

ASSESSMENT OF ICESAT-2 LEVEL 3A PRODUCTS FOR SNOW DEPTH
ESTIMATION IN REMOTE, MOUNTAINOUS WATERSHEDS

by

Colten Michael Elkin



A thesis

submitted in partial fulfillment

of the requirements for the degree of

Master of Science in Geophysics

Boise State University

May 2021

© 2021

Colten Michael Elkin

ALL RIGHTS RESERVED

BOISE STATE UNIVERSITY GRADUATE COLLEGE

DEFENSE COMMITTEE AND FINAL READING APPROVALS

of the thesis submitted by

Colten Michael Elkin

Thesis Title: Assessment of ICESat-2 level 3A products for Snow Depth Estimation in Remote, Mountainous Watersheds

Date of Final Oral Examination: 03 March 2021

The following individuals read and discussed the thesis submitted by student Colten Michael Elkin, and they evaluated the student's presentation and response to questions during the final oral examination. They found that the student passed the final oral examination.

Ellyn M. Enderlin, Ph.D. Chair, Supervisory Committee

HP Marshall, Ph.D. Member, Supervisory Committee

Dylan Mikesell, Ph.D. Member, Supervisory Committee

The final reading approval of the thesis was granted by Ellyn M. Enderlin, Ph.D., Chair of the Supervisory Committee. The thesis was approved by the Graduate College.

DEDICATION

This thesis is dedicated to my grandfather, Peter Atkin.

ACKNOWLEDGMENTS

I'd like to thank my advisor, Dr. Ellyn Enderlin, and my committee members Drs. HP Marshall and Dylan Mikesell for their patience and support as teachers, mentors, and critics. I'd also like to thank my family members in Idaho and in Indiana for their support during my academic career. Finally, I'd like to thank my partner for her unwavering support and patience during the thesis-writing process.

ABSTRACT

Seasonal snowpack accounts for ~70% of the water supply in the western United States, and measuring snow accumulation and ablation remotely has long been a stated goal of NASA. The 2018 launch of ICESat-2, a spaceborne Lidar system, has offered unparalleled spatial and temporal coverage of mountainous terrain with the potential for unprecedented vertical accuracy. Data from ICESat-2 are used to measure seasonal snow depths using the level-3A ATL08 (land and canopy elevation) product for the Reynolds Creek Experimental Watershed in southwest Idaho and the ATL06 (land ice elevation) product for Wolverine Creek in the Kenai Mountains of Alaska. The methodology for coregistering ICESat-2 transects to reference digital terrain models then estimating snow depths as the difference between the ICESat-2 and reference elevations is described. Median and MAD snow depths for transects from 2019 and 2020 are 3.1 +/- 6.7m at Reynolds Creek EW and are 5.5 +/- 2.1m at Wolverine glacier. Here we find that measuring snow depths using ICESat-2 is crude in variable, vegetated terrain covered by the ATL08 data product, and that there is not a strong relationship between the residual values reported at Reynolds Creek EW and terrain parameters such as slope, aspect, vegetative coverage, and elevation. We do find that the ATL06 analysis results in reasonable first-order estimates of snow depth but that the evolution of the glacier surface elevations must be more accurately constrained in order to ensure the snow depth estimates are unbiased.

TABLE OF CONTENTS

DEDICATION	iv
ACKNOWLEDGMENTS	v
ABSTRACT	vi
LIST OF TABLES	ix
LIST OF FIGURES	x
LIST OF ABBREVIATIONS	xiii
CHAPTER ONE: SNOW DEPTHS AND ICESAT-2.....	1
I. Introduction	1
II. Background	4
III. Methods	7
A. Study Areas	7
B. Reference Elevations.....	10
C. Coregistration and Snow Depth Estimation	11
IV. Results	18
A. Reynolds Creek Experimental Watershed	18
B. Wolverine Glacier	25
V. Discussion.....	32
A. Error Quantification.....	32
B. Reynolds Creek Experimental Watershed and the ATL08 Product	34

C. Wolverine Glacier and the ATL06 Product	38
VI. Conclusions	39
REFERENCES	41
APPENDIX	47

LIST OF TABLES

Table 1.1. Reynolds Creek SNOTEL Site 2029 data for the month of January 2020.
.....37

