-dimensional imagery, rather than directly using the aforementioned 90% boundaries calculated from the data, we use polynomial functions fitted to these boundaries so that the final interval limits are smoother and more representative of realistic DIM/ice thickness distributions. In the case of the lower 5% bound calculated for the histogram, a single 7^{th}-degree polynomial was sufficient to satisfactorily represent the true lower bound. The upper 5% bound, however, presents a sharp angle in the histogram data around a distance to ice margin of approximately 2,000 meters, which cannot be properly represent by a single polynomial of relatively low degree. Any single polynomial function of relatively low degree will maintain its general shape in the low-DIM areas, and is therefore not able to properly fit the sharp angle in the histogram. This sharp angle around the DIM measurement of 2,000 meters is noticeable in Fig 3.3. For this reason, for the upper bound we fit a piecewise polynomial function

composed of a 7th-degree polynomial for the high-DIM/high-thickness areas and a 3rd-degree polynomial for the low-DIM/low-thickness region. The 7th-degree polynomial was found to match the overall shape of the lower 5% bound in its entirety and the upper 5% bound satisfactorily for most parts of the histogram. The 3rd-degree polynomial was then applied to correct for the distributions found for DIM measurements up to approximately 2,000 meters, which the 7th-degree polynomial did not satisfactorily represent. Fig. 3.5 demonstrates the polynomial-fitted interval boundaries discovered for the histogram generated from the nadir-looking 2D training dataset, and Fig. 3.6 demonstrates this histogram with the same axes limits as Fig. 3.3. In Fig. 3.6, the piecewise polynomial fitting of the upper bound can be seen in detail.

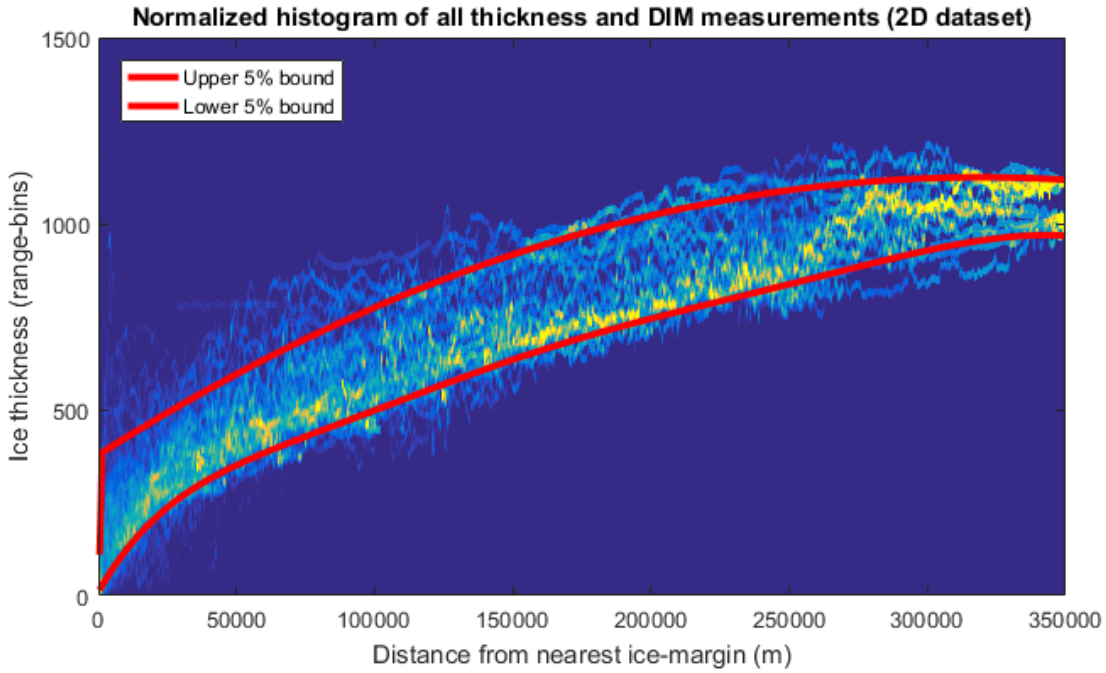

Fig. 3.5. Histogram of the ice-thickness measurements (vertical axis) found for all distances-to-ice-margin (horizontal axis) from the ground-truth two-dimensional dataset, displayed up to the maximum distance to ice margin considered statistically significant (350,000 meters). The red lines represent the interval in which 90% of all data points are located.

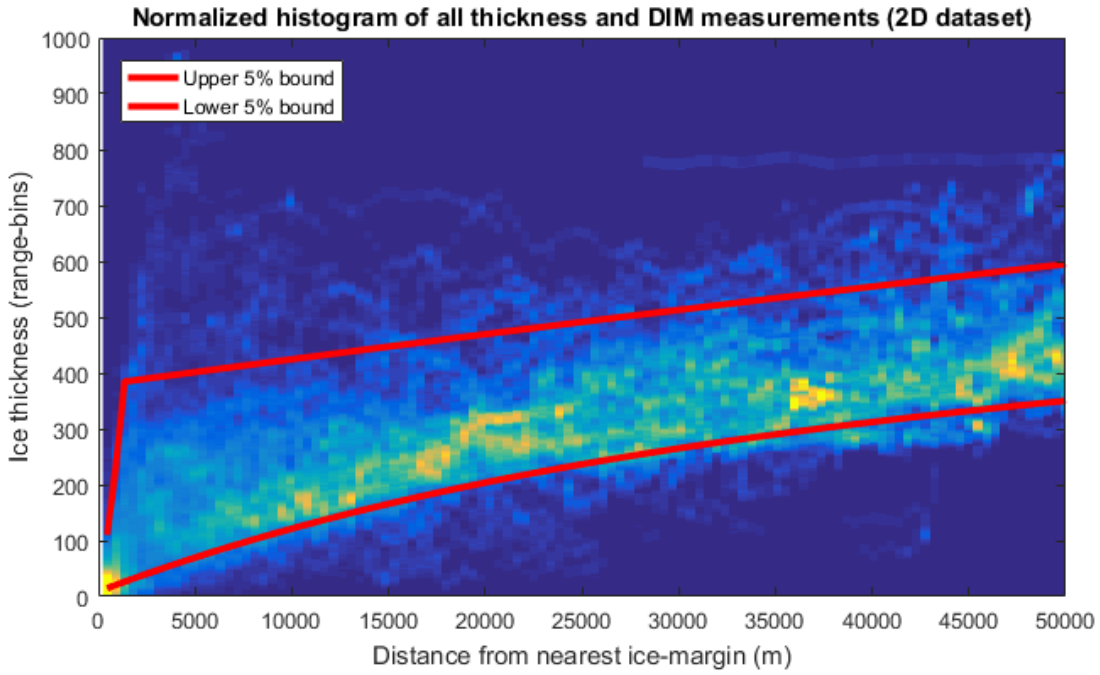

Fig. 3.6. Histogram presented in Fig. 3.5, with the low-DIM/low-thickness data points in detail. The 3rd-degree polynomial and the 7th-degree polynomial fitted to the upper 5% bound merge at a DIM of approximately 2,000 meters.

In both 2D and 3D calculations of the ψ_{DIM} term, we assign to the high probability area (i.e. the data points between the upper and lower bounds) of each distance to ice margin a uniform probability value equal to the mean of the data inside the high probability area. To parametrically model the histogram in areas outside the upper and lower bounds, we fit two exponential functions to each distance to ice margin, one for the data above the upper bound and one for the data below the lower bound for each distance to ice margin. An illustration of this method is provided in Fig. 3.7, in which the parametric model for a distance to ice margin equal to 800 meters in the histogram acquired from three-dimensional data is shown. These parametric histogram models are calculated in the same manner for both histograms.

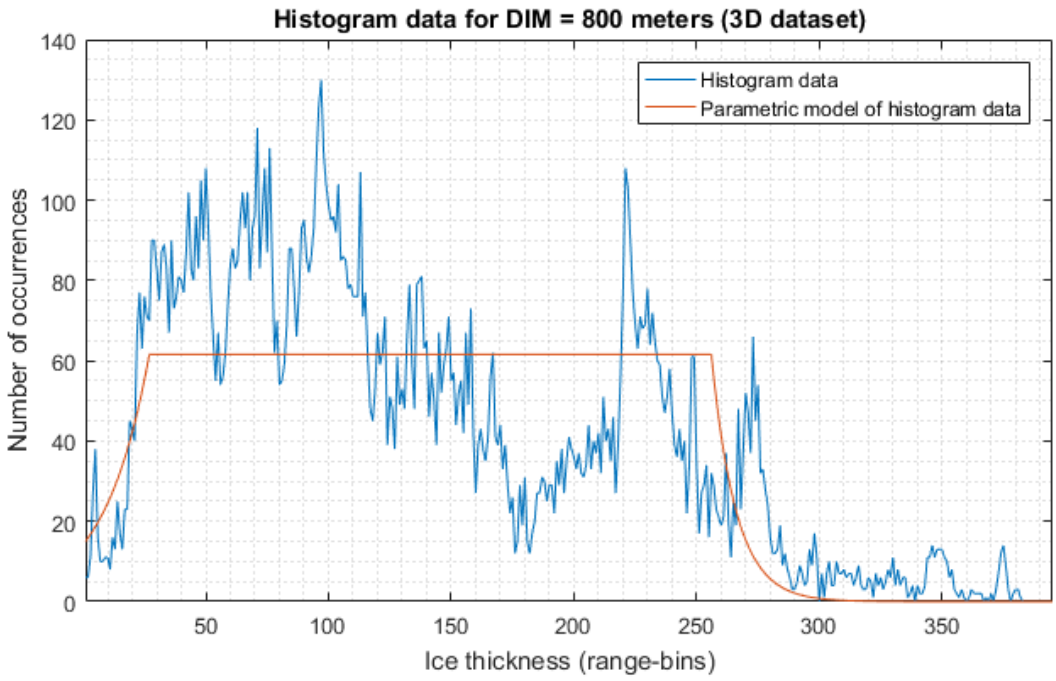

Fig. 3.7. The histogram data for all points with distance to ice margin equal to 800 meters in the three-dimensional training set is represented by the blue line. The parametric model fitted for these data, represented by the orange line, is composed of a fitted exponential function for points under the lower bound (ice thickness from 1 to 27 range-bins), a uniform distribution across the high probability area (ice thickness from 28 to 255 range-bins), and a fitted exponential function for points over the upper bound (ice thickness from 256 to 395 range-bins).

In Fig. 3.8, we present the mean and standard deviation (STDEV) of the data above the upper bound and below the upper bound for each distance to ice margin in the histogram acquired from the three-dimensional dataset. Note that there is a strong correlation between the values of the mean and standard deviation found for the data. This is an indication that the exponential function that was fitted to each distance to ice margin was a relatively good approximation of the data, since the mean and standard deviation of an exponential distribution are the same. In other words, if the data were distributed according to an exponential distribution, we would find these vectors to have precisely the same values on average. In Fig. 3.9, we demonstrate the same results for the histogram obtained from the two-dimensional layer data.

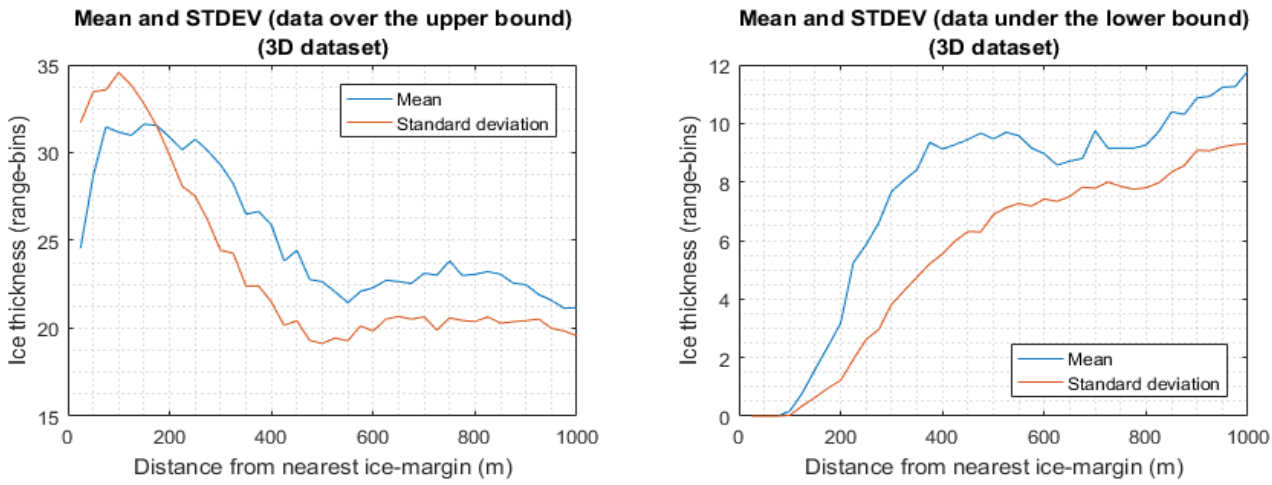

Fig. 3.8. Mean and standard deviation of the data over the upper bound (plot on the left) and under the lower bound (plot on the right) from the three-dimensional layer data.

