2021

INFERENCE OF SURFACE VELOCITIES FROM OBLIQUE TIME LAPSE PHOTOS AND TERRESTRIAL BASED LIDAR AT THE HELHEIM GLACIER

Franklyn T. Dunbar II
University of Montana, Missoula

INFERENCE OF SURFACE VELOCITIES FROM OBLIQUE TIME LAPSE PHOTOS AND TERRESTRIAL BASED LIDAR AT THE HELHEIM GLACIER

By

FRANKLYN THOMAS DUNBAR II

Thesis presented in partial fulfillment of the requirements for the degree of Master of Science in Computer Science

The University of Montana
Missoula, MT
January 2022

Approved By:
Scott Whittenburg, Dean of The Graduate School
Graduate School

Douglas Brinkerhoff, Chair
Dept. of Computer Science

Jesse Johnson
Dept. of Computer Science

Toby Meierbachtol
Dept. of Geosciences

Contents

Abstract

Using time dependent observations derived from terrestrial LiDAR and oblique time-lapse imagery, we demonstrate that a Bayesian approach to glacial motion estimation provides a concise way to incorporate multiple data products into a single motion estimation procedure effectively producing surface velocity estimates with an associated uncertainty. This approach brings both improved computational efficiency, and greater scalability across observational time-frames when compared to existing methods. To gauge efficacy, we apply these methods to a set of observations from the Helheim Glacier, a critical actor in contemporary mass loss trends observed in the Greenland Ice Sheet. We find that the Helheim glacier exhibits a mean velocity of approximately 19md^{-1} and discuss the implications of these methods as they pertain to ongoing efforts to characterize the Greenland Ice Sheet and its contributions to global mean sea level rise.

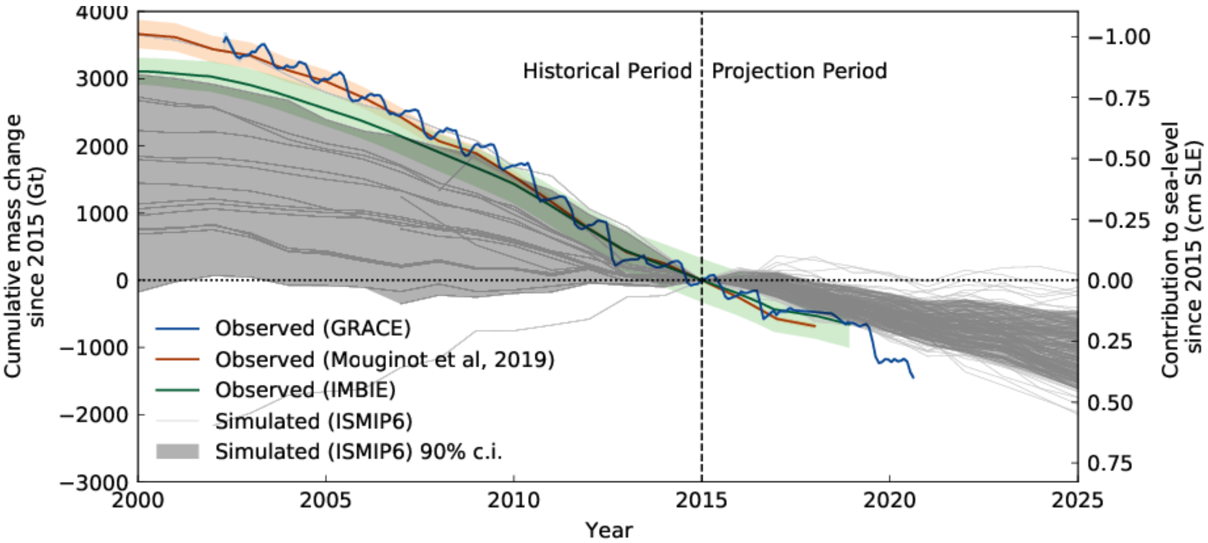

Figure 1: Observed and simulated historical mass changes from the Greenland Ice Sheet 2000–2020 in gigatons (Gt) and centimeters of sea level equivalent (cm SLE). Note that the 90% confidence interval from various models fail to envelop, or reasonably predict historical and contemporary observations. Image from Aschwanden et al. (2019)

1 Introduction

Contributions from melting ice sheets and glaciers to the global mean sea level (GMSL) are projected to increase sea levels anywhere between 0.26m to 0.77m by the year 2100 (Masson-Delmotte et al., 2018). The difference between a realized increase of 0.26m and 0.77m would mean a displacement of 10.4 million people currently living in high density coastal communities, which demonstrates the need for accurate estimates of changes in GMSL; low uncertainty estimates provide global leadership the necessary information needed to implement resource intensive socio-economic and infrastructural adaptions. Estimates of contemporary dynamics and longer term predictions are based on numerical models conditioned on past observations. The efficacy of modeling efforts to quantify the dynamics of a given feature of the cryosphere (be it an ice sheet or individual glacier) are gauged by their ability to reproduce observed behaviour. Figure 1 illustrates a set of simulations produced from various models, with their associated 90% confidence intervals in grey. Overlaid onto these simulation plots are observations of mass change from various studies. From this figure we can see the substantial discrepancy between these simulations and observations. Owing to complex dynamics and sensitivity to external forcings, estimating the future behaviour of the cryosphere is a challenge.

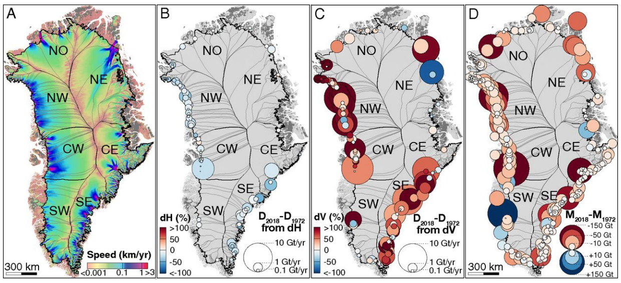

Figure 2: Current State of the GIS - A) Glacier catchments/basins for the GIS and seven regions overlaid on a composite map of ice speed (12). (B–D) For 1972–2018, the percentage (B) thickness change, (C) acceleration in ice flux from each basin, and (D) cumulative loss per basin. The surface area of each circle is proportional to the change in ice discharge caused by a change in (B) thickness or (C) speed; the (blue/red) color indicates the (positive/negative) sign of the change in (B) thickness, (C) speed, and (D) mass. Image From Mouginot et al. (2019)

There is considerable uncertainty in producing regional estimates of mass change due to the non-linearity in the physical behaviour of ice relative to its scale (Gardelle et al., 2013) (e.g. because a mountain glacier of mass X contributes Y to sea level rise does not imply that an ice sheet of mass 10×X contributes 10×Y to sea level rise), the intrinsic characteristics of a given outlet glacier, and the intra-region variance of environmental forcings. To reduce the scaling errors in regional estimates, their underlying models are constantly refined from a continually evolving understanding of dynamics through observation in the forms of both multi-year field campaigns and remote sensing.

The Greenland Ice Sheet (GIS) is a significant contributor to both the observed and predicted changes in GMSL, with current net contributions to GMSL at approximately 0.8 - 1.1 mm yr^{-1} (Mouginot et al., 2019). Mass loss from the GIS is partitioned between surface mass balance (or the net flux of glacier mass from the surface) and ice loss from marine terminating glaciers. The distribution of net annual loss between these mechanisms is one of the primary sources of uncertainty in current estimates, as the ice discharge from these marine terminating glaciers is highly sensitive to factors such as fjord bathymetry, ocean temperature and circulation (Howat et al., 2005). For example, hydrologic processes on the GIS's surface have proven difficult to model (Smith et al., 2017) and the dynamics of marine terminating glaciers exhibit larger variance than that of the ice sheet's interior (Parizek and Alley, 2004).

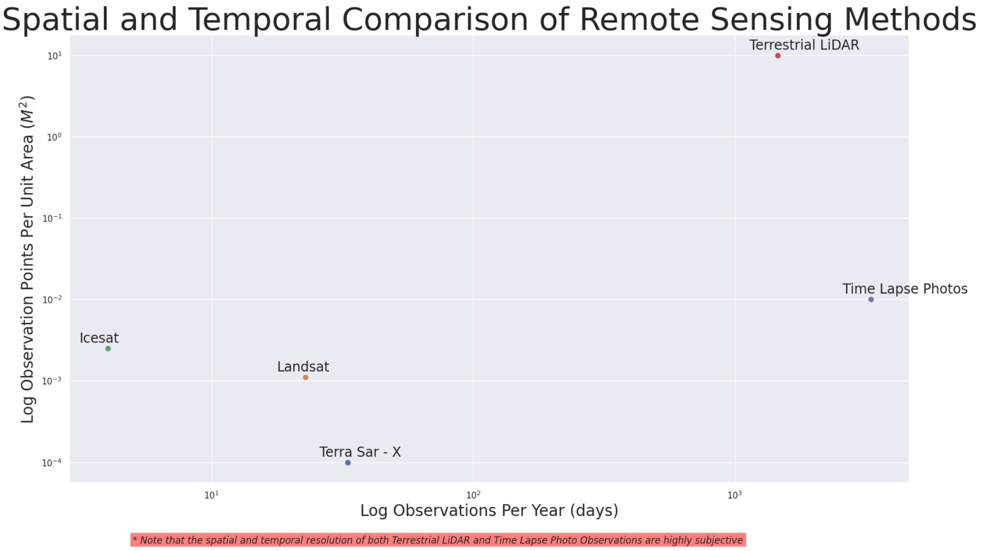

Figure 3: Log spatial vs log temporal resolution of remote sensing methods actively used in glaciology

Constraining uncertainties in modeling results of respective marine terminating glaciers is a critical step in the process of drawing a complete portrait of large-scale dynamics, which in turn produces better long term predictions. Remote sensing data are useful for characterizing large-scale dynamics. Existing remote sensing satellites, such as Landsat, typically sample at a spatial resolution of up to $900m^2$ per pixel, and a temporal resolution of once every 16 days (Masek, Rocchio, and Taylor, 2020). 30m resolution is useful for larger scale ice sheets and glaciers that may span 10's of kilometers, but these resolutions are not optimal for smaller scale features; techniques must be applied to a broad distribution of glacier sizes, and over time scales potentially at the hourly scale to fully capture the influence of ice flow moving mass from the high altitude accumulation zones down to warmer regions. High spatial resolution can potentially be useful in the extreme near terminus where peak surface velocities, rapid changes in stress distribution and calving occurs. Making observations of these dynamics at a high resolution serves to resolve their respective signals, such that casual forcings which occur at a frequency above that of satellite-based remote sensing (e.g. precipitation) may be identified. In contrast to satellite-based remote sensing, DSLR cameras and Terrestrial Based LiDAR (TBL) can capture observations from many perspectives with high spatial and temporal resolution and over a time-span limited solely by memory and power constraints.