LIST OF FIGURES

Figure 1.1.	Wolverine Glacier in coastal Alaska (A) and Reynolds Creek Experimental Watershed in southwestern Idaho (B).	3
Figure 1.2.	An artist’s depiction of the ICESat-2 system orbiting above sea ice (Magruder et al., 2019). Image is not to scale.	4
Figure 1.3.	Reynolds Creek EW in southwestern Idaho (outlet coordinates 43.3241, -116.6858). The white triangles mark the locations of Reynolds Creek Critical Zone Observatory instruments. Elevations and seasons of 47 available ICESat-2 transects are shown in panel a. The base map is the digital terrain model (elevations in legend). The location of the watershed in relation to the Treasure Valley area is shown in panel b, and in relation to Idaho in panel c.....	9
Figure 1.4.	Wolverine glacier on the Kenai peninsula near Anchorage, AK, USA (outlet coordinates 60.383, -148.918). The locations and seasons of 42 available ICESat-2 transects are shown in panel a. The approximate glacier extent is from the most recent available GLIMS data. The base map is the digital terrain model from September 2018 Lidar data (elevations in legend). Wolverine glacier’s location in relation to the Kenai Peninsula is shown in panel b, and to Alaska in panel c.....	10
Figure 1.5.	An example of the coregistration process for ATL08 transects for an August 26, 2019 ICESat-2 pass at Reynolds Creek EW. Panel (a) shows the raw ATL08 elevation transect in red and the reference DTM transect in black. Panel (b) shows the same elevation transects after horizontal coregistration of the ATL08 data. For this transect, the optimal northing shift was 58.4m and the optimal easting shift was -81.9m. Coordinates are referenced to UTM Zone 11N.	13
Figure 1.6.	Two transects from Wolverine Glacier and corresponding reference elevations. A representative summer transect from 26 August 2019 is shown in (a); a representative winter transect from 26 February 2019 is shown in (b). The vertical error bars are the respective reported vertical geolocation errors.....	16

Figure 1.7.	Scatterplot of Wolverine glacier elevations from 2008 (x-axis) and 2018 (y-axis). The red line represents a line of no elevation change over the period, and points below that line (the area shaded grey) indicate that a negative elevation change. The average annual rate of change in elevation between 2008 and 2018 is -1.1 m a^{-1} with a standard deviation of 0.03 m a^{-1}	17
Figure 1.8.	Histograms of coregistration offsets in the easting (left column; a, c, and e) and northing (right column; b, d, and f) directions extracted from snow-free regions for the Reynolds Creek EW transects. The top row (a, b) shows offsets for all 47 transects. The middle row (c, d) shows offsets segregated by the orientation of the satellite (reverse or forward), and the bottom row shows the offsets segregated by beam strength (strong or weak).	19
Figure 1.9.	Histogram of all footprint elevation differences between ATL08 and reference DTM elevations at Reynolds Creek EW after horizontal coregistration. The histogram includes 5528 ATL08 mean footprint elevations from 2018-2020.	20
Figure 1.10.	Histogram of elevation differences (residual values) after coregistration for 47 ICESat-2 transects totaling 5528 mean footprint elevations at Reynolds Creek EW in 2019 and 2020. Histograms are shown for summer (a), autumn (b), winter (c), and spring (d). Seasons are defined as summer (July – September), autumn (October – December), winter (January – March), and spring (April – June).	21
Figure 1.11.	Plots of measured elevation difference as a function of reference elevation above sea level for ATL08 transects at Reynolds Creek EW for (a) winter, (b) spring, (c) summer, and (d) autumn.	22
Figure 1.12.	Plots of measured elevation difference as a function of slope for ATL08 transects at Reynolds Creek EW for (a) winter, (b) spring, (c) summer, and (d) autumn.	23
Figure 1.13.	Plots of measured elevation difference as a function of aspect for ATL08 transects at Reynolds Creek EW for (a) winter, (b) spring, (c) summer, and (d) autumn.	24
Figure 1.14.	Histogram of elevation differences (residual values) as a function of vegetation height at Reynolds Creek EW for all 5528 sample points.	25

Figure 1.15. Histograms of elevation differences (residual values) after coregistration for 42 ATL06 transects totaling 5110 mean footprint elevations at Wolverine glacier in 2019 and 2020. The seasons are defined as winter (January – March), spring (April – June), summer (July – September), and autumn (October – December). 27

Figure 1.16. Estimated snow depths (residual values) at Wolverine glacier for winter (January-March) and spring (April – June) of 2018-2020. Residual values at each point along the transects are interpreted as estimated snow depths. 28