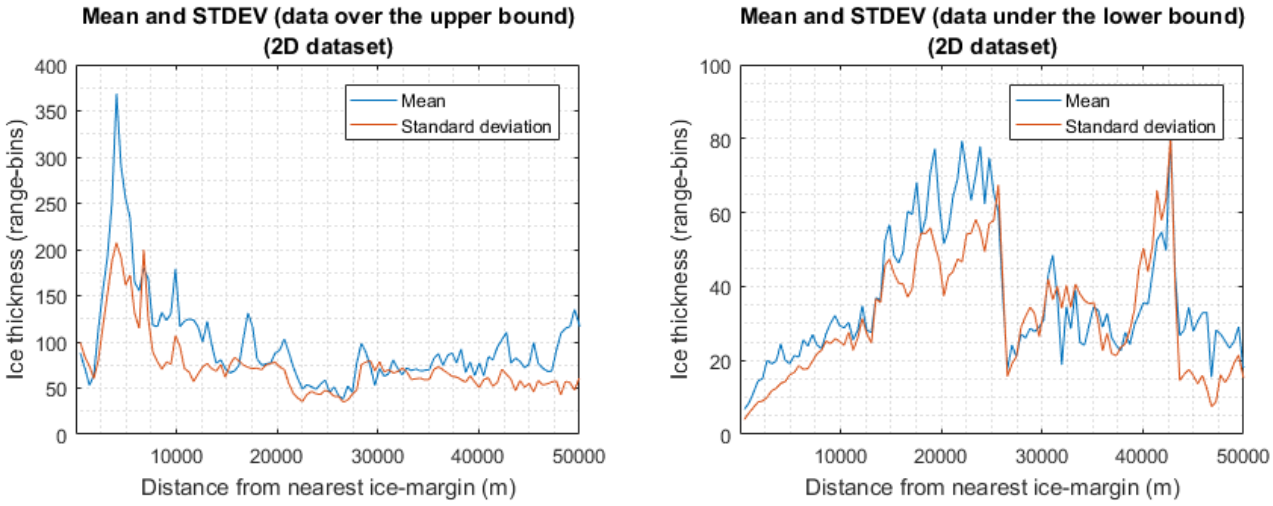

Fig. 3.9. Mean and standard deviation of the data over the upper bound (plot on the left) and under the lower bound (plot on the right) from the two-dimensional layer data. For ease of visualization, only the data at distance to ice margin up to 50 km is presented, but the correlation between the mean and standard deviation values was noted across all distances to ice margin.

Finally, we calculate the negative natural logarithm of the parametrized distribution of each DIM to obtain the final cost values for each combination of ice thickness and distance to ice margin. The high probability area at all DIMs is fixed to zero cost, and the points outside of this area are normalized proportionally. The resulting cost matrix obtained from the three-dimensional training dataset, before any scaling factor is applied, can be seen in Fig. 3.10. For comparison, the negative natural logarithm of the histogram found from the three-dimensional dataset is presented in Fig. 3.11. To allow this ψ_{DIM} unary cost term to properly match the other terms of the unary cost function, the values presented are multiplied by a scaling factor equal to 200 based on some earlier tuning of the cost function. The final cost matrix and the negative natural logarithm of the histogram found from the two-dimensional dataset are presented in Fig. 3.12 and 3.13, respectively. Figs. 3.14 and 3.15 demonstrate the same, using the axes limits as Fig. 3.3. The calculated cost function and the negative logarithm of the histogram are more similar in the three-dimensional case than the 2D case, as the real data distributions outside of the high probability are closer to exponential distributions.

In our proposed solution, any valid pixels with ice thickness or distance to ice margin greater than the maximum values defined in the matrices below are assigned the values calculated for the nearest defined ice thickness or DIM.

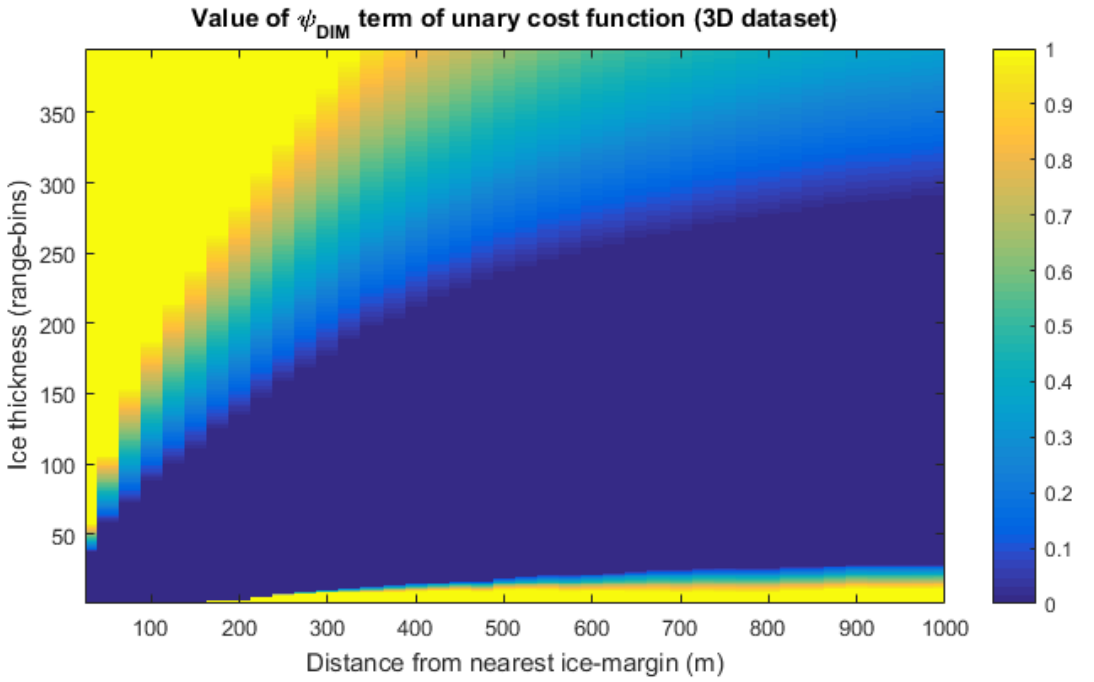

Fig. 3.10. Final normalized values of the proposed ψ_{DIM} term of the unary cost function applied to three-dimensional data tracking. The values shown above are multiplied by a scaling factor equal to 200.

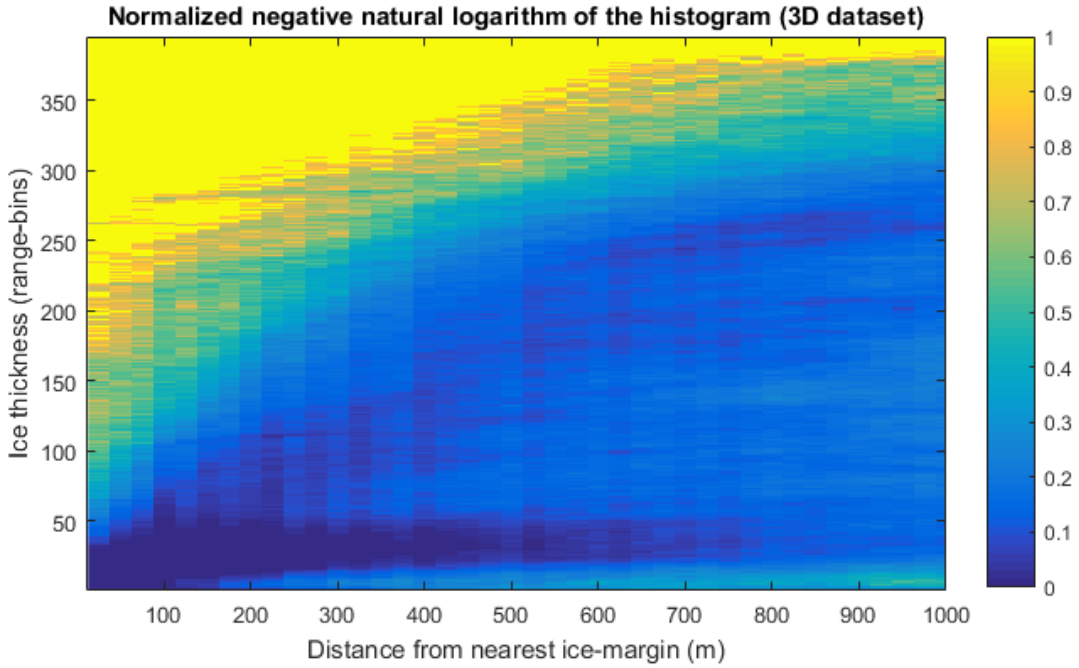

Fig. 3.11. Negative natural logarithm of the histogram presented in Fig. 3.4. Note the similarities between this result and our proposed final cost values shown in Fig. 3.10.

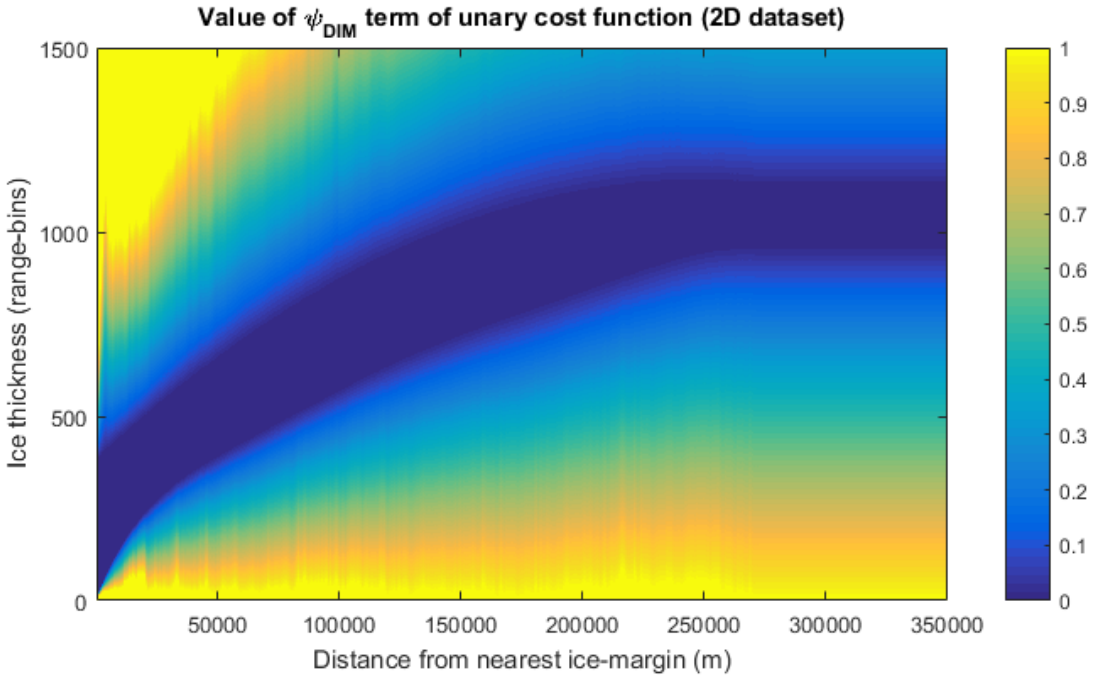

Fig. 3.12. Final normalized values of the proposed ψ_{DIM} term of the unary cost function applied to two-dimensional data tracking. The values shown above are multiplied by a scaling factor equal to 200.

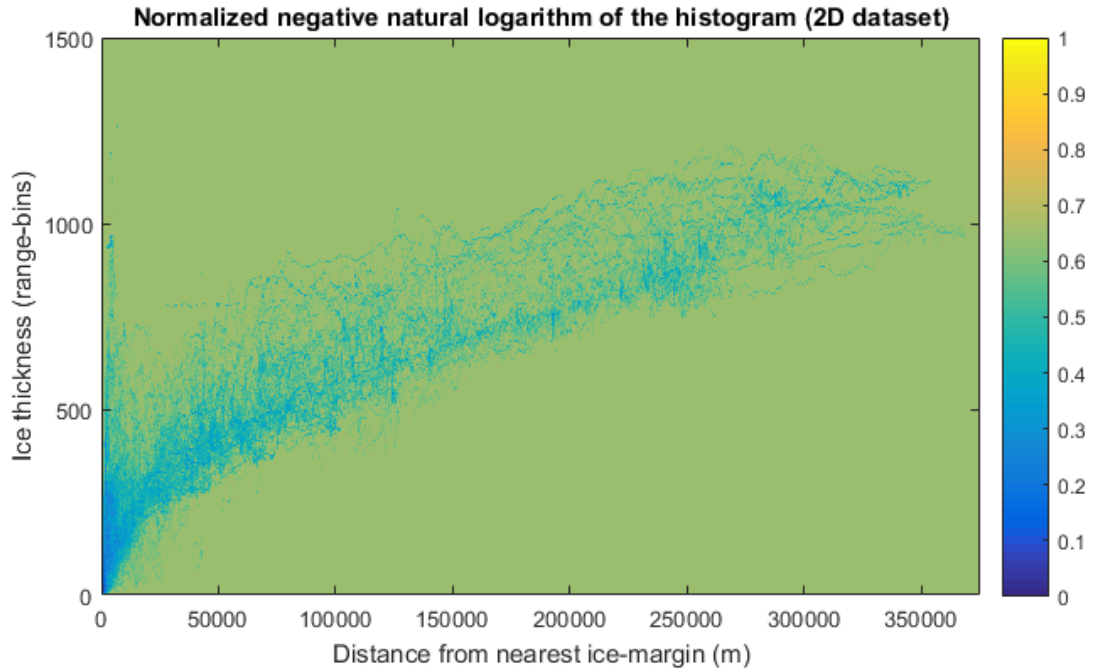

Fig. 3.13. Negative natural logarithm of the histogram presented in Fig. 3.5. Note the similarities between this result and our proposed final cost values shown in Fig. 3.12.