These autonomous high-resolution observation products are not without drawbacks. Existing methods to process high-resolution TBL data are computationally expensive and limit the their application to small spatio-temporal scales. These novel methods have been applied to TBL and time-lapse imagery captured at Helheim Glacier in south-eastern Greenland. Change detection between sequential observations, whether between oblique time-lapse images, satellite based ortho-photos, or LiDAR derived point clouds serves to elucidate the physical behaviour of a given glacier across multiple epochs. Taking a Bayesian approach to utilizing oblique time-lapse imagery to infer glacial surface motion was first described in Brinkerhoff and O'Neel (2017) and Ahn and Box (2010), whose efforts provided a robust framework for using high resolution terrestrial data (e.g., mounted time-lapse camera) to infer glacial surface motion. Welty (2018) described the photogrammetry techniques used to incorporate these observations into the motion estimation procedure in Brinkerhoff and O'Neel (2017). Methods to detect change between sequential point clouds are well established; at the Helheim Glacier, LiDAR derived point cloud data was used to infer 3D motion between pairs of sequential scans using Particle Imaging Velocimetry (PIV) and Iterative Closest Point (ICP) as described in Gadomski (2016).

In this work, the Bayesian approach to motion estimation and photogrammetry described in Brinkerhoff and O'Neel (2017) and Welty (2018) (respectively) is extended to incorporate LiDAR derived point clouds for the purpose of inferring glacial surface velocities. The accuracy of results produced by the methods developed in this work are directly compared to estimates of velocity fields produced by both coherent point drift (Gadomski, 2016) and radar interferometer TerraSAR-X (TSX) (Joughin et al., 2010). In the following sections we develop our approach to motion estimation, describe the model's interface used to include time dependent observations from TBL and oblique time-lapse images, and discuss the relative performance of our approach to existing methods. We find that the velocity estimates are in good agreement with existing methods used to produce these estimates, and provide estimates with a quantifiable uncertainty in the form of a posterior covariance matrix.

2 A Bayesian Approach to Tracking Glacier Surface Features From Multiple Data Sources

The fundamental goal of estimating glacial motion and inferring velocity is to take a set of observations through time, and produce a state estimate for arbitrary spatio-temporal locations within the observation domain (e.g., velocity along a given axis). Because observations are both sparse and imperfect, an accurate quantification of uncertainty in the motion estimations is essential. Doing this requires accurately mapping the observation set to real world coordinates, utilizing elements of the observation set in temporally sequential order, and defining reasonable probability distributions over the glacier's observable states. Incorporating these observations to complete this process from various data sources can be described as a problem in feature tracking, or locating identical objects across multiple images. While the techniques of feature tracking and the inference of motion from observation sources such as DLSR images and other remote sensing products are ubiquitous in the field of computer vision, their application in glaciology poses unique problems (Harrison et al., 1992). Glaciers are often located in areas with severe weather, which can occlude the entire field of view for a given sensor. The instruments used, when deployed in environmentally volatile areas are subject to events such as severe snowfall, cold weather, earthquakes, power issues, and wildlife interference; all probable mechanisms for potentially disorienting and/or disabling the instruments to some degree until they are field serviceable. These factors necessitate an approach that is robust to glacier occlusion from weather and debris, variable instrument orientation, and discontinuous observation sets.

2.1 Glacier Motion Model

If we define a vector of a glacier's state within given spatial and temporal bounds via probability distributions along each axis (e.g. x, y, v_x, v_y), then then a distribution over the glacier's dynamics prior to a given observation may be described by a probability density function.

In order to represent a given feature's state as a probability density function, a Lagrangian state-space model is used to represent the motion of tracked features on the glacier's surface as they evolve through time. Originally described in Brinkerhoff and O'Neel (2017), this model's variables are the features' DEM-plane coordinates $\mathbf{x}_k$, map-plane velocities $\mathbf{v}_k$, DEM-plane elevation $\mathbf{z}_k$, and variance in the DEM elevation values δS. These variables form the state vector at the k-th time step $\mathbf{M}_k = [\mathbf{x}_k, \mathbf{v}_k, \mathbf{z}_k, \delta S_k]$. This model assumes that a specific feature moves tangential to the glacier's surface with its velocity subject to random accelerations, giving the discrete equations as follows:

$$\mathbf{x}_{k+1} = \mathbf{x}_k + \Delta t v_k + \frac{\Delta t^2}{2}\mathbf{a}_k \tag{1}$$

$$\mathbf{v}_{k+1} = \mathbf{v}_k + \Delta t \mathbf{a}_k \tag{2}$$

$$\mathbf{z}_{k+1} = S\left(\mathbf{x}_{k+1}\right) + \delta \mathbf{S}_{k+1} \tag{3}$$

$$\delta \mathbf{S}_{k+1} = \delta \mathbf{S}_k + \sigma_z \left\|\mathbf{v}\right\| \Delta t. \tag{4}$$

$\mathbf{a}_k$ are random acceleration values in both horizontal directions drawn from a normal distribution,

$$\mathbf{a}_k \sim \mathcal{N}\left(0, \Sigma_{v,k}\right), \tag{5}$$

where $\Sigma_{v,k}$ is a diagonal covariance matrix with entries given an assumed characteristic variance in glacier velocities. $S(\mathbf{x}_{k+1})$ is the reference surface for the glacier as provided by a DEM. Errors in the DEM are likely to be systematic, and thus we account for the uncertainty in their reference δS_{k+1} via a random walk dependent on the particle's motion over the surface. We specify the initial state $\mathbf{M}_0$:

$$\mathbf{x}_0 \sim \mathcal{N}\left(\mathbf{x}', \Sigma_{\mathbf{x}}\right) \tag{6}$$

$$\mathbf{v}_0 \sim \mathcal{N}\left(\mathbf{v}', \Sigma_{\mathbf{v}}\right) \tag{7}$$

$$\mathbf{z}_0 = S\left(\mathbf{x}_0\right) + \delta S_0 \tag{8}$$

$$\delta S_0 \sim \mathcal{N}\left(0, \Sigma_S\right) \tag{9}$$

where $\mathbf{x}'$ is the location of the particle in DEM-plane space, $\mathbf{v}'$ is an initial guess for velocity, and the various Σ are covariance matrices associated with these initial distributions.

More concisely, we can define this stochastic state-space model as a random vector drawn from a distribution conditioned on the previous state (Brinkerhoff and O'Neel, 2017), or

$$P\left(\mathbf{M}_k \mid \mathbf{M}_{k-1}\right) = \mathcal{N}\left(\mathcal{F}\left(\mathbf{M}_{k-1}\right), \Sigma_{v,k}\right) \tag{10}$$

Where $\mathcal{F}$ corresponds to Eq.'s [1-4] and Σ_v is the co-variance matrix associated with the model's process noise.

2.2 Particle Filtering

The initial state of the filter $\mathbf{M}_0$ as described by Eq.'s [6-9] in section 2.1 describes the general motion of the glacier without additional knowledge about a specific location. From the initial state distributions produced by this model, we seek to approximate the true probability distribution of the current state $\mathbf{M}_k$ with respect to all observations up to and including the current time $\mathcal{O}_k = \{\mathbf{o}_j : j \in 0, \ldots, k\}$, where $\mathbf{o}_k$ is an observation at time k. This is done by measuring the respective likelihoods for a set of samples (called particles) drawn from these state distributions with a likelihood estimation function unique to the observation product as described in section 3.

With this set of potential solutions and their associated likelihoods the state's probability distribution may be estimated under two assumptions, the first being that the transition between states is a Markov process, meaning that the state of a given time step $\mathbf{M}_k$ is independent of all previous states of that particle except $\mathbf{M}_{k-1}$:

$$P\left(\mathbf{M}_k \mid \mathbf{M}_{k-1}, \ldots, \mathbf{M}_0\right) = P\left(\mathbf{M}_k \mid \mathbf{M}_{k-1}\right) \tag{11}$$

This implies that the transition between states is independent of states prior to the previous state. The second assumption is that observations depend exclusively on the current state, and are independent of all previous states and observations:

$$P\left(\mathcal{O}_k \mid \mathbf{M}_k, \ldots, \mathbf{M}_0\right) = \prod_{k=1}^{N} P\left(\mathbf{o}_k \mid \mathbf{M}_k\right) \tag{12}$$

Using these assumptions in conjunction with Bayes' Rule (Tarantola, 2005) we can recursively update our distribution as additional measurements are added:

$$P\left(\mathbf{M}_k \mid \mathcal{O}_k\right) \propto P\left(\mathbf{o}_k \mid \mathbf{M}_k\right) P\left(\mathbf{M}_k \mid \mathcal{O}_{k-1}\right), \tag{13}$$

where the likelihood $P\left(\mathbf{o}_k \mid \mathbf{M}_k\right)$ describes the probability of observing $\mathbf{o}_k$ with samples from $\mathbf{M}_k$, and $P\left(\mathbf{M}_k \mid \mathcal{O}_{k-1}\right)$ is the prior distribution which describes how feasible a state is given the previous observations and the dynamics of the motion model. The product of the likelihood and the prior is proportional to the posterior distribution of the state $\mathbf{M}_k$ after consideration of all observations.

Because the posterior distribution of a given point's state is a product of the prior distribution and likelihood, the model naturally weights the influence of these two parameters depending on the availability of data. For example, in the case of failed instrumentation can assume that all potential points for the image are obscured from view, producing a constant likelihood and thus rendering the posterior distribution proportional only to the prior distribution, which is to say dependent only on the motion model for evolution beyond the last successful observation. Conversely if the observations are high fidelity (e.g. DLSR images were captured in optimal sunlight and/or LiDAR point clouds are available for the time frame in question) and the prior state distribution is relatively vague (high variance probability density function) then the likelihood would dominate and the images would contribute more to the posterior distribution.