Figure 1.17. Plots of measured elevation difference as a function of reference elevation above sea level for ATL06 transects at Wolverine glacier for (a) winter, (b) spring, (c) summer, and (d) autumn. 29

Figure 1.18. Plots of measured elevation difference as a function of slope for ATL06 transects at Wolverine glacier for (a) winter, (b) spring, (c) summer, and (d) autumn. 30

Figure 1.19. Plots of measured elevation difference as a function of aspect for ATL06 transects at Wolverine glacier for (a) winter, (b) spring, (c) summer, and (d) autumn. 31

Figure A.1. A simplified flowchart of code steps used for coregistration of ICESat-2 transects. For a complete user guide, please contact the author Colten Elkin at coltenelkin@u.boisestate.edu. 67

LIST OF ABBREVIATIONS

ATL06	Land Ice Elevation data product
ATL08	Land and Vegetation Elevation data product
ATLAS	Advanced Topographic Laser Altimeter System
ASL	Above Sea Level
CRREL	Cold Regions Research and Engineering Laboratory
DTM	Digital Terrain Model
EW	Experimental Watershed
GLIMS	Global Land Ice Measurement from Space
ICESat-2	Ice, Cloud, and land Elevation Satellite 2
MAD	Median Absolute Deviation
RMSD	Root Mean Squared Difference
SNOTEL	Snow Telemetry
SWE	Snow Water Equivalent
USGS	United States Geological Survey

CHAPTER ONE: SNOW DEPTHS AND ICESAT-2

I. Introduction

Measuring the seasonal snowpack remotely has long been desired by the water resources community (Fiebrich, 2009). The Western United States generally receives about 70% of its water as seasonal snowpack (Barnett et al., 2005), and the year-to-year variability in streamflow associated with snow accumulation is critically important to water resources in this region. Globally, more than one billion people rely on water at least partially supplied by seasonal snow or glacial melt (Beniston et al., 2003; Barnett et al., 2005), but in situ measurements of snow water equivalent (SWE) are expensive and notoriously spatially and temporally variable (Anderton et al., 2004; Grunewald et al., 2010). In glacierized regions, seasonal snow influences both annual water resources as well as the mass balance of glaciers (Huss and Hock, 2018). Changes in the mass balance of mountain glaciers also negatively impacts local biology, economy, society, and culture (Archer et al., 2010; Mishra et al., 2018; O'Neel et al., 2015).

Various satellite systems and methodologies have been applied to the problem of measuring snow depth and SWE, but challenges remain with accuracy and resolution. One of the most recent remote sensing approaches relies on Lidar (light detection and ranging) and visible light cameras mounted to aircraft, data from which are used to generate outputs such as structure from motion (SfM) and dense point clouds representing a three-dimensional map of the imaged area (Kelly and Belmont, 2018; Kim

et al., 2017). After removing the baseline snow-free ground surface elevations, elevation time series can be converted to time series of snow depth.

Although aerial Lidar can yield highly accurate and spatially-detailed estimates of snow depth, Lidar coverage is spatially and temporally sparse. Perhaps the most pertinent example of using Lidar to monitor snow depths is the NASA Aerial Snow Observatory (ASO) in the Tuolumne River basin in California (Brandt et al., 2020; Pflung & Lundquist, 2020). While successful in the Tuolumne, the price of repeat Lidar observations over a large ($> 1000\text{km}^2$) watershed is prohibitive, and Lidar observations are available for only a fraction of the Rocky Mountains in the US. These limitations to Lidar, price and otherwise, of targeted aerial Lidar can potentially be overcome by the Ice, Cloud, and Land Satellite 2 (ICESat-2) program (Rebold et al., 2019). Here we use two level-3A Lidar products from the ICESat-2 program: ATL08 and ATL06. The ICESat-2 land and vegetation height (ATL08) data product has a 100 m by ~ 11 m footprint and estimated vertical accuracy of 40cm in sparsely vegetated terrain (Neuenschwander and Pitts, 2019), and the land ice height (ATL06) product has a 40 m by ~ 11 m footprint and estimated vertical accuracy of 10cm (Smith et al., 2019). The spatial resolution and estimated errors of these data products are small enough that they could be reasonably used to measure snow depths on the order of one or more meters. Here we assess whether the higher-order ICESat-2 data products ATL06 and ATL08 can be used to estimate seasonal snowpack in remote, mountainous terrain. This pilot study focuses on two watersheds with dense in situ observational coverage: Wolverine glacier, a part of the Wolverine Creek drainage in Alaska, USA (outlet coordinates 60.383, -148.918) and Reynolds Creek Experimental Watershed in Idaho, USA (outlet coordinates

43.3241, -116.6858) (figure 1.1). While snow blankets the terrain, the difference between ICESat-2 elevations and reference snow-free elevations yields estimated snow depth. We compare preliminary snow depth estimates with *in situ* data for the watersheds of interest to assess the utility of ICESat-2 to accurately map snow depth in different terrain types, discuss current methodological limitations, and propose future methodological improvements.