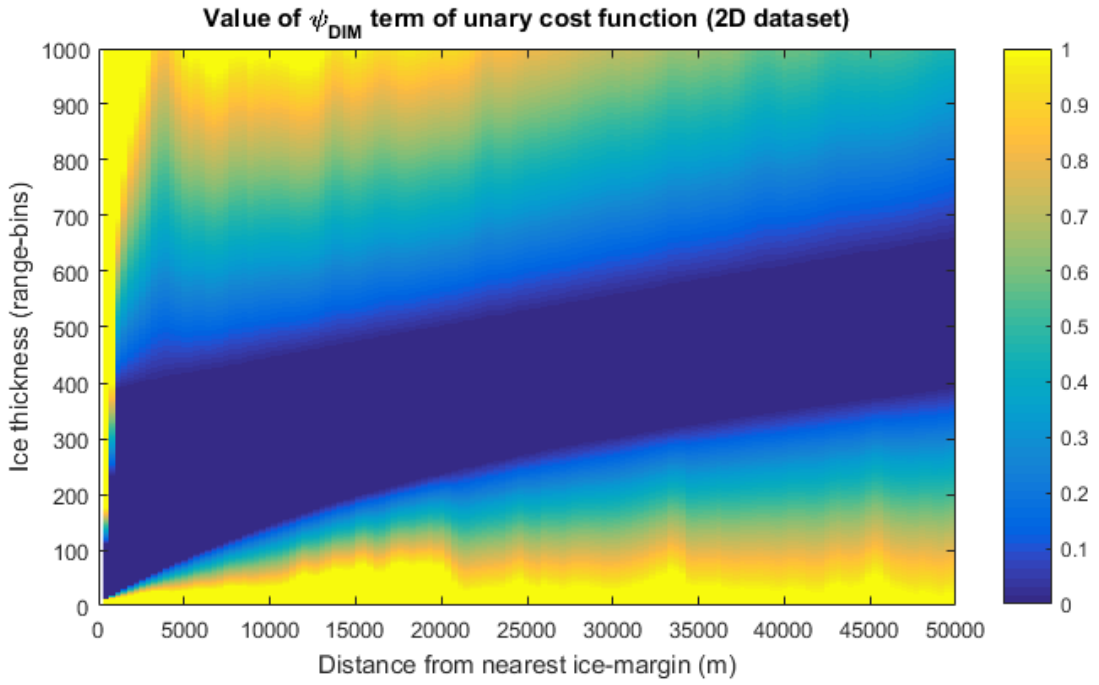

Fig. 3.14. Final normalized values of the proposed ψ_{DIM} term of the unary cost function applied to two-dimensional data tracking. The values shown above are multiplied by a scaling factor equal to 200.

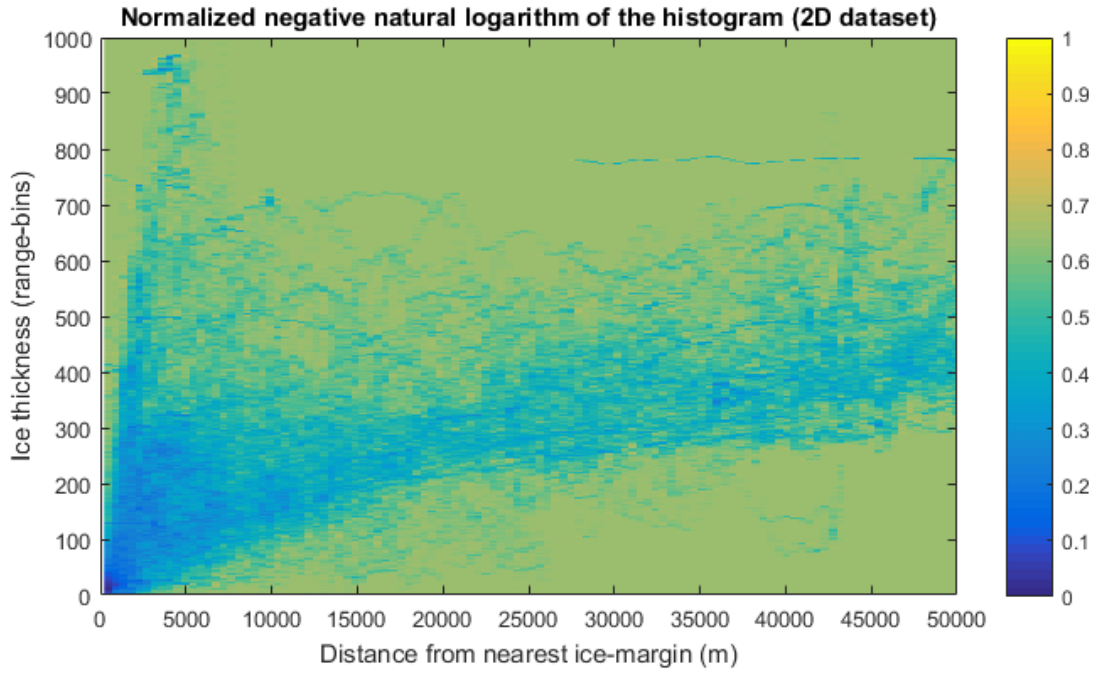

Fig. 3.15. Negative natural logarithm of the histogram presented in Fig. 3.5. Note the similarities between this result and our proposed final cost values shown in Fig. 3.12.

3.3 - Range-bin variation in the along-track dimension

The second cost term constructed from the results of the geostatistical analysis defines an along-track (range-line to range-line) transition model. Note that transitions between adjacent range-lines occur between columns in two-dimensional imagery, and between slices in the three-dimensional data products. As mentioned in Section 3.1, this information is acquired from the 2D and 3D ground-truth ice-bottom labels that compose the training sets described in Chapter 4. The distributions discovered from the ground-truth three-dimensional data are applied to 3D tracking using the MRF/TRW-S framework, and in this case all direction-of-arrival distributions are used. The model obtained from the geostatistical analysis performed on the ground-truth 2D layer data is applied to two-dimensional tracking using the HMM/Viterbi algorithm framework. However, no along-track transition model is applied when the Viterbi algorithm is tasked with tracking of three-dimensional imagery due to the lack of slice-to-slice cost propagation in the HMM framework since each slice is modeled and processed independently of the other slices.

This transition model was implemented as part of the binary cost function of both the HMM and MRF models. In both, this cost term serves as an empirical substitute for the previously suggested fixed smoothness constraint presented in Section 2.5, used to increase the cost of pixels that generate a large vertical discontinuity between two consecutive range-lines. While it still penalizes transitions that generate increased vertical discontinuities between range-lines, it applies the variance values discovered for each DoA bin rather than using a fixed value for all directions-of-arrival bins. Furthermore, the variance values are calculated directly from ground-truth data rather than being estimated or tuned.

The geostatistical analysis performed to discover this term consisted of calculating the variance of the vertical differences, measured in range-bins, between consecutive range-lines. In

the two-dimensional case, a single variance value $\sigma^2_{\mathrm{AT}_{2D}}$ corresponding to the variance at the nadir direction was calculated. In the three-dimensional case, a variance value was calculated for every direction-of-arrival bin. In other words, in the 3D case for each DoA bin we calculate the variance of the difference between the row-indices of the ice-bottom labels at all "source" and "destination" range-lines.

We assume a zero mean for all distributions used (i.e., all transition probabilities are centered at zero vertical difference, therefore the lowest-cost transition still always generates a perfectly flat layer), and only the variance between transitions is calculated. As an example, the histogram of all vertical differences found for DoA bin 30 (out of 64) can be seen in Fig. 3.16. Note that this distribution presents an approximately Gaussian shape; this can be seen in Fig. 3.17 which depicts the best-fit unimodal Gaussian distribution for the presented histogram. This approximate Gaussian pattern was similarly noted in the distributions of all DoA bins.

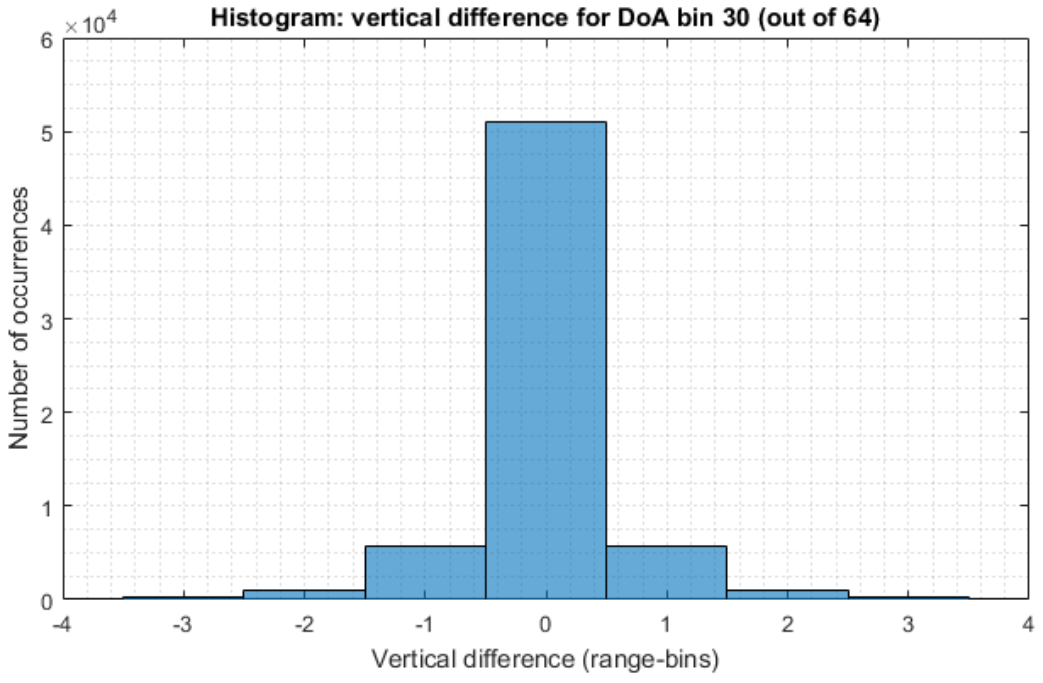

Fig. 3.16. Histogram depicting the distribution of along-track vertical differences, measured in range-bins, at direction-of-arrival bin 30 (out of 64).

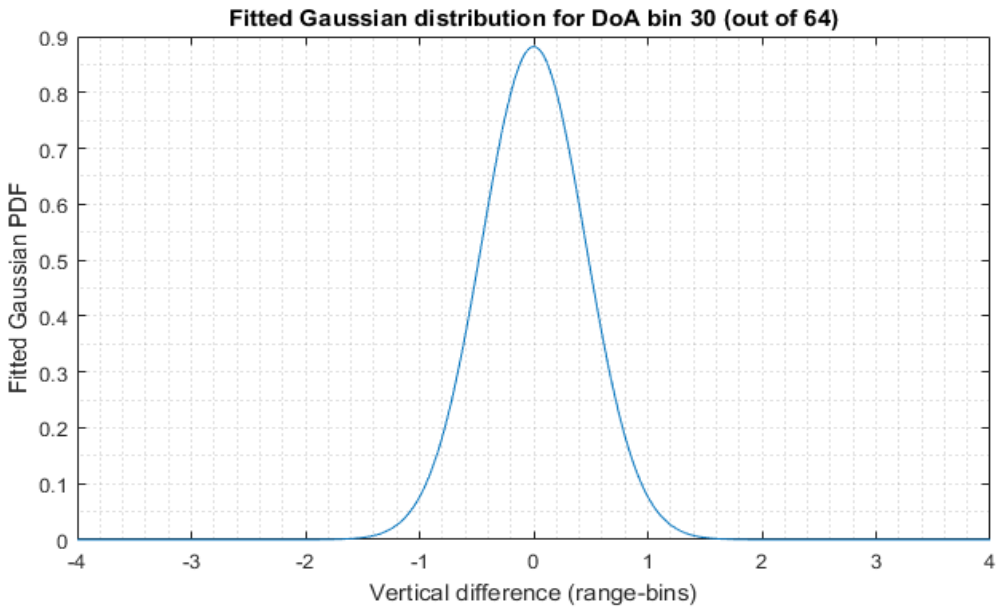

Fig. 3.17. The normalized best-fit Gaussian PDF relative to the histogram shown in Fig. 3.6.

For this reason, the proposed binary term for this transition model assigns costs based on an inverted Gaussian distribution with mean equal to the range-bin value that produces zero vertical difference and variance equal to the empirically-discovered value for the direction-of-arrival bin of interest. The inverse Gaussian distribution generates the desired effect that a lower binary cost is assigned to transitions with higher probability according to the analyzed data. Similarly to what was done in the original implementation of the binary cost function, shown in Eq. 7, this zero-mean Gaussian probability distribution is transformed into a quadratic function by a negative logarithm, which increases computational efficiency and reduces the occurrence of rounding errors.