In order to compute the prior state distribution, the posterior distribution must be propagated from $\mathbf{M}_{k-1}$ through the current state. In other words the best guess for the current state is the previous state that has been updated by the model dynamics described by Eq.'s [1-4] in section 2.1. Mathematically, this is formulated as the Chapman-Kolmogorov forward equation (Doucet and Johansen, 2009):

$$P\left(\mathbf{M}_k \mid \mathbf{o}_{k-1}\right)=\int P\left(\mathbf{M}_k \mid \mathbf{M}_{k-1}\right) P\left(\mathbf{M}_{k-1} \mid \mathbf{o}_{k-1}\right) \mathrm{d}\mathbf{M}_{k-1} \tag{14}$$

In this equation, we can see that the first term is the probability of the new state given the old state (e.g. the forward model) and the second term in the equation is the posterior distribution from the previous time step. Because of the non-linearity inherent to both capturing observations and glacial dynamics, computing the posterior distribution by taking an analytical solution is impossible and instead done through the use of an approximate numerical solution. The approached used is known variously as the Sequential Monte Carlo Method (Doucet and Johansen, 2009), sequential importance re-sampling, bootstrap filtering (Gordon, Salmond, and Smith, 1993), the Condensation Algorithm (Blake and Isard, 1997), each of which are commonly referred to as the particle filter. This so-called particle filter relies on a random sample of states ('particles') ($\mathbf{m}^i \sim P(\mathbf{M})$) that are sequentially updated by a stochastic state-space model and observations as they are available. This method assumes that we can represent the probability distribution of the state $P(\mathbf{M}_k)$ as a weighted set $\{(\mathbf{m}_k^i, \boldsymbol{w}_k^i) : i \in 1, \ldots, N\}$ of random particles, forming a probability mass function:

$$P(\mathbf{M}_k) \approx \sum_{i=1}^{N} w_k^i \delta\left(\bar{\mathbf{m}}_k - \mathbf{m}_k^i\right) \tag{15}$$

where $\delta(\cdot)$ is the Dirac delta function and N is the number of random samples. The respective state of each particle is drawn from a a probability distribution that represents the next state, or the 'proposal distribution', $q\left(\mathbf{m}_k^i \mid \mathbf{m}_{k-1}^i, \mathbf{O}_{k-1}\right)$, and each particles respective weight is computed as

$$w_k^i \propto w_{k-1}^i \frac{P\left(\mathbf{o}_k \mid \mathbf{m}_k^i\right) P\left(\mathbf{m}_k^i \mid \mathbf{m}_{k-1}^i\right)}{q\left(\mathbf{m}_k^i \mid \mathbf{m}_{k-1}^i, \mathbf{o_k}\right)}. \tag{16}$$

In this case a reasonable proposal distribution is to use the prior distribution at time k itself

$$q\left(\mathbf{m}_k^i \mid \mathbf{m}_{k-1}^i, \mathbf{o}_{k-1}\right) = P\left(\mathbf{m}_k^i \mid \mathbf{m}_{k-1}^i\right), \tag{17}$$

which leads to the simple scheme of evolving the particles according to the model dynamics and updating the weights via the likelihood at the current time step

$$w_k^i \propto P\left(\mathbf{o}_k \mid \mathbf{m}_k^i\right) = \mathcal{L}(\mathbf{m}_k^i), \tag{18}$$

where we have assumed constant weights at the previous time step, and $\mathcal{L}$ is a likelihood function described in sections 3.1.3 and 3.2.2, respectively.

Theoretically, the model approximates the posterior distribution given an adequate number of particles, but by nature of the particle filter's likelihood estimation step only a select few (e.g. less than 5% of the particles instantiated at initialization) would have a significant weight after several iterations. This is overcome by re-sampling at each step from the samples $\mathbf{m}_k^i$, with a probability given by the particles weight w_k^i using a re-sampling method (Carpenter, Clifford, and Fearnhead, 1999). This re-sampling step produces a set of particles with equal weights

$$w_k^i = \frac{1}{N}. \tag{19}$$

The resulting posterior distribution converges to an approximation to the true posterior probability. This algorithm operates as described in Algorithm 1.

3 Incorporating Time Dependent Observations

In this section we describe the methods used to incorporate time dependent observations from oblique time-lapse imagery and LiDAR derived point clouds to infer surface motions via a particle filter. To the best of the author's knowledge, inferring surface motion from time dependent observations via a particle filter has thus far only been accomplished with RGB images, as originally demonstrated by Brinkerhoff and O'Neel (2017) and Welty (2018). Section 2.2 illustrates a proven framework for utilizing arbitrary data products, provided they accommodate both a referenced likelihood estimation function for the respective particles that serves to approximate Eq. 18 / step 3 in Algorithm 1 and

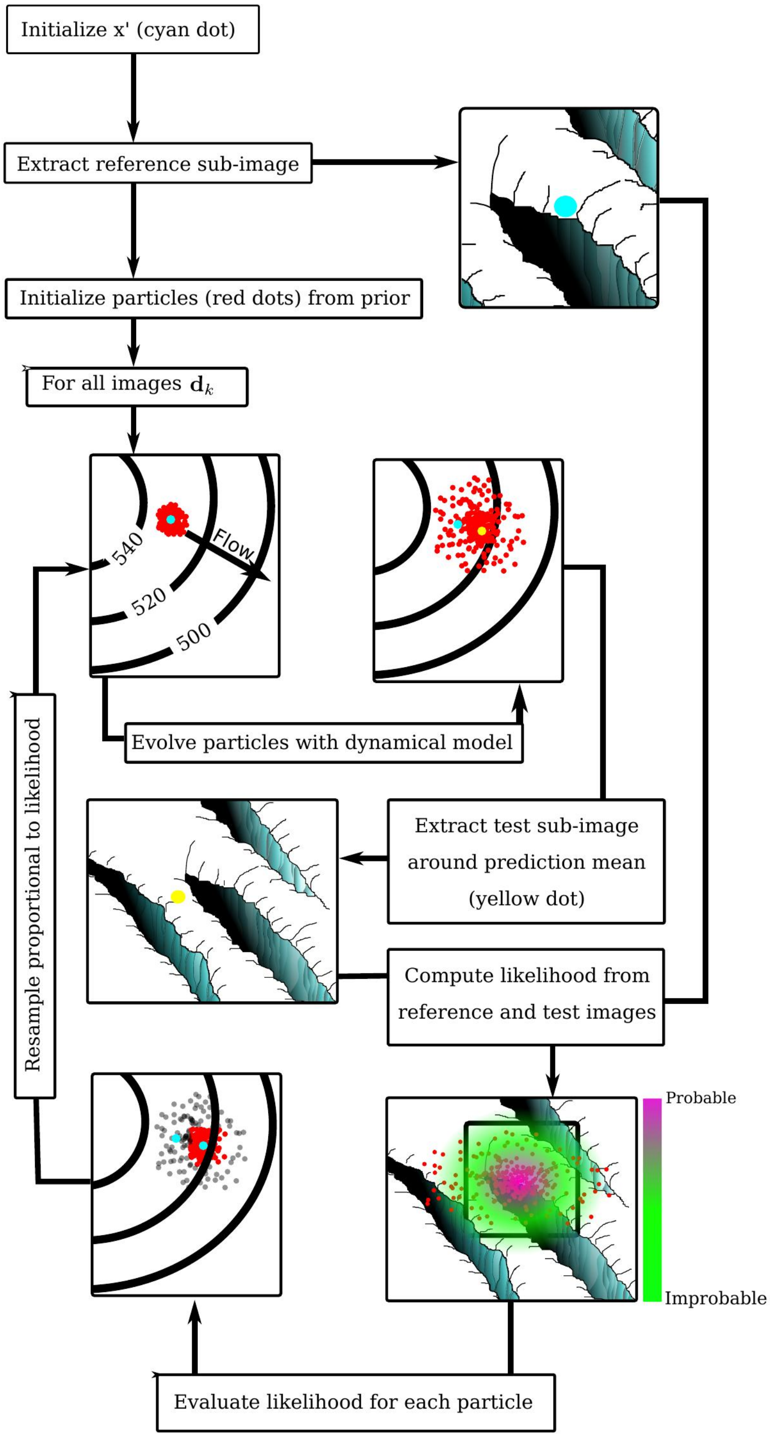
Initialize x' (cyan dot)
Extract reference sub-image
Initialize particles (red dots) from prior
For all images $\mathbf{d}_k$
540
520
500
Flow
Evolve particles with dynamical model
Resample proportional to likelihood
Extract test sub-image around prediction mean (yellow dot)
Compute likelihood from reference and test images
Probable
Improbable
Evaluate likelihood for each particle

Figure 4: Graphical depiction of the likelihood estimation and posterior approximation steps. Figure from Brinkerhoff and O'Neel (2017)

```
Algorithm 1: Algorithmic Summary of The Particle Filter Used in This Work
  Init: Initialize $\mathbf{m}_{k=0}^{i\in 0:N}$ state vectors centered about the feature(s) locations
        projected into world coordinates. At the mean location of the initialized
        state vectors, $[\bar{X}, \bar{Y}, \bar{Z}] = \frac{1}{N}\sum_{i=1}^{N}\mathbf{m}[\mathbf{x},\mathbf{y},\mathbf{z}]_{k=0}^{i}$, initialize the reference
        feature.
  for $\mathbf{o}_k$ in {observations} do
      1. Evolve particles $\mathbf{m}_k^i \sim P(\mathbf{M}_k)$ according to the stochastic state space model
         described in section 2.2.
      2. For each particle $\mathbf{m}_k^i \in [\mathbf{m}_k^0, \ldots, \mathbf{m}_k^{n-1}]$, extract a test feature $\mathbf{test}_k^i$ from $\mathbf{o}_k$ at the
         particle's respective location projected into the coordinate system of $\mathbf{o}_k$
      3. Compute $\mathbf{w}_k^i$ for each particle $[\mathbf{m}_k^0, \ldots, \mathbf{m}_k^{n-1}]$ by comparing its test observation to
         the initialized reference observation via a comparison metric unique to the data
         product so that $w_k^i = \mathbf{F}(\mathbf{test}_k^i, \mathbf{feat}_0)$
      4. Systematically re-sample the particles proportional to their likelihood.
      5. (Optional) Characterize the posterior probability distribution via its mean and
         covariance
         $$P(\mathbf{M}_k) \approx \mathcal{N}\left(\frac{1}{N}\sum_{i=1}^{N}\mathbf{m}_k^i, \Sigma_{M_k}\right)$$
```

an interface to transform between the filter's DEM plane coordinates to the observation's coordinate system (e.g. easting, northing, elevation to pixel coordinates in the case of RGB images).