The along-track binary cost $\psi_{B-AT-2D}(c_i, c_j)$ for two-dimensional tracking is mathematically defined as follows:

$$\psi_{B-AT-2D}(c_i, c_j) = -\log(\mathcal{N}(\Delta_Y, s_{c_i}, \sigma^2_{AT_{2D}})) \quad (16)$$

and the along-track binary cost $\psi_{B-AT-3D}(c_i, c_j)$ for three-dimensional tracking is defined as

$$\psi_{B-AT}(c_i, c_j) = -\log(\mathcal{N}(\Delta_Y, s_{c_i}, \sigma^2_{AT_{DoA}})) \tag{17}$$

where $\sigma^2_{AT_{DoA}}$ is the variance empirically discovered for the DoA bin of interest. Furthermore, Δ_Y is the vertical difference between the "source" and "destination" range-lines of interest, corrected to account for the range-slope of the ice-surface. This term is thus defined as

$$\Delta_Y = s_{c_i} - s_{c_j} - \left(s_{SURF}(c_i) - s_{SURF}(c_j)\right). \tag{18}$$

In both equations above, s_{c_i} and s_{c_j} are the row-indices assigned to adjacent range-lines, with s_{c_i} being the row-index of the "source" range-line and s_{c_j} the row-index of the "destination" range-line. Furthermore, $s_{SURF}(c)$ is the row-index of the known ice-surface layer at column c, and thus $s_{SURF}(c_i)$ refers to the row-index of the ice-surface layer at the "source" range-line, and $s_{SURF}(c_j)$ refers to the same at the "destination" range-line. The Gaussian PDF calculation of $\mathcal{N}(x, \mu, \sigma^2)$ is defined in Eq. 9. Note that the usage of s_{c_i} as the mean (μ) of the Gaussian distribution above generates the aforementioned effect of centering the PDF at the range-bin which generates zero vertical difference between the ice-bottom layer at the "source" and "destination" range-lines.

In Fig. 3.18, we demonstrate the final along-track cost distribution calculated as a function of the vertical difference between the ice-bottom layer point tracked at a given "source" range-line and the relevant pixels at the consecutive "destination" range-line for DoA bin 30, which is also presented in Figs. 3.16 and 3.17. These values were found by applying the value of $\sigma^2_{AT_{30}} = 0.4524$ (obtained from geostatistical analysis) onto Eq. 17. Furthermore, in Fig. 3.19 we demonstrate the variances found for every direction-of-arrival bin. These variances were smoothed by a moving-average filter to generate a more representative distribution despite our limited training dataset.

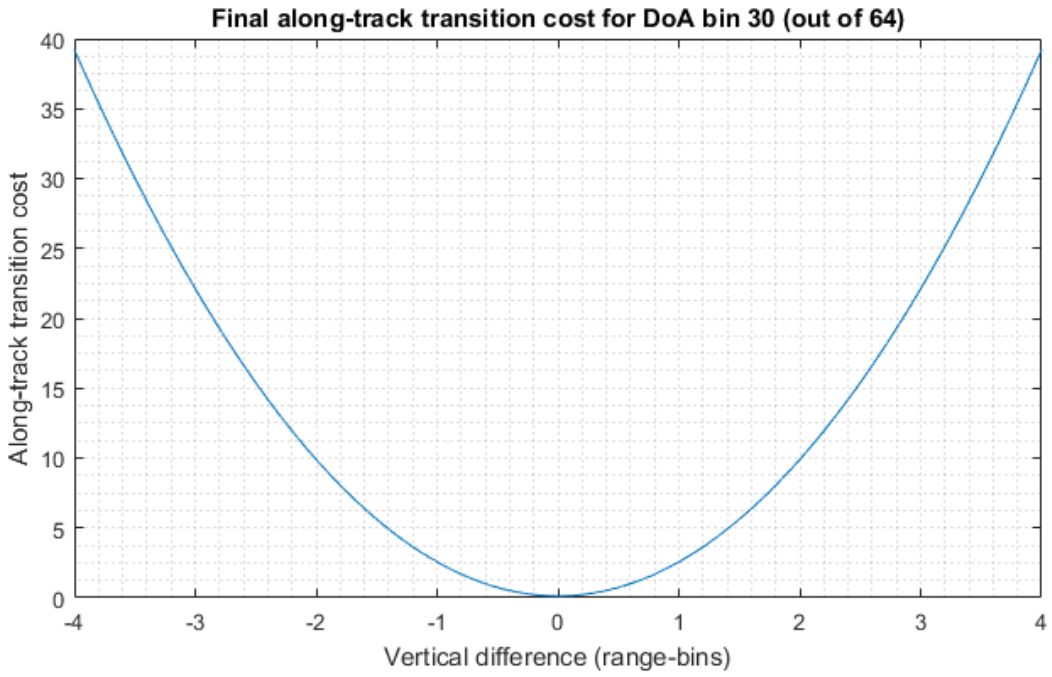

Fig. 3.18. Final along-track transition costs for DoA bin 30 (out of 64).

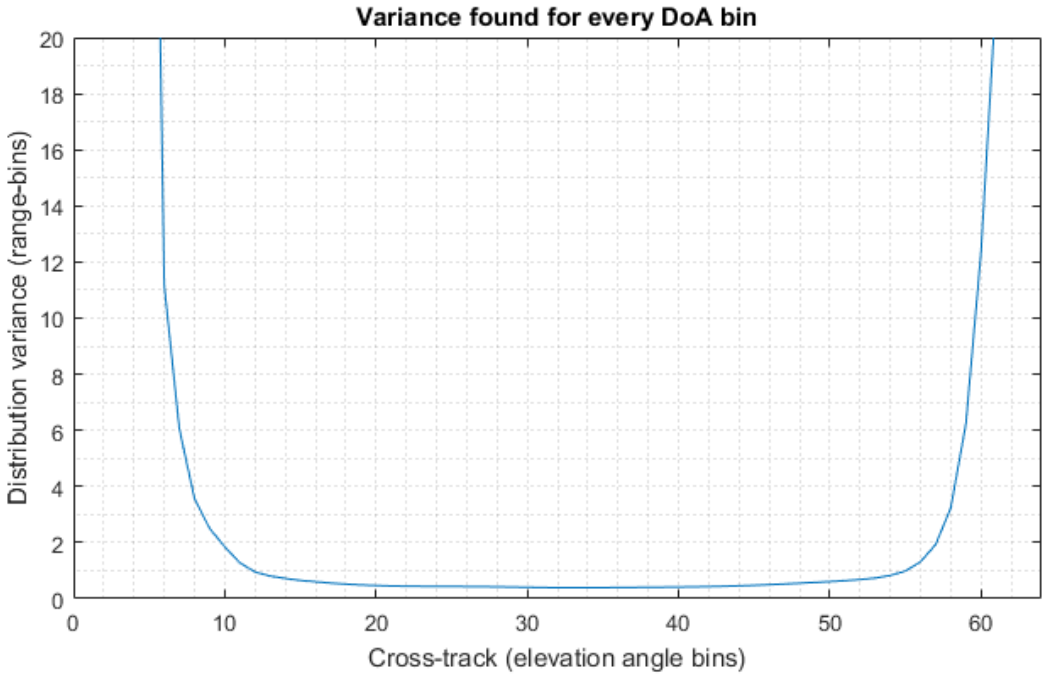

Fig. 3.19. Values of the variances, in range-bins, found for the along-track transitions for every DoA bin.

3.4 - Range-bin variation in the cross-track dimension

The final cost term derived from the geostatistical analysis serves as a cross-track (DoA-bin to DoA-bin) transition model, and is associated with the binary cost function of both HMM and MRF models. It refers to the vertical variation, measured in range-bins, of the ice-bottom layer between two consecutive direction-of-arrival bins in slices of three-dimensional imagery. Therefore, it is related to the same measurement used in the along-track transition model but calculated over the elevation angle (cross-track) dimension. Because this term operates in the cross-track dimension, it is only applicable in the case of three-dimensional imagery. It is employed by the MRF/TRW-S algorithm in all cases, and by the HMM/Viterbi algorithm when it is tasked with tracking individual slices of 3D datasets.

As is done with the along-track transition model, this cost term is implemented as part of the binary cost function used by both the HMM and MRF models, and acts as a smoothness constraint by penalizing large vertical discontinuities in the ice-bottom layer. The values used with this technique were obtained from geostatistical analysis performed on the same 3D training dataset as mentioned in Section 3.1. The geostatistical analysis consisted of calculating the mean and variance of the vertical differences between the ice-bottom layer locations in consecutive direction-of-arrival bins. We generate a histogram of all the vertical differences found for a given DoA bin over all slices of 3D imagery, calculate the mean variance of those vertical differences, and use the best-fitting unimodal Gaussian distribution to model the discovered histogram.

As an example, see Fig. 3.20 which displays the histogram of the vertical differences found in the transitions between source DoA bin 30 and destination DoA bin 31 (out of 64) across all slices of the 3D training dataset. Note that this histogram may be approximately modeled as a Gaussian distribution. This can be seen in Fig. 3.21, which depicts the best-fitting unimodal

Gaussian distribution for the presented histogram. This approximate Gaussian pattern was similarly noted in the distributions found for all DoA bins.

The final cross-track transition cost calculation is very similar to the one proposed for the along-track case, in which costs are assigned based on an inverted Gaussian distribution. The major difference between the two transition models is that the along-track model uses a zero-mean value, whereas the cross-track model uses the empirically-discovered mean and variance values for the direction-of-arrival bin of interest. The final cost distribution for the cross-track term is also transformed into a quadratic function by a negative logarithm.

The cross-track binary cost term $\psi_{B-CT}(c_i, c_j)$ is mathematically defined as follows:

$$\psi_{B-CT}(c_i, c_j) = -\log(\mathcal{N}(\Delta_Y, \mu_{CT_{DoA}}, \sigma^2_{CT_{DoA}})) \tag{19}$$

where $\mu_{CT_{DoA}}$ is the mean and $\sigma^2_{CT_{DoA}}$ is the variance empirically discovered for the DoA bin of interest, and all other variables have the same definition as in Eq. 17 and Eq. 18.

In Fig. 3.22, we demonstrate the final cross-track cost distribution calculated for transitions between DoA bins 30 and 31, found by applying the value of $\sigma^2_{CT_{30}} = 20.4725$ (obtained from geostatistical analysis) onto Eq. 18. Furthermore, in Fig. 3.23 we demonstrate the variances found for every direction-of-arrival bin. These variances were smoothed by a moving-average filter to generate a more representative distribution despite our limited training dataset.

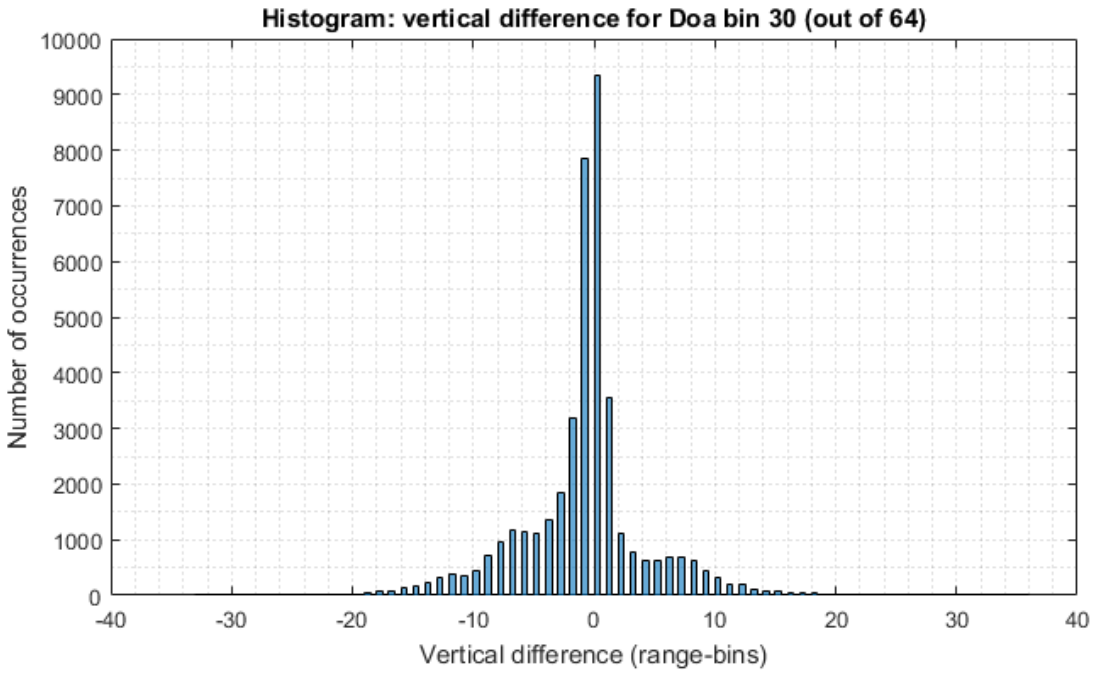

Fig. 3.20. Histogram depicting the distribution of cross-track vertical differences, measured in range-bins, at direction-of-arrival bin 30 (out of 64).

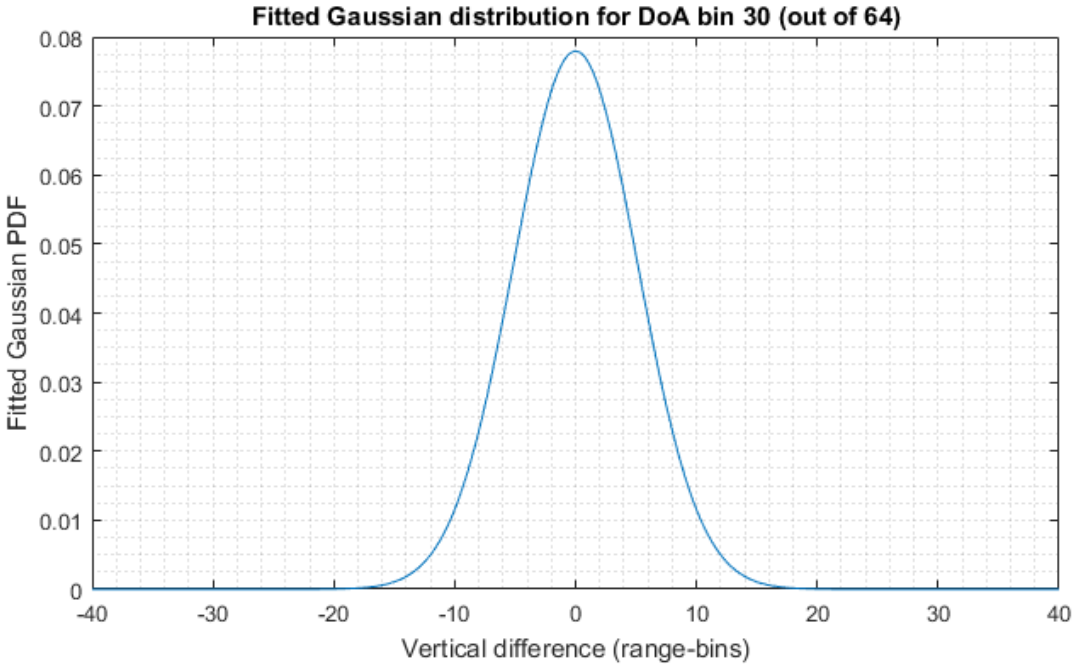

Fig. 3.21. The normalized best-fit Gaussian PDF relative to the histogram shown in Fig. 3.10.

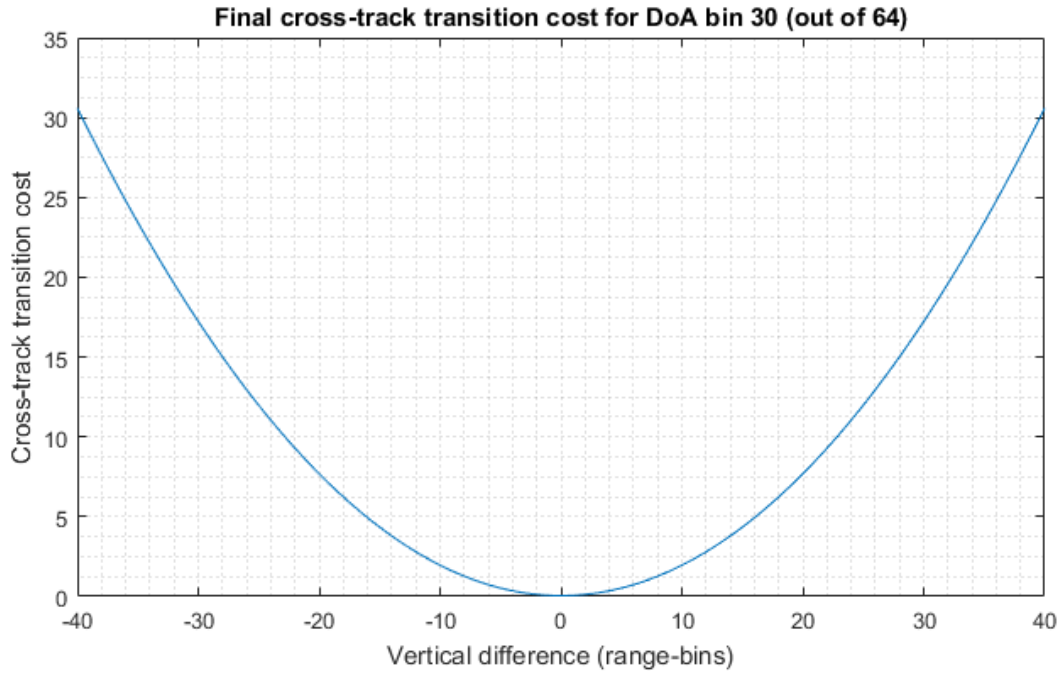

Fig. 3.22. Final cross-track transition costs for DoA bin 30 (out of 64).

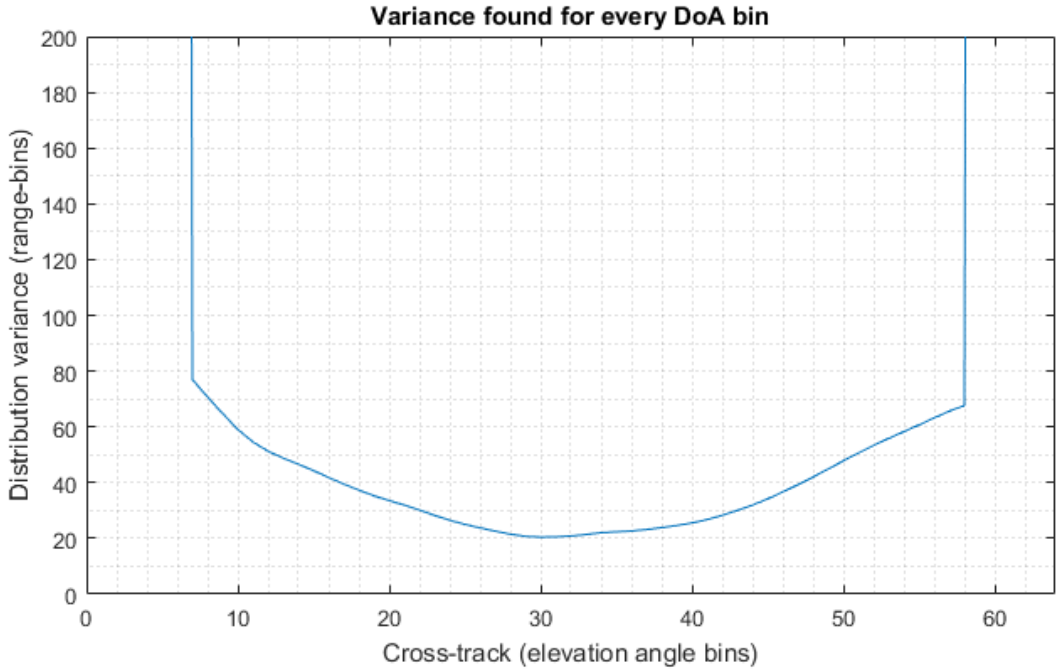

Fig. 3.23. Values of the variances, in range bins, found for cross-track transitions for every DoA bin.

Chapter 4 - Results and Comparison Between State-of-the-Art Solutions

4.1 - Introduction

The purpose of this chapter is to define the settings in which the tracking frameworks presented in this work were executed, and to present the results achieved by these methods in both 2D and 3D tracking scenarios. We describe the training and testing datasets used, explain the result criteria, and evaluate the outputs obtained from each of the trackers. In doing so, we present a comparison between our proposed tracking techniques and several other recently published methods for ice-layer tracking applied to CReSIS imagery.

Previously published works applied their proposed tracking solutions on different radar imagery datasets, measured the accuracy of their techniques using different benchmarks, and offered results using different units and calculations. This has complicated a general evaluation of the state-of-the-art in this area. For this reason, we offer a comprehensive and standardized comparison between several recently proposed automated tracking methods. An effort has been made to apply the software provided by the authors of these solutions exactly as intended, and to perform a uniform and impartial assessment of the results by applying the solutions to the same 2D and 3D CReSIS datasets under the same conditions. All of the tracking algorithms included in this general comparison are mentioned in Section 1.4 of this document, and are briefly reviewed in Section 4.2.

The remainder of this chapter is organized as follows: Section 4.2 presents a review of the tracking solutions for two-dimensional imagery included in this comparison, along with the settings in which these methods were applied. Section 4.3 does the same for the tracking solutions for three-dimensional imagery. In Sections 4.4 and 4.5, we present the datasets used in the 2D and

3D comparisons, respectively, and in Section 4.6 we define the error calculations used for all the trackers included. In Section 4.7 we present and evaluate the results obtained for the 2D tracking methods, and in Section 4.8 we do the same for three-dimensional tracking results.

4.2 - Two-dimensional tracking solutions included in the comparison

4.2.1 - Level-sets approach by Rahnemoonfar *et al.*

The first solution considered in this comparison is the distance-regularized level-sets model (LSM) proposed by Rahnemoonfar *et al.* [22]. The existing implementation of this method is applicable to tracking of nadir-looking 2D echograms only, is able to simultaneously solve for both ice-surface and ice-bottom layers, and requires no additional sources of evidence regarding the ice sheets. Furthermore, it does not allow for the addition of manual or automated ground-truth data points. It presents three tunable hyperparameters which significantly modify the final results:

1. A scalar variable y which defines the initial location of the ice-surface layer (which is then iteratively modified and evolved by the algorithm). The value of y defines the range-bin in which the ice-surface contour is initialized for all range-lines. The ice-surface contour is therefore initialized as a flat horizontal layer in the 2D echogram.
2. A scalar Δy which defines the initial vertical offset between the ice-surface and the ice-bottom layers (the initial ice-bottom contour therefore being defined by $y + \Delta y$). The ice-bottom contour is therefore also initialized as a flat horizontal layer in the 2D echogram.
3. The number of iterations performed by the algorithm. Note that the level-sets technique is not necessarily expected to output improved results as the number of iterations performed is increased.

We tuned the hyperparameters using an exhaustive grid-search over the parameter ranges that we expect to contain the optimal results. The purpose is to find the optimal combination for the values of the three variables listed above. This tuning routine is performed on the 2D training set described in Section 4.4, and the results presented here are obtained by applying the algorithm on the testing set using the optimal combination found from the training set.

The tested values for the aforementioned ice-surface initialization parameter are $y = \{160, 180, 200, \dots, 300\}$. Since we would like to see how the algorithm performs using the a priori ice-surface information, we also test the mean value of the range-bin labels of the known ice-surface layer of each data frame as an alternative initial position of the ice-surface contour of each frame, calculated as

$$\mathrm{mean(S)} = \frac{1}{N_{RL}} * \sum_{i=1}^{N_{RL}} s_{SURF}(i), \tag{19}$$

where $s_{SURF}(i)$ is the range-bin of the known ice-surface layer at range-line i, and N_{RL} is the number of range-lines in the 2D data frame of interest. The final values used for this variable were therefore $y = \{160, 180, 200, \dots, 300, \ \mathrm{mean(S)}\}$. The values used for the ice-bottom offset Δy were $\Delta y = \{5, 10, 20, 40\}$. For the number of iterations performed by the algorithm, we tested values of $\{50, 100, \dots, 800\}$. In total, $9 * 4 * 16 = 576$ combinations were evaluated. When using a fixed y value, we found that the combination of $y = 200$, $\Delta y = 5$, and 400 iterations produces the most favorable mean error results. When the mean range-bin of the ice-surface layer is used as the y value, the best results were achieved when combining $y = \mathrm{mean(S)}$ with a value of $\Delta y = 40$ and a total of 350 iterations. The optimal parameters in both choices for y are taken to be the parameters that minimize the absolute error of the ice surface and the ice bottom. Fig. 4.1 depicts the two-dimensional error plot generated by each of the combinations tested for $y = \mathrm{mean(S)}$.

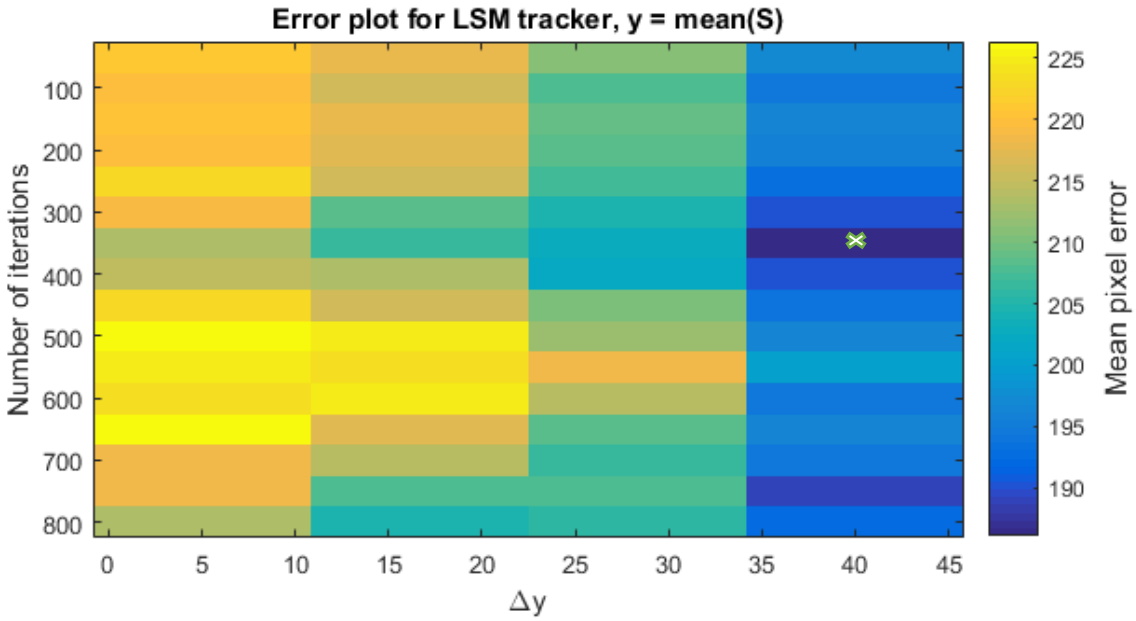

Fig. 4.1. Mean pixel error found for each tested combination of the number of iterations and Δy parameters. The white 'x' marks the combination with the lowest mean error, equal to a number of iterations of 350 and $\Delta y = 40$. The same procedure was performed for the case in which y, Δy, and the number of iterations are unknowns.

We evaluate the results of the LSM tracker using both definitions of the y parameter and the respective parameter combinations described above. This tracking technique is assigned the index of (1) in the two-dimensional comparison when the y parameter is equal to the fixed value of 160 for all data frames, found by the hyperparameter tuning routine to be the optimal value. Furthermore, this tracking technique is assigned the index of (2) in the two-dimensional comparison when the y parameter is assigned a variable value for each frame according to the calculated mean range-bin of the ice-surface layer.