3.1 Photogrammetry

The first step to identifying tangible motion in a glacier is done in image space. During the Boreal melt season of May through October, visible changes on a given glacier's surface can take place on the daily scale and the human eye can visually detect differences in perspective and object location. In glaciology, this foreground is typically the fjord in which a tidewater glacier terminates and the background a mountain range in which the glacier is settled. In the past decade, the use of digital time-lapse photography to capture geomorphic processes has become ubiquitous in research efforts (Harrison et al., 1992; Welty, 2018; Brinkerhoff and O'Neel, 2017). Digital camera's provide a relatively cost effective method of performing both high spatial and temporal resolution measurements that would otherwise be missed by satellite remote sensing due to low temporal frequency (e.g. LANDSAT captures images approximately every 2 weeks).

3.1.1 World Coordinates to Image Coordinates

Cameras capture an image by performing coordinate transformations that project an image in 3-dimensional (3-D) world coordinates (x,y,z) into a 2-dimensional pixel space (u,v). To interface a given image with a particle filter, we must define a camera model which reproduces the projection performed by the camera upon the image's capture, or

$$\mathbf{F}_{cam} : \mathbf{X}_{world} \mapsto \mathbf{U}_{pixel} \tag{20}$$

The first step to modeling this projection is defining a physical 3-D reference system with the camera's optical center and orientation at the origin. We can describe a given camera's physical 3-D coordinate system (x', y', z') is augmented via rotation $\mathbf{R}$ and translation $\mathbf{P}$ so that x' and y' point right and down in the image and z' points through the focal point of the lens when viewed from the back of the camera as shown in Figure 5. We can represent this mapping from $\mathbf{X}_{world}$ to the reciprocal $\mathbf{X}_{camera}$ as such:

$$\mathbf{X}_{camera} = \mathbf{R}(\mathbf{X}_{world} - \mathbf{P}) \tag{21}$$

where the rotation $\mathbf{R}$ translation $\mathbf{P}$ are matrices which describe the orientation of the camera's sensor at the time of capture, defined by indices from the vector

$$\boldsymbol{\beta} : [x_{world}, y_{world}, z_{world}, \theta_{yaw}, \theta_{pitch}, \theta_{roll}] \tag{22}$$

In practice, the cameras position $[x_{world}, y_{world}, z_{world}]$ is typically surveyed in the field, while angular orientation $[\theta_{yaw}, \theta_{pitch}, \theta_{roll}]$ is estimated through the process of pose estimation described in section 3.1.2. This transformed point $\mathbf{X}_{world}$ is then projected onto the 2-D image plane by dividing by z' to produce normalized coordinates $\mathbf{X}_c$:

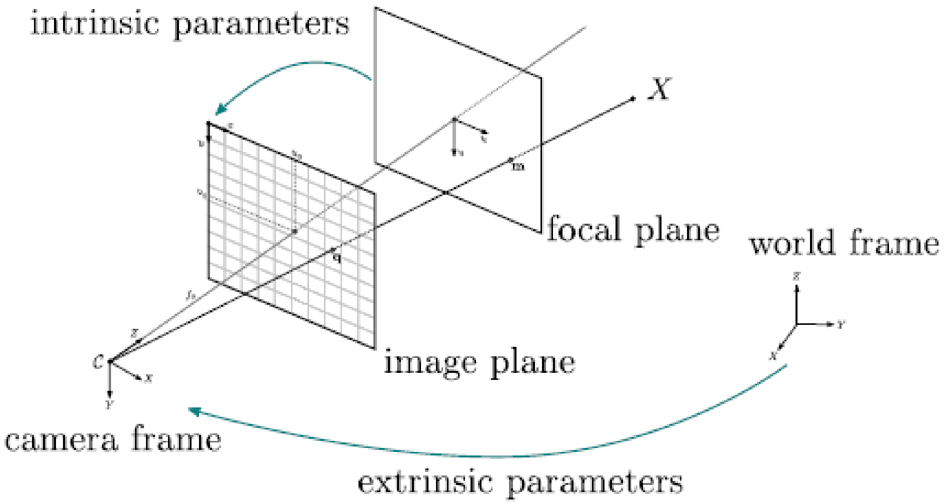

Figure 5: The Pinhole Camera Model (photo from https://openmvg.readthedocs.io/en/latest/openMVG/cameras/cameras/)

$$\mathbf{X}_c = \frac{1}{z'} \begin{pmatrix} x' \\ y' \end{pmatrix} \tag{23}$$

Where z' is the component of $\mathbf{X}_{camera}$ that represents the points physical distance from the camera's focal center. Neglecting the effect of lens distortion, the image's pixel coordinates $\mathbf{U}_i$ are a linear function of of these normalized camera coordinates:

$$\mathbf{U}_i = f_{pixel}\mathbf{X}_c + \mathbf{c}, \tag{24}$$

where f_{pixel} is the camera's focal length in pixels, and $\mathbf{c} = (c_x, c_y)$ is the center point (intersection of the camera's reference systems axes) in pixel coordinates. The conversion of focal length between pixel and physical units (m) is taken by multiplying the ratio of the sensor size in millimeters to the image size in pixels:

$$f_{pixel} = f'_{mm} \frac{sensor_{mm}}{sensor_{pixel}} \tag{25}$$

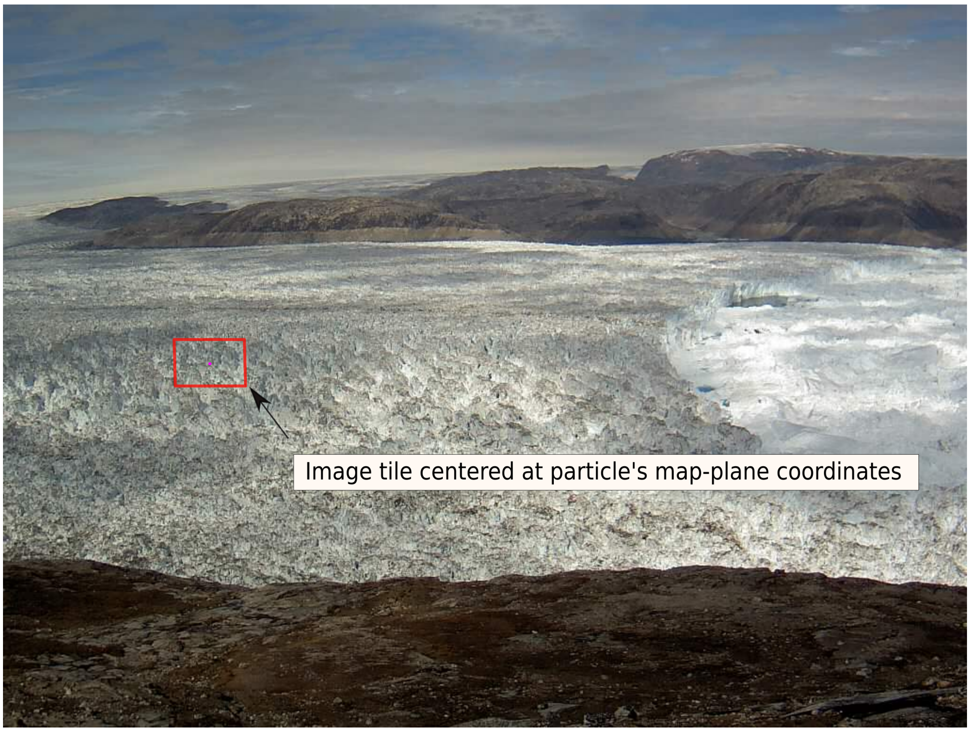

Figure 6: For a given point in map-plane coordinates, the image sub-sampling function extracts a tile centered around the point from the camera's perspective

3.1.2 Camera Model Calibration

The primary task of modeling the projection from world to image coordinates is estimating the correct orientation of the respective camera's sensor. Completing this task required both laboratory calibration and the use of landmarks (e.g. mountain tops, glacial erratics, installed markers) with their location in the world coordinate system . Camera's used for outdoor time-lapse photography are subject to considerable amounts of motion, and thus the orientation $\boldsymbol{\beta}$ must be defined for each image. The goal of refining the camera's pose parameters is to minimize the prediction error between image coordinates $\mathbf{U}$ from real world coordinates $\mathbf{X}_{world}$ using Eq. 20:

$$\arg\min_{\boldsymbol{\beta}} \sum_{j=0}^{n} \|\mathbf{F}_{cam}(\mathbf{X}_{world,j}, \boldsymbol{\beta}) - \mathbf{U}_j\|^2 \approx 0 \tag{26}$$

so as to estimate for a particular model parameter (e.g. view direction, focal length). Solving Eq. 26 for respective images is done by first performing this on reference image. A set of ground control points (GCP) containing the world coordinates of distinct features (mountain tops, glacial erratics, etc.) $\mathbf{X}_{world} = \{Easting_{gcp}, Northing_{gcp}, Elevation_{gcp}\}$

and their corresponding location in the anchor image $\mathbf{X}_{image} = \{U_{gcp}, V_{gcp}\}$ was compiled. With these GCPs and their world to image mappings (e.g. this disk is located at UTM point 3937272 E, 6695729 N, 934 M and at image point [500,4500]), camera model parameter estimation can be completed and further refined by methods described in Welty (2018).

3.1.3 Computation of the likelihood

This section describes the technique used to estimate a given particle's likelihood from time-lapse photos by way of solving Eq 18 and completing step 3 in 1 for time-lapse images. To begin, the particle filter's mean location is projected into the first image of the observation set from which a $m_r \times n_r$ reference template T centered at this point is extracted. This serves to capture the surface feature to be tracked through the period of interest. As test images become available, a test sub-image I with size m_t x n_t (with $m_t > m_r, n_t > n_r$) is extracted for each particle at it's location projected into image space. With these reference and test sub-images, we compute the area-averaged sum of squared differences between the reference template and test templates for all possible pixel offsets u', v' for which the reference template falls entirely within the test template:

$$\ell(u,v) = \frac{1}{m_r n_r} \sum_{u',v'} \left(T\left(u',v'\right) - I\left(u+u', v+v'\right)\right)^2, \tag{27}$$

for which the likelihood is defined as

$$\mathcal{L}(\mathbf{m}_k^i) \propto \exp\left(-\frac{\ell(u_k^i, v_k^i)}{\sigma_\ell^2 + \sigma_m^2}\right), \tag{28}$$

where σ_ℓ is the measurement uncertainty due to illumination changes and deformation, and σ_m is the uncertainty due to camera motion in pixels (Brinkerhoff and O'Neel, 2017; Nakhmani and Tannenbaum, 2008). In this expression, u_k^i and v_k^i are evaluated at a particle-wise basis by projecting their spatial coordinates $\mathbf{X}_{world}$ into camera coordinates.

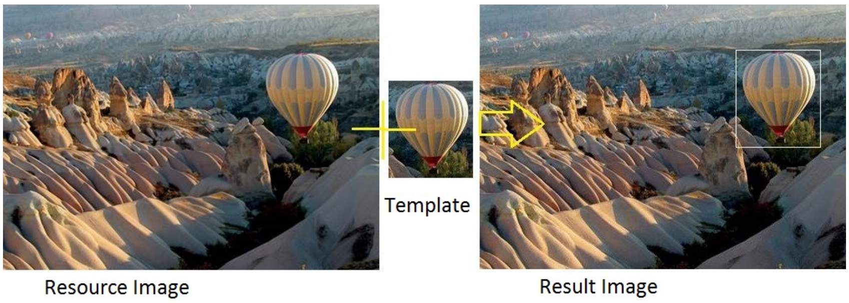

Figure 7: Illustration of the template matching procedure. Figure from: https://riptutorial.com/opencv/example/22915/template-matching-with-java

3.2 LiDAR Derived Point Clouds

Light Detection and Ranging (LiDAR) scanners provides measurements of distance from a sensor to a given object by illuminating a target with a pulse of laser light and capturing the return time. This return time provides distance measurements which can be used to construct a projected surface of a target from the sensor's vantage point, defined by:

$$d = c\frac{t}{2}, \tag{29}$$

where d is the distance between the scanner and object, c is the speed of light, and t is the captured return time.

A given scanner converts the measurements of the pulse-based emissions into raw distances, and further translates these distances into the scanners respective coordinate system (Gadomski, 2016). Typically centered inside the scanner, this coordinate system has the x axis aligned in the direction of the laser emission, the z axis pointed vertically up from the scanner, and the y axis orthogonal to the first two. To convert from raw distance to a 3-D geo-referenced point, a scanner records r, ϕ, and θ where r is the raw distance from the scanner to the target point, ϕ us the angle around the z-axis $[0 - 2\pi]$ and θ is the angular offset from the z-axis $[0 - \pi]$. These measurements (r, ϕ, θ) can be then converted into geo-referenced cartesian coordinates using the scanner's location, $\mathbf{X}_{scanner}$:

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} r \sin\theta \cos\phi \\ r \sin\theta \sin\phi \\ r \cos\theta \end{bmatrix} + \mathbf{X}_{scanner} \tag{30}$$

In aggregate, these measurements produce a set of discrete un-ordered points in a given frame of reference, otherwise known as a point cloud. When multiple independent point clouds are captured in sequential order they serve to provide time dependent observations of a given surface. In this section we describe the techniques developed in this work to utilize geo-referenced time dependent point clouds as observations within particle filter for estimating glacier surface velocities.

3.2.1 Model Coordinates to Point Cloud Coordinates

Point cloud data products are typically stored as a set of un-ordered points in 3-D, with some associated metadata. While this raw data format is memory efficient, it does to provide consistent information about the spatial relationship between points, nor a way to spatially index the discrete points from a continuous surface. Accessing a point cloud so as to extract descriptive features from the glacier's surface must be done in a way that is both computationally efficient, and admits for querying a set of discrete points by their relative location. To do this, the respective point clouds are sorted with a data structure known as the KD-Tree, or K-Dimensional Tree (KDT). The KDT allows for k-nearest neighbor and radial distance searches, while only requiring $O(\log N)$ construction time and $O(N)$ memory use (where N is the number of points in the set) (Bentley, 1975).

Because point clouds representing a glacier's surface have an inconsistent point density (Gadomski, 2016), querying the point clouds for a predetermined number of points would provide inconsistent features. A radial distance search is used to access the KDT: this search algorithm returns all spatially sorted points of dimension k within a euclidean distance R of search point P. In order to capture a complete portrait of the glacier's surface, KDT's are built using elements from the glacier's surface plane projection, or in other words the point cloud's x and y axes. This allows for the radial distance search

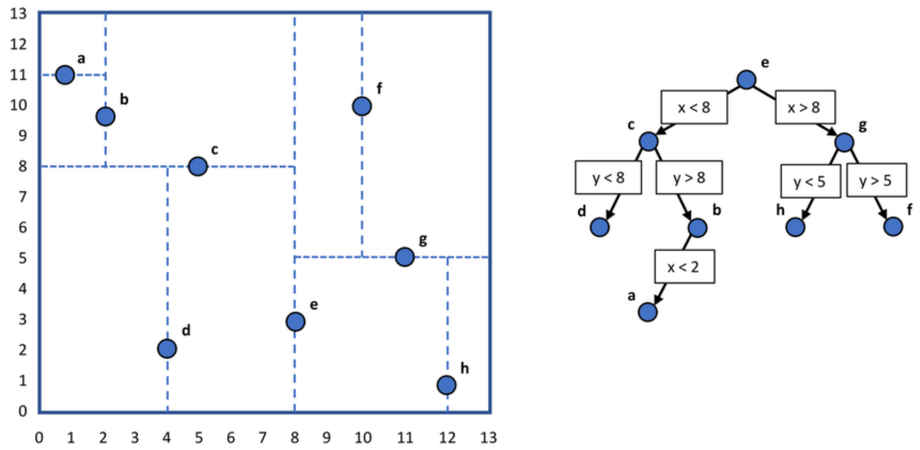

Figure 8: An example two-dimensional k-d tree (k=2) built from points a through h. Dividing planes are constructed by cycling through each coordinate and determining the median node (left). This gives rise to a tree structure (right) that, in conjunction with an input node, cant hen be searched recursively for a corresponding rectangular domain in physical space. The last leaf node is labeled as the best candidate for nearest neighbor and the tree is “unwound” to test other potential candidates. Figure from Ullrich and Zarzycki (2017)

to ignore surface topography, and sample points from a projected surface. For this application, the KDT only stores the point’s indices in the original data, allowing for the retrieval of elevation values upon querying the surface plane location from the tree.

3.2.2 Computation of the Likelihood

To gauge the degree of which two point clouds represent the same glacier surface feature (e.g. the same crevasse translated between the two positions) we utilize the technique of Gaussian Process Regression.

To approximate Eq. 18 and complete step 3 in 1, we pose the problem of gauging similarity between reference and test point clouds of the glacier’s surface:

Given an observed region of the glacier’s surface at time $\mathbf{k}_0$ centered at point $\mathbf{p}_0$ represented by a point cloud $(\mathbf{X},\mathbf{Z})_{train}$, paired with an observed region of the glacier at time $\mathbf{k}_1$ centered at $\mathbf{p}_1$ represented by the point cloud $(\mathbf{X},\mathbf{Z})_{test}$, compute the likelihood that $(\mathbf{X},\mathbf{Z})_{train}$ and $(\mathbf{X},\mathbf{Z})_{test}$ are representative of the same glacier surface feature (e.g. the same crevasse) translated between the two positions $\mathbf{p}_0$ and $\mathbf{p}_1$.

Taking the view that a given point cloud is a collection of random variables drawn from a probability distribution defined by the glacier’s true surface, we can formulate

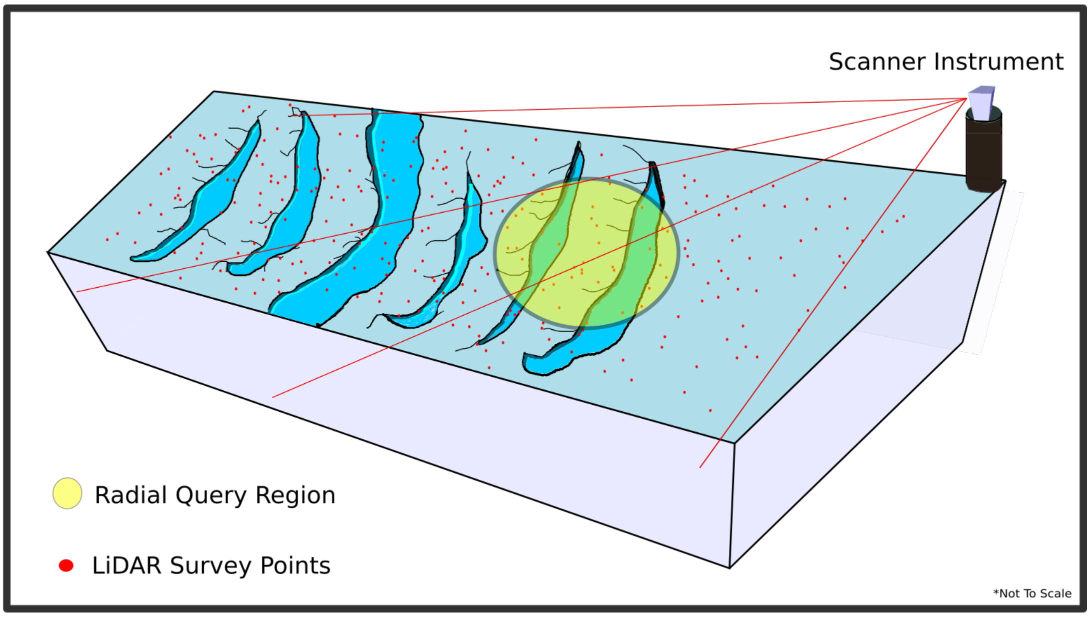

Figure 9: A graphical depiction of the observation points and the subsequent radial query schema used in this work. The red dots illustrate respective points from the point cloud and the yellow circle illustrates the radial query used to compute a respective particles likelihood.