4.2.2 - HMM + Viterbi algorithm approach by Crandall *et al.*

The second technique included in this comparison is the original probabilistic graphical approach proposed by Crandall *et al.* [14] which constructs an HMM to model the ice sheets and uses the Viterbi algorithm to perform inference on the HMM. The improved HMM/Viterbi

algorithm framework presented in Chapters 2 and 3 of this document is based on this work. This method is applicable to tracking of nadir-looking 2D echograms only, is able to simultaneously solve for both ice-surface and ice-bottom layers, and requires no additional sources of evidence regarding the ice sheets. This technique allows for the addition of ground-truth coordinates for both ice interfaces.

We have tested this technique in two different ways. First, no ground-truth points of any kind were provided to the algorithm, and the outputs for both ice layers are included in the comparison. Second, we provide the model with ground-truth points for the ice-bottom from all crossovers (previous data collections at the same location as data points in the data frame being tracked), and provide the a priori ice-surface in the format of ground-truth points for the ice-surface. In this second case, only the output for the ice-bottom is included in the comparison. No parameter tuning was performed for this method, and so only the parameters proposed by the authors were applied. The results presented for this technique were acquired from execution on the 2D testing set only. This tracking technique is assigned the index of (3) in the two-dimensional comparison when no ground-truth points are provided to the algorithm, and the index of (4) when the ice-surface location and the available ice-bottom ground-truth points are provided to the model.

4.2.3 - MRF + Markov Chain Monte Carlo approach by Lee *et al.*

The third approach included in this comparison, proposed by Lee *et al.* [24], builds on the probabilistic graphical approach of [14] but constructs an MRF to represent the model. In order to solve for the maximum-likelihood solution, this approach attempts to estimate functionals of the full joint probability distributions via Gibbs sampling. The Gibbs sampler is a Markov Chain Monte Carlo technique, and so this technique is referred to as “MCMC” in this document.

The proposed implementation of this method is applicable to tracking of nadir-looking 2D echograms only, is able to simultaneously solve for both ice-surface and ice-bottom layers, and requires no additional sources of evidence regarding the ice sheets. Furthermore, it does not allow for the addition of manual or automated ground-truth data points. No parameter tuning was performed for this method, and so only the parameters proposed by the authors were applied. The results presented for this technique were acquired from execution on the 2D testing set only. This tracking technique is assigned the index of (5) in the two-dimensional comparison.

4.2.4 - Modified HMM + Viterbi algorithm approach

The final solution included in the 2D comparison is the modified HMM + Viterbi algorithm technique proposed in Chapters 2 and 3. This approach is applicable to both nadir-looking 2D echograms and individual slices of 3D imagery, and therefore it was tested in both cases. This method detects the ice-bottom layer only, and requires several additional sources of evidence regarding the ice sheets, such as the location of the ice-surface and the ice-mask of the surveyed terrain, as mentioned in Chapter 2. All available ice-bottom crossovers were provided to this algorithm in the form of high-confidence ground-truth points. No low-confidence ground-truth points were provided. Furthermore, this method is the only one in the 2D comparison to benefit from the empirically-derived model definitions discovered by the geostatistical analysis described in Chapter 3 of this document. The results presented in the 2D comparison for this technique were acquired from execution on the 2D testing set only. This tracking technique is assigned the index of (6) in the two-dimensional comparison.

4.3 - Three-dimensional tracking solutions included in the comparison

4.3.1 - Modified HMM + Viterbi algorithm approach

The first solution included in the 3D comparison is the modified HMM + Viterbi algorithm technique proposed in this document. This is the same algorithm as discussed in Section 4.2.4. This method performs tracking on individual slices of 3D imagery. Results from previously tracked slices were not provided to the algorithm in any form. However, a ground-truth point in the nadir direction-of-arrival bin was provided to the algorithm in each slice. The results presented in the 3D comparison for this technique were acquired from execution on the 3D testing set only. This tracking technique is assigned the index of (1) in the three-dimensional comparison.

4.3.2 - MRF + TRW-S algorithm approach by Xu *et al.*

The second technique included, proposed by Xu *et al.* [17], is the original probabilistic graphical approach constructing an MRF to model the three-dimensional space defined by the 3D radar imagery and using the TRW-S algorithm to iteratively find the minimum-cost solution conditioned on the MRF. The modified MRF/TRW-S algorithm framework presented in Chapters 2 and 3 of this document is based on this algorithm. This method solves for the ice-bottom layer only, and may include additional evidence of the ice-sheets, such as ground-truth points and ice-mask datasets, if available. A ground-truth point in the nadir direction-of-arrival bin was provided to the algorithm in each slice, as well as the available ice-mask dataset. No parameter tuning was performed for this method, and so only the parameters proposed by the authors were applied. The results presented for this technique were acquired from execution on the 3D testing set only. This tracking technique is assigned the index of (2) in the three-dimensional comparison.

4.3.3 - C3D + RNN approach by Xu *et al.*

One of the two approaches included in this comparison that does not make use of a probabilistic graphical model (along with the level-sets solution described in Section 4.2.1), the recently proposed solution by Xu *et al.* [31] is notable for being one of the first efforts in the field of automated ice-layer tracking to employ deep-learning techniques. The proposed method employs a combination of three-dimensional convolutional neural networks (C3D) and recurrent neural networks (RNN) to find both the ice-surface and ice-bottom layers. The neural networks were trained using the manually-corrected 3D training set (described in Section 4.5) for both the ice-surface and ice-bottom layers, and tested on the 3D testing dataset. This method requires no additional sources of evidence and does not allow for the input of ground-truth points. This tracking technique is assigned the index of (3) in the three-dimensional comparison.

4.3.4 - Modified MRF + TRW-S algorithm approach

The final solution included in this comparison is the modified MRF + TRW-S algorithm technique proposed in Chapters 2 and 3. This approach detects the ice-bottom layer in three-dimensional imagery only, and requires the aforementioned additional sources of evidence regarding the ice sheets. A ground-truth point in the nadir direction-of-arrival bin was provided to the algorithm in each slice. Furthermore, this method is the only one in the 3D comparison to benefit from the empirically-derived model definitions achieved through the geostatistical analysis described in Chapter 3 of this document. The results presented for this technique were acquired

from execution on the 3D testing set only. This tracking technique is assigned the index of (4) in the three-dimensional comparison.

4.4 - Dataset used in the comparison between 2D tracking solutions

The two-dimensional dataset used by all the tracking solutions presented in this comparison was acquired by the MCoRDS instrument as part of the 2014 NASA Operation IceBridge deployment to Greenland and the Canadian Arctic Archipelago. Included in the comparison are a total of 57 data segments, out of which 23 segments (approximately 40%) were randomly selected as the training set, and the remaining 34 segments were used as the testing set. The 23 data segments in the training set are composed of a total of 716 data frames, and the 34 data segments used as the testing set are composed of a total of 1,146 data frames. The 2D data segments included in this comparison are listed in Table 4.1 along with their division between training and testing sets. The ground-truth ice-surface layer dataset for both training and testing sets was obtained by executing a simple threshold tracker guided by the ArcticDEM [19] digital elevation model dataset. The results obtained from the automated ice-surface tracking were occasionally manually corrected where necessary. The ground-truth ice-bottom layer dataset for both training and testing sets was acquired via manual layer tracking performed by experienced CReSIS analysts.

4.5 - Dataset used in the comparison between 3D tracking solutions

The three-dimensional dataset used in this work is a proper subset of the 2D dataset used and thus was acquired by the same instrument and in the same season. This dataset is far smaller, being composed of only 5 data segments which correspond to a total of 102 data frames. From the

102 data frames in the 3D dataset, 5 were discarded due to the altitude of the measurement causing the range gate to not include the ice bottom, and thus 97 data frames were considered in this comparison. Out of the 97 frames, 57 (approximately 60%) were used as the training set, with the 40 remaining data frames being used as the test set. The increased percentage of frames used as training data in the 3D case is due to the reduced amount of total data, and the fact that the tracking approach of [31], based on a neural network technique, heavily relies on large amounts of training data to perform well. The 5 data segments used in this comparison were all the currently available three-dimensional data for this season. The 3D data segments included in this comparison are listed in Table 4.2 along with their division between training and testing sets. Due to the processing parameters used in the generation of the imagery in this dataset, the elevation angle of all slices in both the training and testing datasets is divided into 64 DoA bins.

The ground-truth ice-surface layer dataset for both training and testing sets was acquired entirely from the existing ArcticDEM [19] dataset. These estimates were occasionally manually corrected where necessary. The ground-truth ice-bottom layer dataset for both training and testing sets was acquired by executing the original MRF/TRW-S tracking solution (described in Section 4.3.2) and then performing manual analysis and correction of the automated results. For this reason, it is likely that the 3D ground-truth layer data present a bias towards the results output by the automated TRW-S technique. The initial use of the TRW-S tracker to create the ground-truth data is necessary for the 3D dataset due to its large size. This is not the case for 2D layer data, as manual tracking of these images is a far more tractable problem.

4.6 - Definition of the error measurements

The main result obtained from the comparison of the automated tracking frameworks analyzed in this work is the difference between the automated output of a given tracking solution and the available manually-corrected ground-truth layer data for each data frame. These differences are calculated for each column as the absolute column-wise vertical distance between the ground-truth and the automatically tracked layers, from which the mean and median values over all data frames considered are presented. These error measurements are calculated in terms of pixels (or range-bins) and are found for every range-line in the case of nadir-looking 2D echograms, and for every range-line and DoA-bin in the case of three-dimensional imagery.

For both 2D and 3D tracking scenarios, we also present accuracy and precision calculations, using three different definitions of a true positive result: a particular image column is considered to have been correctly tracked if it is within 3 range-bins, within 5 range-bins, and within 10 range-bins of the ground-truth. A true negative and a false positive column are then any column for which this is not the case. Therefore, the accuracy and precision measurements are identical and equivalent to the percentage of correctly labeled columns (i.e. columns labeled within 3, 5, and 10 range-bins of the ground-truth).

The two-dimensional ice-bottom tracking results of mean error, median error, and accuracy are presented according to two different criteria. First, all range-lines of all data frames in the testing set are considered. However, this method occasionally generates misleadingly large mean and median tracking errors for the techniques that do not consider the ice-mask during the tracking. This is due to the fact that these techniques will attempt to detect the ice-bottom layer in all range-lines of each data frame, even in areas in which no ice-bottom is detectable because the terrain is not covered by ice. Several of the data frames included in the testing set used are partially or

entirely devoid of ice. The only 2D tracking technique which accounts for the ice-mask included in this comparison is the proposed modified HMM/Viterbi tracking framework. For this reason, the second method used for the ice-bottom results of all trackers in this comparison excludes the range-lines in which the ice-mask dataset indicates the inexistence of ice.

The results presented in the comparison between trackers of 3D imagery consider only DoA bins $[5, 6, 7, \ldots, 60]$ of the 64 total DoA bins in the dataset used. This was done because the data in the extreme elevation angles typically present very low quality, and so are ignored in the comparison.

We also perform a self-assessment of tracking error for all four of the 3D layer trackers included in this comparison. This is done by calculating the difference between the results output by each tracker in all areas of the testing set where flightlines crossed and two independent measurements were acquired over the same location. For comparison, we also provide the mean and median error measurement calculated in the same areas using the manually-corrected layer data.

Finally, we present the mean execution time measurement for all of the trackers presented in this comparison. These values are obtained for the tracking of a single data frame in both 2D and 3D tracking scenarios. Note that all tracking solutions presented here are trivially parallelizable, as tracking a given data frame is independent from other frames. The only exception to this is the modified HMM/Viterbi algorithm framework proposed here, which horizontally concatenates all frames of the same data segment as discussed in Section 2.5. However, this solution is still parallelizable since tracking of separate data segments is independent and thus they can still be processed simultaneously. Although execution time is less critical for fully automated analysis, it is very beneficial for trackers to present low-enough execution times to be used

interactively when manual ground truth points are being added. For instance, an ice layer analyst may desire to correct part of the results obtained from an initial run of a given automated tracking solution. In this case, the analyst may manually enter additional ground-truth points for the desired ice layer and re-execute the tracking algorithm for the region that needs to be corrected. Near real-time evaluation allows for the algorithm to be run as ground points are added so that the analyst knows immediately when enough ground points are added.

Two-dimensional data training and testing sets

2014_Greenland_P3			
Training Set		Testing Set	
Segment	Number of Frames	Segment	Number of Frames
20140313_07	1	20140313_05	1
20140313_08	3	20140313_06	1
20140313_10	2	20140313_09	5
20140325_07	5	20140314_04	3
20140331_02	13	20140325_05	2
20140401_04	5	20140325_06	1
20140405_01	69	20140331_01	13
20140409_02	48	20140401_01	8
20140410_01	70	20140401_03	48
20140412_01	9	20140407_01	72
20140412_02	33	20140408_01	68
20140412_04	8	20140409_01	24
20140415_04	6	20140412_03	5
20140415_05	19	20140412_05	1
20140416_01	68	20140414_02	69
20140421_01	78	20140415_01	6
20140501_01	46	20140415_02	29
20140506_01	46	20140415_03	2
20140507_01	70	20140416_02	1
20140508_01	73	20140419_02	11
20140520_04	4	20140419_03	42
20140520_05	3	20140423_02	36
20140521_01	37	20140424_01	73
		20140426_01	46
		20140429_01	69
		20140502_01	61
		20140505_01	70
		20140509_01	60
		20140512_01	61
		20140514_01	69
		20140515_02	71
		20140516_01	50
		20140520_03	31
		20140521_02	37
Total:	716	Total:	1,146

Table 4.1. List of the data segments from the 2014 Greenland P3 season used in the comparison between 2D tracking frameworks. The training and testing sets were randomly selected. For a given data segment shown above, all of the data frames that compose it were used.