$(\mathbf{X}, \mathbf{Z})$ as a Gaussian process (GP) of $\mathbf{Z}$ conditioned on $\mathbf{X}$. Gaussian processes are specified by a mean function $m(\mathbf{x})$ and covariance function $K(\mathbf{x}, \mathbf{x}^{'})$ of a real process $f(\mathbf{x})$ defined as:

$$m(\mathbf{x}) = \mathbb{E}[f(\mathbf{x})]$$

$$k(\mathbf{x}, \mathbf{x}^{'}) = \mathbb{E}[(f(\mathbf{x}) - m(\mathbf{x}))(f(\mathbf{x}^{'}) - m(\mathbf{x}^{'}))] \tag{31}$$

and the Gaussian process is written as

$$f(\mathbf{x}) \sim \mathcal{GP}(m(\mathbf{x}), k(\mathbf{x}, \mathbf{x}^{'})) \tag{32}$$

Here $k(\mathbf{x}, \mathbf{x}')$, the covariance function, specifies the hypothesized covariance between points in space and the matrix of which has entries $K_{ij} = k(\mathbf{x}_i, \mathbf{x}_j)$. In this work, we evaluate kernels using the stationary Matérn covariance function:

$$K(X, X') = \frac{1}{\Gamma(\nu)2^{\nu-1}} \left(\frac{\sqrt{2\nu}}{l} d(x_i, x_j) \right)^{\nu} K_{\nu} \left(\frac{\sqrt{2\nu}}{l} d(x_i, x_j) \right). \tag{33}$$

Where $d(x_i, x_j)$ is the euclidean distance between points, ℓ is the characteristic length-scale, and ν controls function smoothness (Rasmussen and Williams, 2006). The hyperparameters σ^2 and ℓ are found through marginal maximum likelihood estimation on the training set. We solve the stated problem by making the assumption that should the two point clouds indeed be representative of the same surface feature, then the test point cloud will provide adequate training data to produce a probabilistic model of the training point cloud's elevation values. This is done by evaluating the distribution of predicted elevation values at $\mathbf{X}_{train}$, which we call $\mathbf{Z}_{pred}$, as

$$P(\mathbf{Z}_{pred}|\mathbf{Z}_{train}, \mathbf{X}_{test}, \mathbf{X}_{train}) = \mathcal{N}(\mu, \Sigma), \tag{34}$$

with

$$\mu = K_*(K + \sigma^2 obs I)^{-1}\mathbf{Z}_{train} \tag{35}$$

$$\Sigma = K_{**} - K_*(K + \sigma^2_{obs} I)^{-1} K_*^T, \tag{36}$$

where μ is the posterior mean of $\mathbf{Z}_{pred}$ and Σ its posterior covariance. The matrices K, K_* and K_{**} are covariance matrices (between $\mathbf{X}_{train}$ and itself, $\mathbf{X}_{train}$ and $\mathbf{X}_{test}$, and $\mathbf{X}_{test}$ and itself, respectively). With a predictive distribution over elevations at test set locations, the likelihood of the true elevations $\mathbf{Z}_{test}$ given these predictions this prediction $\mathbf{Z}_{pred}$ is evaluated as

$$\mathcal{L}(\mathbf{m}_k^i) = \exp\left(-\big(n \log(2\pi) + \log(\mid \Sigma_k^i \mid) + \frac{1}{2}(\mathbf{Z}^i_{test,k} - \mathbf{Z}^i_{pred,k})^T \Sigma_k^{-1,i} (\mathbf{Z}^i_{test,k} - \mathbf{Z}^i_{pred,k})\big) \right) \tag{37}$$

where n are the number of predictions in $\mathbf{Z}^i_{pred,k}$.

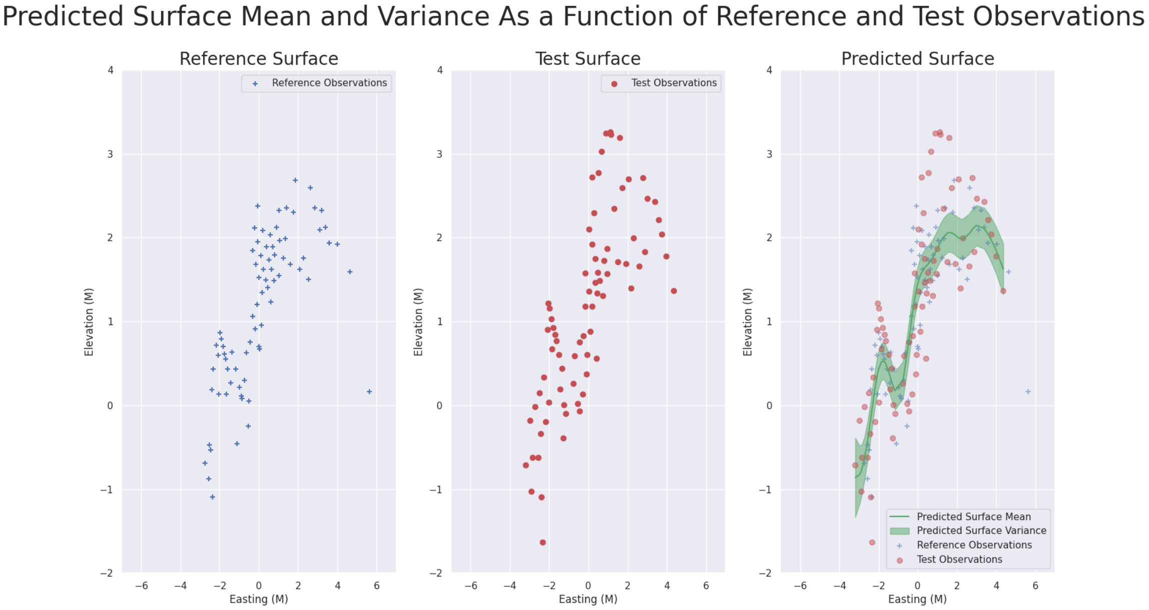

Figure 10: Illustration of the the likelihood estimation function used for point cloud data. The reference surface (blue) is used to generate a probabilistic model of the test surface using the easting/northing location as inputs. The log-likelihood of this model is evaluated using the true elevation values of the test surface, which serves to gauge congruity between the respective observations.

4 Application to the Helheim Glacier

The Helheim Glacier (66.38 N, 38.8 W) is a marine terminating glacier located in southeast Greenland, and is one of the Greenland Ice Sheet's (GIS) largest outlet glaciers. Boasting some of the fastest surface velocities of any glacier on the GIS, the Helheim exhibits velocities of $\approx$ 5-11 km yr^{-1} or $13-27$ m d^{-1}. The Helheim Glacier, in conjunction with the Jacobshavn and Kangerlussaq glaciers jointly account for 12% of the GIS's mass and jointly contain enough ice to raise the GMSL by 1.3 m (Parizek and Alley, 2004).

Because of the Helheim's contributions to the net ice flux of the GIS, it has been the subject of numerous field/remote sensing studies and is well positioned for the installation and management of field instruments. In the summer of 2015, the Autonomous Terrestrial Laser Scanner (ATLAS) system was installed at the Helheim to observe year round glacier dynamics at an unprecedented level of detail via time dependent point clouds of the glacier's surface approximately every 6 hours. ATLAS, in addition to two time-lapse cameras provides constant year round observations of the Helheim.

The particle filter described in Section 2.2 is used to infer surface motion at the Helheim Glacier from 2016-08-01 : 2016-10-01. The predicted velocities are then compared with the nearest spatially and temporally coincident velocity estimations produced via Coherent Point Drift (Gadomski, 2016) and TerraSAR-X surface images. Filter instances are initialized on an 8x16 grid DEM-plane points and on the overlapping region of the ATLAS and time-lapse camera view-sheds as shown in later sections.

4.1 Prior Distribution, Kernel Selection, and Filter Parameters

The particle filter described in 2.2 requires a defined prior distribution over the Helheim Glacier's dynamics. Samples drawn from this join distribution serves to provide the test hypothesis for model likelihood. To define the prior distribution over the initial parameters we use reported and observed velocities from Howat et al. (2005) and TerraSAR-X respectively, described in figure 11. The density of points m−2 is highly variable, and proportional to the relative distance to the scanner instrument. Because of this, a respective filter's kernel length scale ℓ and observational variance σ_{obs} are set to be inversely proportional to the requisite radius needed to observe 200 points of data. We can see that $d_1\ell_1 = d_2\ell_2$, and so it follows that $\ell_2 = \frac{d_1\ell_1}{d_2}$. Prior to utilizing point cloud data in the filter, points are sorted by their distance from the median value along the elevation axis upon the radial distance query. Points within the 60^{th} percentile of distance from the median elevation are kept, which was found to be the optimal value between signal and noise during development which was empirically determined to be the optimal amount of signal vs. noise after multiple test runs.

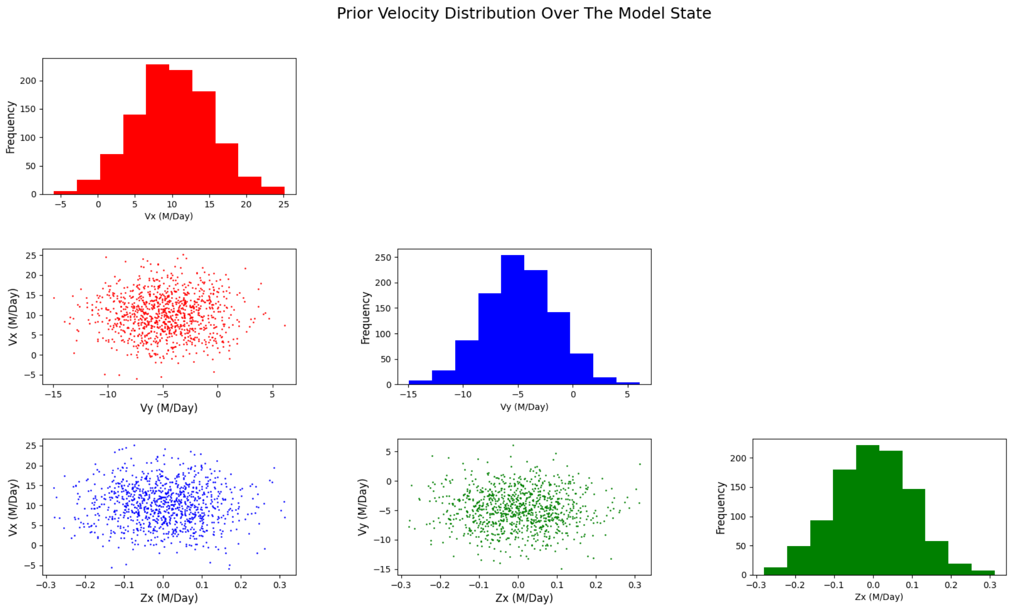

Figure 11: Samples drawn from the prior distribution of the particle filter over the respective velocities along the easting, northing, and elevation axes

4.2 Instrumentation and Data Coverage

Time-lapse images were taken every hour from 7:00 AM to 7:00PM local time, and the LiDAR scans were captured at 6 hour intervals. Respective terrestrial LiDAR and camera parameters are described in tables 1 and 2, in addition to spatial coverage illustrated in figure 12.

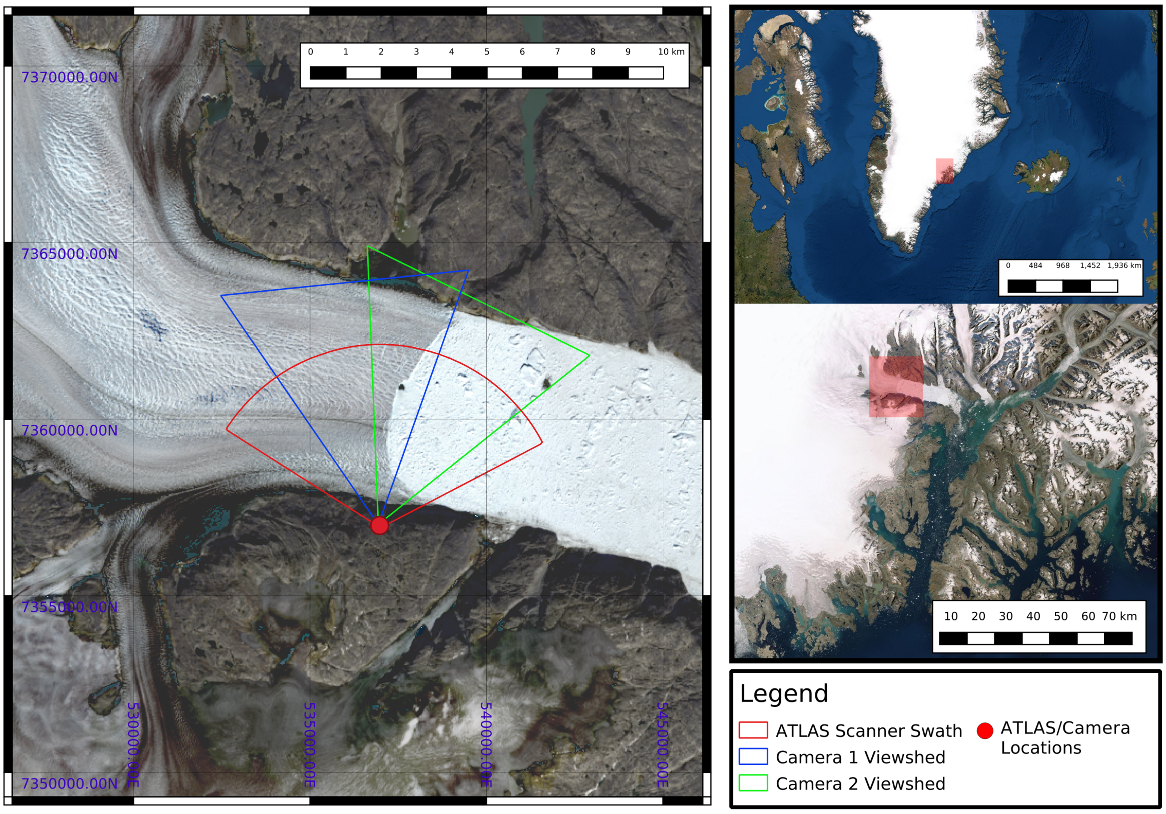

Figure 12: Relative location and approximate extent of the Helheim glacier captured by Landsat 8 on 08-28-16, overlaid by the view-sheds of ATLAS and the time-lapse cameras used in this work.

Table 1: Terrestrial LiDAR scanner specification

RIEGL Laser Measurement Systems Model VZ-6000	
Laser wavelength	1064 m^{-9}
Pulse rate	50 khz
Max range	6 km
Points/scan (apprx.)	50,000,000
Time per scan (apprx.)	30 m
ϕ range	$135° - 234°$
Angular velocity of scanner head	3.3degm^{-1}
Scan Interval	6 hours

Table 2: Camera instrument specification

Stardot - NetCam SC Multi-Megapixel Hybrid IP Camera	
Resolution (pixels)	(1024,768)
Sensor Size (mm)	(5.7,4.28)
Focal Length (pixels)	(1794,1338)
ISO	Auto
Frame rate	Auto

4.2.1 Results

We take the results of the set of respective filter's posterior estimates across the defined time-span and compare them to estimates of velocity fields produced by both coherent point drift (Gadomski, 2016) and radar interferometer TerraSAR-X (TSX) (Joughin et al., 2010). We show that across all epochs, there is agreement in root median square error (RMSE) between the filter produced velocity estimates of approximately 1 m d^{-1} with respect to CPD and 2 m/day with respect to TSX (Fig. 13). Across the time series of velocity estimates, we smooth estimates by taking the median value of the given posterior estimate and its 8 nearest neighbors. The lower subplot in Fig. 14 illustrates the associated uncertainty to each corresponding filter instance. Note how the uncertainty increases with gaps in observational data.

5 Discussion

Thus far, methods to utilize various data products for surface motion estimate have been completed independently. In (Brinkerhoff and O'Neel, 2017), oblique time-lapse images are used to estimate the velocities and the Columbia Glacier in South-central Alaska while (Gadomski, 2016) explores various methods used to estimate velocities from TBL derived point clouds at the Helheim Glacier. By generalizing existing methods to accommodate both oblique time-lapse images and TBL point clouds, we have coupled multiple observational data products towards the goal of motion estimation. This approach makes the observational time-span more robust to equipment failure, occlusion from weather, and sources of uncertainty unique to the respective instruments. Provided there are frequent observations with minimal gaps, the design of the particle filter used in this work is

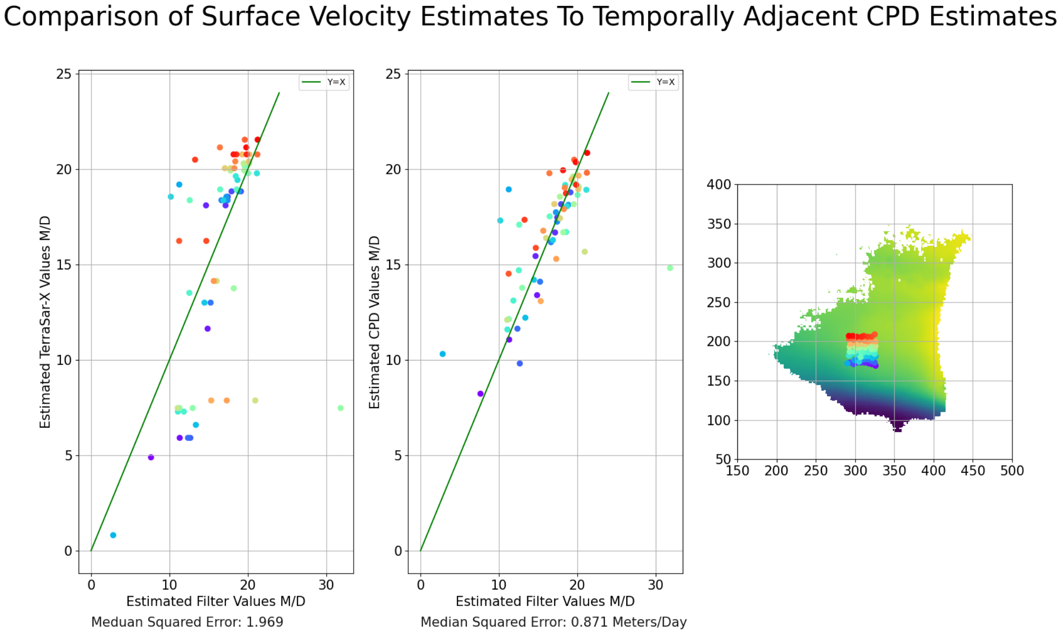

Figure 13: Depiction of surface velocity estimates from temporally adjacent estimates from coherent point drift and TerraSar-X. The plot on the far right shows the tracker locations super-imposed on rasterized velocity estimates produced by coherent point drift.

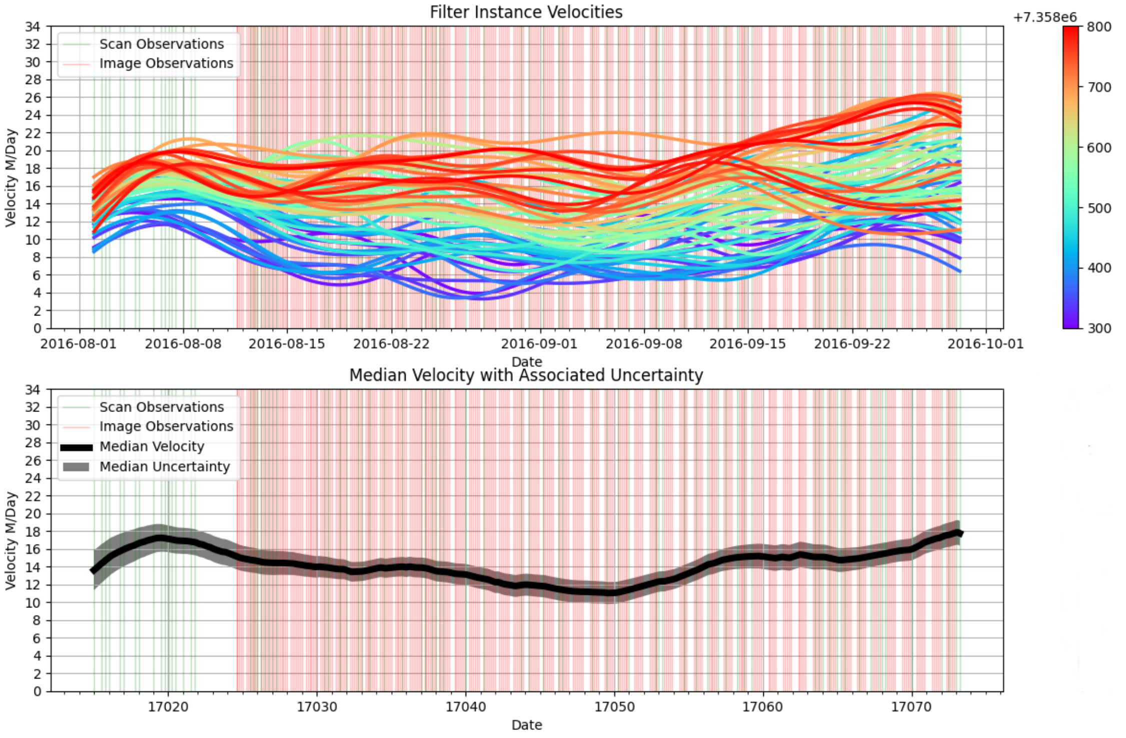

Figure 14: Depiction of Inferred Tracker Velocities. The lower plot shows the median velocity and median uncertainty for all plots. The color-bar in the both plots represents the relative northing position, which in this case is proportional to distance from the glacier's margin. Solid green lines represent scan observations and solid red lines represent image observations.

trivially generalized to produce estimates across time-spans from a few days up to several months and at resolutions proportional to the available observations. This flexibility in a given estimate time-span, and quantified uncertainty can provide the requisite knowledge of dynamics at the terminus needed to draw conclusions about environmental influences, and resolve both daily and seasonal changes in the respective velocity signals.