Three-dimensional data training and testing sets

2014_Greenland_P3			
Training Set		Testing Set	
Segment	Frames	Segment	Frames
20140325_05	1	20140401_03	5:24
20140325_07	1:4	20140506_01	11:30
20140401_03	1:4, 25:47		
20140506_01	1:10, 31:45		
Total:	57	Total:	40

Table 4.2. List of the data segments from the 2014 Greenland P3 season (Canadian Arctic Archipelago) used in the comparison between 3D tracking frameworks. The training and testing sets were randomly selected. As shown above, data segments 20140401_03 and 20140506_01 were split between the training and testing sets. Notation such as "1:4" in the "Frames" column should be understood as frames 1, 2, 3, and 4.

4.7 - Results and evaluation of two-dimensional tracking techniques

TWO-DIMENSIONAL TRACKING RESULTS – ICE-SURFACE

	(1)	(2)	(3)	(5)
Ice-surface mean error	5.62	4.06	12.93	2.50
Ice-surface median error	0	0	0	0
Ice-surface accuracy (3 px)	94.31%	95.26%	92.47%	95.62%
Ice-surface accuracy (5 px)	94.71%	95.63%	92.49%	96.27%
Ice-surface accuracy (10 px)	95.37%	96.19%	92.59%	97.51%

Table 4.3. Ice-surface error and accuracy measurements obtained from the two-dimensional automated trackers. Only the 2D trackers designed to detect the ice-surface layer are presented. All mean and median errors are in pixels.

(1): Level-sets technique [22], fixed initial ice-surface contour value

(2): Level-sets technique [22], variable initial ice-surface contour value equal to the mean of the ice-surface layer

(3): Original HMM/Viterbi algorithm [14], no ground-truth provided

(5): MCMC [24]

TWO-DIMENSIONAL TRACKING RESULTS – ICE-BOTTOM, ALL RANGE-LINES

	(1)	(2)	(3)	(4)	(5)	(6)
Ice-bottom mean error	16.21	15.04	24.80	4.63	13.18	1.67
Ice-bottom median error	0	0	0	0	0	0
Ice-bottom accuracy (3 px)	93.64%	96.51%	92.71%	94.29%	96.02%	98.03%
Ice-bottom accuracy (5 px)	94.00%	97.05%	92.81%	95.31%	96.40%	98.34%
Ice-bottom accuracy (10 px)	94.40%	97.45%	93.09%	96.35%	97.51%	98.69%

Table 4.4. Ice-bottom error and accuracy measurements obtained from the two-dimensional automated trackers. These results include all range-lines, regardless of whether or not ice is present in every location. All 2D trackers are presented. All mean and median errors are in pixels. See column indexes below.

TWO-DIMENSIONAL TRACKING RESULTS – ICE-BOTTOM

ONLY ICE RANGE-LINES INCLUDED

	(1)	(2)	(3)	(4)	(5)	(6)
Ice-bottom mean error	13.71	12.15	23.15	3.21	11.20	1.70
Ice-bottom median error	0	0	0	0	0	0
Ice-bottom accuracy (3 px)	93.90%	97.22%	93.75%	95.62%	96.84%	98.20%
Ice-bottom accuracy (5 px)	94.23%	97.86%	93.85%	96.39%	97.21%	98.63%
Ice-bottom accuracy (10 px)	94.61%	97.97%	94.04%	97.44%	97.71%	98.99%

Table 4.5. Ice-bottom error and accuracy measurements obtained from the two-dimensional automated trackers. These results exclude the range-lines to which the ice-mask points to the inexistence of ice. Column indexes:

(1): Level-sets technique [22], fixed initial ice-surface contour value

(2): Level-sets technique [22], variable initial ice-surface contour value equal to the mean of the ice-surface layer

(3): Original HMM/Viterbi algorithm [14], no ground-truth provided

(4): Original HMM/Viterbi algorithm [14], entire ice-surface and available ice-bottom ground-truth provided

(5): MCMC [24]

(6): Modified HMM/Viterbi algorithm (ours), entire ice-surface and available ice-bottom ground-truth provided

TWO-DIMENSIONAL TRACKING RESULTS – MEAN TIME MEASUREMENTS

	(1)	(2)	(3)	(4)	(5)	(6)
Mean tracking time per frame (seconds)	147.10	116.44	1.42	1.35	3,625.18	1.25

Table 4.6. Timing measurements obtained from the two-dimensional automated trackers. All values in seconds. For tracking techniques (1), (2), (3) and (5), these values refer to the total tracking time of both ice layers.

(1): Level-sets technique [22]

(2): Level-sets technique [22]

(3): Original HMM/Viterbi algorithm [14], no ground-truth provided

(4): Original HMM/Viterbi algorithm [14], entire ice-surface and available ice-bottom ground-truth provided

(5): MCMC [24]

(6): Modified HMM/Viterbi algorithm (ours), entire ice-surface and available ice-bottom ground-truth provided

As expected, we find that the automated trackers consistently output better tracking results for the ice-surface layer, given that it is typically characterized by a far stronger and smoother interface in the radar echograms than the ice-bottom boundary. In evaluating the ice-bottom tracking error results presented in Tables 4.4 and 4.5, note that all solutions, with the exception of our modified HMM/Viterbi algorithm technique, are able to extract both the ice-surface and ice-bottom layers from a two-dimensional echogram. Providing a layer tracking algorithm with the location of the ice-surface boundary, such as we propose, simplifies the tracking framework and strengthens the location constraints for the ice-bottom. Furthermore, all of the previously published solutions included here make use of less information than our proposed solution. In all three, the a priori location of the ice-surface and the ice-mask raster were not used by the algorithms when they were initially published; the only input requirements of these techniques consist of the radar echogram itself and a set of tunable hyper-parameters such as weights and scaling factors.

Furthermore, our proposed techniques are the only ones to benefit from the novel empirically-derived cost models described in Chapter 3 of this document. The improved results obtained by our method come at the trade-off of requiring significantly more information regarding the surveyed terrain to be available. In a situation where the required additional data such as ice-surface and ice-mask datasets are not available, it may be impossible to make use of the techniques proposed here. However, these are rare exceptions and should not significantly affect the generality of these methods. Moreover, a small degradation in performance is noted to occur if the available ground-truth data points for the ice-bottom layer are not provided to our model, but this possibility was not explored in this work. The existing implementations of the level-sets technique and the MCMC tracker do not allow for ground-truth data points to be included in the tracking computations.

The mean execution time values presented in Table 4.6 refer to the tracking function calls only, and therefore assume all required data are already loaded, including the input echogram and ice-surface and ice-mask dataset where required. Similarly, the image pre-processing steps of all tracking techniques, such as the ones presented in Section 2.5 for our proposed HMM/Viterbi tracking framework, are excluded from the timing demonstrated in Table 4.6. The execution times of LSM tracker are linearly dependent on the total number of iterations performed by the algorithm. The results presented in Table 4.6 refer to the optimal number of iterations presented in Section 4.2.1. Therefore (1) refers to 400 iterations and (2) refers to 300 iterations.

4.8 - Results and evaluation of three-dimensional tracking techniques

THREE-DIMENSIONAL TRACKING RESULTS – ICE-SURFACE

	C3D/RNN (3)
Ice-surface mean error	3.68
Ice-surface median error	2
Ice-surface accuracy (3 px)	71.19%
Ice-surface accuracy (5 px)	82.21%
Ice-surface accuracy (10 px)	92.62%

Table 4.7. Ice-surface error and accuracy measurements obtained from the three-dimensional automated tracker. The only 3D tracker designed to detect the ice-surface layer in this comparison is the C3D/RNN tracker [31]. All mean and median errors are in pixels.

THREE-DIMENSIONAL TRACKING RESULTS – ICE-BOTTOM

	(1)	(2)	(3)	(4)
Ice-bottom mean error	9.21	7.90	8.88	4.48
Ice-bottom median error	1	1	4	1
Ice-bottom accuracy (3 px)	67.95%	52.18%	43.20%	73.59%
Ice-bottom accuracy (5 px)	72.60%	71.82%	57.98%	78.31%
Ice-bottom accuracy (10 px)	79.59%	81.18%	77.26%	85.54%

Table 4.8. Ice-bottom error and accuracy measurements obtained from the three-dimensional automated trackers. All 3D trackers are presented. All mean and median errors are in pixels.

(1): Modified HMM/Viterbi algorithm (ours)

(2): Original MRF/TRW-S algorithm [17]

(3): C3D + RNN [31]

(4): Modified MRF/TRW-S algorithm (ours)

THREE-DIMENSIONAL TRACKING RESULTS – MEAN TIME MEASUREMENTS

	(1)	(2)	(3)	(4)
Mean tracking time per frame (seconds)	25	1,742	45	1,792

Table 4.9. Timing measurements obtained from the two-dimensional automated trackers. All values in seconds. For the C3D/RNN tracking technique, the value refers to the total tracking time of both ice layers.

(1): Modified HMM/Viterbi algorithm (ours)

(2): Original MRF/TRW-S algorithm [17]

(3): C3D/RNN [31]

(4): Modified MRF/TRW-S algorithm (ours)

THREE-DIMENSIONAL TRACKING RESULTS – CROSSOVER MEASUREMENTS

	(1)	(2)	(3)	(4)	MC
Crossover mean error (meters)	25.0	26.0	25.8	24.1	23
Crossover median error (meters)	13	13	14	12	11

Table 4.10. Mean and median crossover absolute error results. All values in meters. For context, we also present the mean error found in the manually-corrected (MC) layer dataset.

(1): Modified HMM/Viterbi algorithm (ours)

(2): Original MRF/TRW-S algorithm [17]

(3): C3D/RNN [31]

(4): Modified MRF/TRW-S algorithm (ours)

The results presented in Table 4.7 include only the C3D/RNN tracker [31]. This is the only 3D image tracker included in this comparison designed to automatically detect the ice-surface layer. As in the 2D tracking case, the results are consistent with the fact that ice-surface layer tracking typically produces more accurate results than ice-bottom tracking, for the same reasons as discussed in Section 4.7. This neural network-based tracker solves for both ice layers

simultaneously, but may output an ice-bottom layer partially or entirely located above the detected ice-surface for the same slice. To obtain the ice-bottom tracking results for this technique presented in Table 4.8, we have thresholded the ice-bottom range-bin values so that these cannot exceed the range-bin value for the ice-surface layer found at the same direction-of-arrival bin.

Furthermore, the results point to the fact that three-dimensional tracking generates larger mean error values than the 2D tracking techniques do for nadir-looking two-dimensional echograms. The main reason for this is that tracking 3D imagery is more difficult than tracking 2D imagery because the image quality generally deteriorates on each side of nadir for the 3D images. When the original MRF/TRW-S algorithm was used to track 2D images using the 3D image (no nadir ground truth was provided), it outperformed the HMM/Viterbi 2D tracker when only the nadir result was compared [17]. The assumption was that the additional information in the 3D image allowed the nadir track to be tracked more precisely since the TRW-S algorithm had all 64 elevation angle bins and the 2D tracker only had the nadir elevation angle bin. The modified HMM/Viterbi tracking framework is also aided by the large amount of available ground-truth points in the 2D case. Also, TRW-S is an approximate algorithm which does not necessarily output the global optimal path during the inference step although it does seem to converge to a stable solution after a total of approximately 50 iterations.

As previously mentioned in Section 4.5, the ground-truth layer data for the three-dimensional imagery was obtained from manual correction of the results output by the original MRF/TRW-S algorithm technique [17]. For this reason, it is possible that the results presented in Table 4.8 present a bias favoring the results of our proposed implementation of the MRF/TRW-S method. The HMM/Viterbi technique, the original MRF/TRW-S solution, and our proposed implementation of the MRF/TRW-S method, all of which employ comparable layer tracking

models defined by cost functions following similar assumptions. Therefore, the error results of Table 4.8 may particularly disfavor the C3D/RNN solution, which does not use a probabilistic graphical model although it is trained on these ground truth so the effect may be minimal.

Along with the mean and median tracking results for the four automated trackers included in the three-dimensional tracking comparison, we present the crossover errors where flight lines crossed and two independent measurements were acquired over the same location. The two independent measurements allow the consistency of the approach to be analyzed. We present results for all automated trackers and for the manually-corrected ground truth data. Crossover errors can be visualized by overlaying the two (crossing) flight lines of interest in a digital elevation model. Fig. 4.2ab displays the flight paths (green and blue lines) of two data frames from the three-dimensional testing set used, as well as the swath imaged by each. The region surrounded by the red line is the intersection of the two swaths and represents the data points that were imaged both times. Fig. 4.2c displays the vertical error between the results obtained by tracking the ice-bottom layer at the intersection of the two data frames shown.

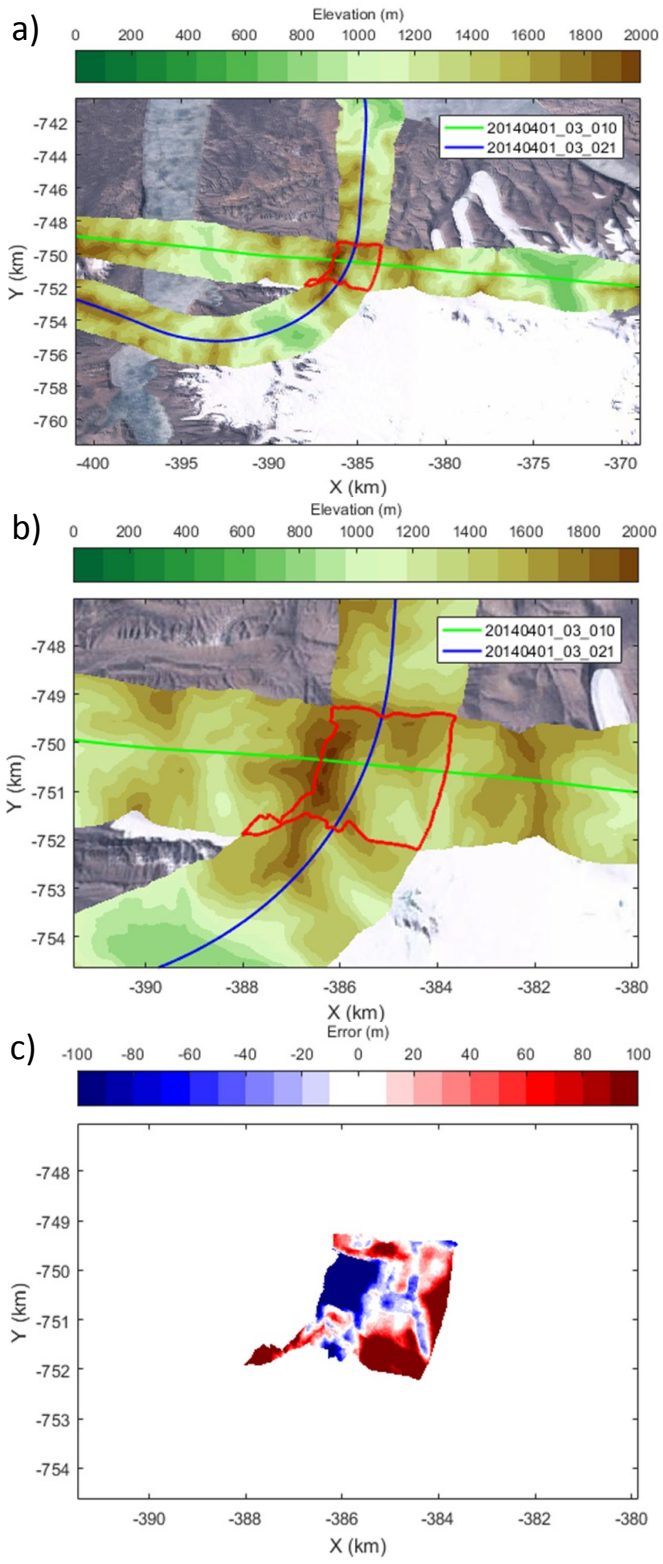

Fig. 4.2. Crossover visualization and error map.

Chapter 5 - CONCLUSION

We have demonstrated ice-bottom layer tracking in two-dimensional SAR images and three-dimensional SAR tomographic images of polar ice sheets using modified versions of two existing layer tracking solutions based on the HMM/Viterbi algorithm framework from [14] and MRF/TRW-S algorithm framework from [17]. We have also presented a thorough comparison between all ice bottom tracking solutions we found in the literature. Evaluation of the state-of-the-art in this area was previously complicated by the fact that the authors of the existing techniques presented the results obtained by their solutions using different datasets and error measurements. Furthermore, the published results of [22] and [24] excluded imagery that was deemed of low quality, whereas we have used precisely the same two-dimensional and three-dimensional datasets in all the comparisons presented. Substantial effort was made to generate an impartial analysis of the results obtained to help guide future research in this area.

Based on the results obtained in this comparison, our methods outperform the existing solutions while being fast enough to allow for corrections to be performed efficiently and in near real-time when errors occur in the automated result. However, our modifications require that more ancillary data regarding the surveyed ice sheets be available. Furthermore, our techniques are able to detect only the ice-bottom interface, whereas the majority of the existing solutions output both the ice-surface and ice-bottom layers.

The improvements proposed here arise from refinements made to the unary and binary cost functions of the probabilistic graphical models used in the original publications, which allow for the integration of further domain-specific knowledge and sources of evidence. Additional automated pre-processing steps are also applied on the two-dimensional radar echograms, such as data detrending, concatenation of adjacent data frames, inclusion of ground truth from previous

radar campaigns, and filtering of the first surface multiple from the image, which increase the accuracy of the tracker in adverse scenarios such as noisy data and very weak ice bed returns.

REFERENCES

[1] E. Rignot and R. H. Thomas, "Mass Balance of Polar Ice Sheets," *Science*, vol 297, pp. 1502-1506, 30 August 2002.

[2] A. Shepherd, E. Ivins, G. A. V. Barletta, M. J. Bentley, S. Bettadpur, K. H. Briggs, D. H. Bromwich, R. Forsberg, N. Galin, M. Horwath, S. Jacobs, I. Joughin, M. A. King, J. T. M. Lenaerts, J. Li, S. R. M. Ligtenberg, A. Luckman, S. B. Luthcke, et al., (2012), "A reconciled estimate of ice-sheet mass balance," *Science* 338(6111), 1183–1189.

[3] B. C. Douglas and W. R. Peltier, "The puzzle of global sea-level rise," *Physics Today*, 55(3):35–40, May 2002.

[4] E. Rignot, J. Mouginot, B. Scheuchl, M. van den Broeke, M.J. van Wessem and M. Morlighem, "Four decades of Antarctic Ice Sheet mass balance from 1979-2017," *Proceedings of the National Academy of Sciences*, 116 (2019) 1095-1103.

[5] H. Yu, E. Rignot, H. Seroussi and M. Morlighem, "Retreat of Thwaites Glacier, West Antarctica, over the next 100 years using various ice flow models, ice shelf melt scenarios and basal friction laws," *The Cryosphere*, 12 (2018) 3861-76.

[6] M. Morlighem, E. Rignot, H. Seroussi, E. Larour, H. Ben Dhia, and D. Aubry, "A mass conservation approach for mapping glacier ice thickness," *Geophys. Res. Lett.*, vol. 38, issue 19, 2011.

[7] A. Gilbert, G. Flowers, G. Miller, B. Rabus, W. Van Wychen, A. Gardner, and L. Copland, "Sensitivity of Barnes Ice Cap, Baffin Island, Canada, to Climate State and Internal Dynamics," *Journal of Geophyiscal Research – Earth Surface*, vol. 121, Issue 8, 2016.

[8] S. Gogineni, T. Chuah, C. Allen, K. Jezek and R. K. Moore. "An improved coherent radar depth sounder," *J. Glaciol*., 44(148), 659–669, 1998.

[9] F. Rodríguez-Morales, P. Gogineni, C. Leuschen, J. Paden, J. Li, C. Lewis, B. Panzer, D. Gomez-García Alvestegui, A. Patel, K. Byers, R. Crowe, K. Player, R. Hale, E. Arnold, L. Smith, C. Gifford, D. Braaten, C. Panton, "Advanced Multifrequency Radar Instrumentation for Polar Research," *IEEE Trans. Geosci. Remote Sens.*, vol.52, no.5, pp. 2824-2842, 2014.

[10] L. Shi et al., "Multichannel Coherent Radar Depth Sounder for NASA Operation Ice Bridge," *International Geoscience and Remote Sensing Symposium (IGARSS)*, 2010 IEEE International, 2010, pp. 1729 –1732.

[11] M. Al-Ibadi, J. Sprick, S. Athinarapu, V. Berger, T. Stumpf, J. Paden, C. Leuschen, F. Rodríguez, M. Xu, D. Crandall, G. Fox, D. Burgess, M. Sharp, L. Copland, W. Van Wychen, "Crossover Analysis and Automated Layer-Tracking Assessment of the Extracted DEM of the Basal Topography of the Canadian Arctic Archipelago Ice-Caps," *2018 IEEE Radar Conference*, Oklahoma City, OK, 2018, pp. 862-867.

[12] J. Li, J. Paden, C. Leuschen, F. Rodríguez-Morales, R. Hale, E. Arnold, R. Crowe, D. Gomez-Garcia, S. Gogineni, "High-Altitude Radar Measurements of Ice Thickness over the Antarctic and Greenland Ice Sheets as a part of Operation Ice Bridge", *IEEE Transactions on Geoscience and Remote Sensing*, 2012, vol. 50, no. 12, doi: 10.1109/TGRS.2012.2203822.

[13] J. Paden, T. Akins, D. Dunson, C. Allen, P. Gogineni, "Ice-Sheet Bed 3-D Tomography," *Journal of Glaciology*, vol.56, no.195, pp. 3-11, 2010.

[14] D. J. Crandall, G. C. Fox and J. D. Paden, "Layer-finding in radar echograms using probabilistic graphical models," *Proceedings of the 21st International Conference on Pattern Recognition* (ICPR2012), Tsukuba, 2012, pp. 1530-1533.

[15] A. Viterbi, "Error bounds for convolutional codes and an asymptotically optimum decoding algorithm," *IEEE Transactions on Information Theory*, vol. 13, no. 2, pp. 260-269, 1967.

[16] L. Rabiner, "A tutorial on HMMs and selected applications in speech recognition,". *Proc. IEEE*, 1989.

[17] M. Xu, D. Crandall, G. Fox, and J. Paden, "Automatic Estimation of Ice Bottom Surfaces from Radar Imagery," *IEEE International Conference on Image Processing*, 2017.

[18] V. Kolmogorov, "Convergent tree-reweighted message passing for energy minimization," *Transactions on Pattern Analysis and Machine Intelligence*, vol.28, no. 10, pp. 1568–1583, 2006.

[19] ArcticDEM, 2018. https://www.pgc.umn.edu/data/arcticdem/

[20] I. M. Howat, C. Porter, B. E. Smith, M. J. Noh, and P. Morin, "The Reference Elevation Model of Antarctica," *The Cryosphere*, 13, 665-674, 2019.

[21] C. M. Gifford, G. Finyom, M. Jefferson, Jr., M. Reid, E. L. Akers, and A. Agah, "Automated polar ice thickness estimation from radar imagery", *IEEE Trans. Image Process*., vol. 19, no. 9, pp. 2456-2469, Sep. 2010.

[22] M. Rahnemoonfar, G. C. Fox, M. Yari and J. Paden, "Automatic Ice Surface and Bottom Boundaries Estimation in Radar Imagery Based on Level-Set Approach," *IEEE Trans. Geosci. Remote Sens*., vol. 55, no. 9, pp. 5115-5122, Sept. 2017.

[23] A. Ilisei, A. Ferro, and L. Bruzzone, "A technique for the automatic estimation of ice thickness and bedrock properties from radar sounder data acquired at Antarctica," *International Geoscience and Remote Sensing Symposium*, 2012, pp. 4457–4460.

[24] S. Lee, J. Mitchell, D. Crandall, G. Fox, "Estimating bedrock and surface layer boundaries and confidence intervals in ice sheet radar imagery using MCMC," *International Conference on Image Processing*, 2014, pp. 111–115.

[25] J. E. Mitchell, D. J. Crandall, G. C. Fox, and J. D. Paden, "A semi-automatic approach for estimating near surface internal layers from snow radar imagery," *International Geoscience and Remote Sensing Symposium*, 2013, pp. 4110–4113.

[26] J. E. Mitchell, "A Progressive Study for the Automatic Analysis of Internal Layers from Polar Radar Imagery," Ph.D. thesis, 2018.

[27] C. Panton, "Automated mapping of local layer slope and tracing of internal layers in radio echograms," Annals of Glaciology, vol. 55, no. 67, pp. 71–77, 2014.

[28] L. Carrer and L. Bruzzone, "Automatic enhancement and detection of layering in radar sounder data based on a local scale hidden Markov model and the Viterbi algorithm," *Transactions on Geoscience and Remote Sensing*, vol. 55, no. 2, pp. 962–977, 2017.

[29] B. Smock, and J. Wilson. "Reciprocal pointer chains for identifying layer boundaries in ground-penetrating radar data." *International Geoscience and Remote Sensing Symposium*, 2012.

[30] V. Berger, M. Xu, D. Crandall, J. Paden, and G. Fox, "Automated Tracking of 2D and 3D Ice Radar Imagery Using Viterbi and TRW-S," *IEEE International Geoscience and Remote Sensing Symposium*, 2018.

[31] M. Xu, C. Fan, J. Paden, G. C. Fox and D. J. Crandall, "Multi-Task Spatiotemporal Neural Networks for Structured Surface Reconstruction," *IEEE Winter Conference on Applications of Computer Vision*, 2018.

[32] A. Viterbi, "The Viterbi Algorithm Demystified", 2017.
https://viterbischool.usc.edu/news/2017/03/viterbi-algorithm-demystified

[33] M. J. Wainwright, T. S. Jaakkola, and A. S. Willsky, “MAP estimation via agreement on trees: message-passing and linear programming,” *IEEE Transactions on Information Theory*, 51(11):3697:3717, November 2005.

[34] A. Arendt, et al., “Randolph Glacier Inventory [v2.0]: A Dataset of Global Glacier Outlines,” *Global Land Ice Measurements from Space*, Boulder Colorado, USA, Digital Media, 2012.

[35] J. Bergstra and Y. Bengio, “Random search for hyper-parameter optimization”, *Journal of Machine Learning Research*, vol.13, pp. 281-305, 2012.

[36] N. F. McIntyre, “Antarctic ice-sheet topography and surface–bedrock relationships,” *Ann. Glaciol.*, vol. 8, pp. 124–128, 1986.

[37] J. de Rydt, G. H. Gudmundsson, H. F. J. Corr, and P. Christoffersen, “Surface undulations of Antarctic ice streams tightly controlled by bedrock topography,” *Cryosphere Discussion*, vol. 6, no. 6, pp. 4485–4516, Oct. 2012.