In the case of the time span used in this work, the Helheim's terminus is, for the most part, out of view of the second camera's view-shed (figure 12) and both camera's are mounted on the same structure. Computing a given particle's likelihood from time-lapse photos (section 3.1.3) resolves motion perpendicular to the camera's focal axis but struggles to resolve motion that is parallel to the focal axis. Because of this, this likelihood approximation scheme performs best when multiple perspectives capture images of the glacier's surface from peripheral locations. Utilizing the point cloud data in conjunction with time-lapse photos ameliorates this issue, as the point cloud captures the 3-D structure of the glacier's surface as opposed to a 2-D projection; observed motion is relatively consistent across all axes.

5.1 Difficulties With This Approach

During development we found the filter's performance to be highly sensitive to the number of points used for likelihood estimation, and co-variance function parameters. What is likely due to the extraordinary resolution of the ATLAS system, The point cloud data used in this work is notoriously noisy; this signal from the underlying surface motion is strongly aberrated by noise from a constantly evolving surface of serac and crevasse structures. In addition to the challenges presented by this noisy data, the density of observational points linearly decreases with distance from the scanner instrumentation. This problems are solved by scaling the respective filter's query radius, co-variance function length-scale, and the assumed observational variance by the distance from the scanner at initialization, which assumes that the particular filter instance remains at a relatively constant distance from the scanner during the observational window prior to re-initialization. The issue of surface noise is solved by implementing what amounts to a low-pass filter to pre-process the observational points prior to filter utilization. In addition to issues stemming from high resolution data, performance at the margins of the LiDAR scanner instruments margins remains questionable. In Figure 14 we can see that there are notable discrepancies that are not physically reasonable estimates. Most egregiously there is a velocity span of 5-20 meters per-day within a span of 500 meters. We speculate that this is due to filter instances utilizing data near the occlusion boundary imposed by the scanner instruments fore-ground.

Algorithmically, there are several bottlenecks in the likelihood estimation scheme for the the point cloud data. Computing the surface prediction μ, its associated uncertainty σ and the likelihood requires computing 3 distance matrices and performing 2 matrix inversions, resulting in an overall computational complexity (which is to say the amount of work needed to be done by a given machine's processing hardware) proportional to $O(n^3)$.

Physically improbable values may be computed in posterior velocity estimates. We address this issue by taking the median velocity between a given filter's posterior and the 8 nearest posterior estimates in DEM plane coordinates, serving to provide higher fidelity estimates than any filter instances independently.

5.2 Future Improvements

There is ample room for improvement in the point cloud likelihood estimation scheme. Most notably is the fast Gauss transform, which may be used in Gaussian Processes Regression to approximate kernel matrices bringing the calculation of the sum of N Gaussians at M points from $\mathcal{O}(MN)$ to $\mathcal{O}(M+N)$ (Raykar and Duraiswami, 2006).

6 Conclusion

By devising a method to incorporate time dependent point cloud data into existing surface motion estimation methods, we can effectively produce estimates that incorporate multiple data products and robustly characterize uncertainty. When applying this novel approach to the Helheim Glacier, we can see similar estimates to conventional methods (e.g. Coherent Point Drift). These methods may be efficiently generalized to produce estimates across relatively long time spans (e.g. multiple months), and to perform the computational workload across multiple processing units. Producing temporally high resolution velocity estimates across broad time spans can enable a more concise understanding of the influence of environmental forcings (e.g. submarine melting, meltwater discharge, etc.) and their overall impacts on GIS dynamics.

7 Acknowledgements

We thank Ethan Welty for his insight into the open source software GLIMPSE and techniques from photo-grammetry, Shad O'Neal for facilitating institutional collaboration, Michael Shahin for graciously providing the necessary time-lapse photos and point cloud data used in this work, and Pete Gadomski for insights into Coherent Point Drift.

References

Ahn, Yushin and Jason Box (Oct. 2010). “Instruments and Methods Glacier velocities from time-lapse photos: Technique development and first results from the Extreme Ice Survey (EIS) in Greenland”. In: *Journal of Glaciology* 56, pp. 723–734. DOI: 10.3189/002214310793146313.

Aschwanden, Andy et al. (2019). “Contribution of the Greenland Ice Sheet to sea level over the next millennium”. In: *Science Advances* 5.6, eaav9396. DOI: 10.1126/sciadv.aav9396. eprint: https://www.science.org/doi/pdf/10.1126/sciadv.aav9396. URL: https://www.science.org/doi/abs/10.1126/sciadv.aav9396.

Bentley, Jon Louis (1975). “Multidimensional binary search trees used for associative searching”. In: *Communications of the ACM* 18.9, pp. 509–517.

Blake, Andrew and Michael Isard (Jan. 1997). “The CONDENSATION algorithm - conditional density propagation and applications to visual tracking”. In:

Brinkerhoff, D. and S. O’Neel (2017). “Velocity variations at Columbia Glacier captured by particle filtering of oblique time-lapse images”. In: *arXiv: 1711.05366* 77, pp. 190–209.

Carpenter, J., P. Clifford, and Paul Fearnhead (Mar. 1999). “Improved particle filter for nonlinear problems”. In: *Radar, Sonar and Navigation, IEE Proceedings* - 146, pp. 2 –7. DOI: 10.1049/ip-rsn:19990255.

Doucet, Arnaud and Adam Johansen (Jan. 2009). “A Tutorial on Particle Filtering and Smoothing: Fifteen Years Later”. In: *Handbook of Nonlinear Filtering* 12.

Gadomski, Peter (Dec. 2016). “Measuring Glacier Surface Velocities With LiDAR: A Comparison of Three-Dimensional Change Detection Methods”. PhD thesis.

Gardelle, Julie et al. (Aug. 2013). “Region-wide glacier mass balances over the Pamir - Karakoram - Himalaya during 1999-2011”. In: *The Cryosphere* 7. DOI: 10.5194/tc-7-1263-2013.

Gordon, N.J., D.J. Salmond, and A.F.M. Smith (May 1993). “Novel Approach to Nonlinear/Non-Gaussian Bayesian State Estimation”. In: *Radar and Signal Processing, IEE Proceedings F* 140, pp. 107 –113. DOI: 10.1049/ip-f-2.1993.0015.

Harrison, W. D. et al. (1992). "The determination of glacier speed by time-lapse photography under unfavorable conditions". In: *Journal of Glaciology* 38, pp. 257–265.

Howat, I. M. et al. (2005). "Rapid retreat and acceleration of Helheim Glacier, east Greenland". In: *Geophysical Research Letters* 32.22. DOI: https://doi.org/10.1029/2005GL024737. eprint: https://agupubs.onlinelibrary.wiley.com/doi/pdf/10.1029/2005GL024737. URL: https://agupubs.onlinelibrary.wiley.com/doi/abs/10.1029/2005GL024737.

Joughin, Ian et al. (2010). "Greenland flow variability from ice-sheet-wide velocity mapping". In: *Journal of Glaciology* 56.197, 415–430. DOI: 10.3189/002214310792447734.

Masek, Jeffrey G, Laura Rocchio, and Michael P Taylor (2020). *NASA Landsat Science.* URL: https://landsat.gsfc.nasa.gov/about.

Masson-Delmotte, V. et al. (2018). "IPCC Special Report on the impacts of global warming". In:

Mouginot, Jérémie et al. (2019). "Forty-six years of Greenland Ice Sheet mass balance from 1972 to 2018". In: *Proceedings of the National Academy of Sciences* 116.19, pp. 9239–9244. ISSN: 0027-8424. DOI: 10.1073/pnas.1904242116. eprint: https://www.pnas.org/content/116/19/9239.full.pdf.

Nakhmani, Arie and Allen Tannenbaum (2008). "Particle filtering using multiple cross-correlations for tracking occluded objects in cluttered scenes". In: *2008 47th IEEE Conference on Decision and Control*, pp. 652–657. DOI: 10.1109/CDC.2008.4738656.

Parizek, Byron R. and Richard B. Alley (May 2004). "Implications of increased Greenland surface melt under global-warming scenarios: Ice-sheet simulations". English (US). In: *Quaternary Science Reviews* 23.9-10. Funding Information: This work was supported by a NASA/GSFC Graduate Student Researchers Program Fellowship (to B.R. Parizek). This material is based in part upon work supported by the US National Science Foundation under grants including Nos. 0126187 and 9814774 and by the comer foundation. We thank S. Anandakrishnan, T.K. Dupont, M. Fahnestock, C.L. Hulbe, T. Johnston, J. Li, D.R. MacAyeal, J. Saba, M.K. Spencer, B. Voight, and H.J. Zwally for their contributions. The helpful remarks of the Scientific Editor, P. Clark, and the critical reviews offered by K. Cuffey and S. Marshall improved the clarity of this manuscript. Special thanks to R.A. Bindschadler (B.R. Parizek's technical advisor at NASA) and H.H. Parizek for their encouragement and support throughout this research effort., pp. 1013–1027. ISSN: 0277-3791. DOI: 10.1016/j.quascirev.2003.12.024.

Rasmussen, CE. and CKI. Williams (Jan. 2006). *Gaussian Processes for Machine Learning*. Adaptive Computation and Machine Learning. Cambridge, MA, USA: MIT Press, p. 248.

Raykar, Vikas Chandrakant and Ramani Duraiswami (2006). "Fast large scale Gaussian process regression using the improved fast Gauss transform". In:

Smith, Laurence C. et al. (2017). "Direct measurements of meltwater runoff on the Greenland ice sheet surface". In: *Proceedings of the National Academy of Sciences* 114.50, E10622–E10631. ISSN: 0027-8424. DOI: 10.1073/pnas.1707743114. eprint: https://www.pnas.org/content/114/50/E10622.full.pdf. URL: https://www.pnas.org/content/114/50/E10622.

Tarantola, Albert (2005). *Inverse problem theory and methods for model parameter estimation*. eng. Philadelphia, PA: Society for Industrial and Applied Mathematics. ISBN: 0898715725.

Ullrich, Paul and Colin Zarzycki (Mar. 2017). "TempestExtremes: A framework for scale-insensitive pointwise feature tracking on unstructured grids". In: *Geoscientific Model Development* 10, pp. 1069–1090. DOI: 10.5194/gmd-10-1069-2017.

Welty, Ethan (Nov. 2018). "High-precision photogrammetry for glaciology". PhD thesis. DOI: 10.13140/RG.2.2.20751.64164.