Figure 1.1. Wolverine Glacier in coastal Alaska (A) and Reynolds Creek Experimental Watershed in southwestern Idaho (B).

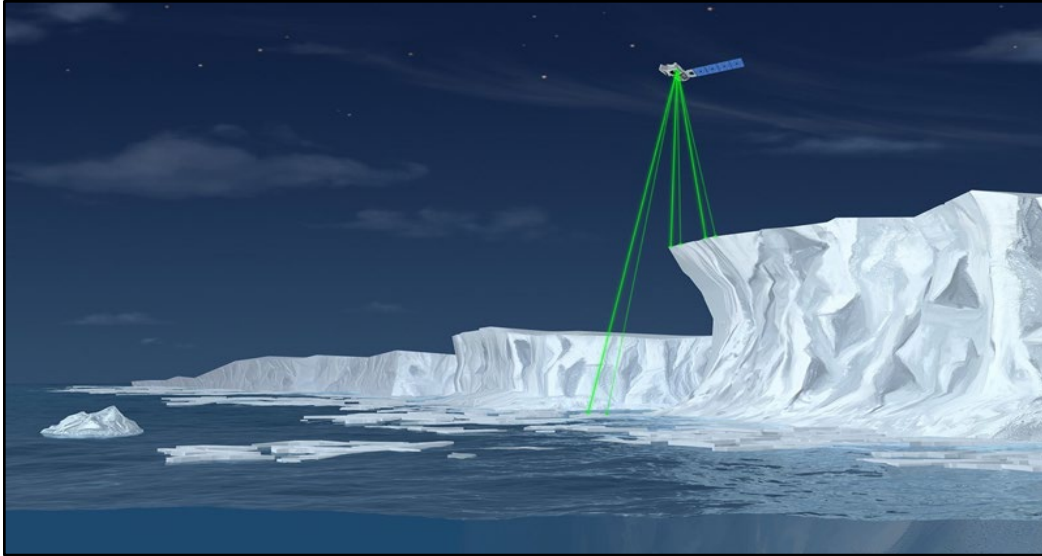


Figure 1.2. An artist's depiction of the ICESat-2 system orbiting above sea ice (Magruder et al., 2019). Image is not to scale.

II. Background

In many ways, water resource engineering has been the driving force behind the study of hydrology and snowpack in the United States. From the first gages on the Colorado and Columbia Rivers to the constant expansion of the National Resource Conservation Service snow telemetry (SNOTEL) network (NRCS, 2012), there is considerable investment in placing observational constraints on mountainous snowpack each winter and the rate at which it melts out in the spring and summer (Fiebrich, 2009). Despite the development of numerous methods to broadly constrain the water equivalent of melted snow (i.e., snow water equivalent or SWE) using satellite data, we still lack the ability to remotely sense SWE from space. Early Landsat imagery was used to map the spatial extent of snow coverage (Itten, 1975; Orheim and Lucchitta, 1987), but the data lacked depth and density measurements and proved generally a poor approximation for snow depths and SWE. Passive satellite radar observations are promising in theory since microwave emissions from the Earth's surface should vary with snow depth (Rango et

al., 1989; Rott et al., 2009; Pettinato et al., 2011; Lievens et al., 2019), but these measurements have thus far failed to live up to the promise of basin-scale measures of snow water equivalent, not the least of the reasons being their generally poor spatial resolution. Repeat airborne Lidar has been used extensively as part of NASA's SnowEx program, and other observational efforts, to accurately measure snow depths over several square kilometers (Kim et al., 2017; Brucker et al., 2018). Lidar-derived snow depths are combined with in situ measures of snow density and simple linear relationships between terrain parameters to estimate snow water equivalent over a large area. The weaknesses of this approach, however, are the cost and temporal resolution: the flights are too expensive to be practical in large watersheds ($> 1000 \text{ km}^2$) without dedicated state or national funding and generally only capture peak snowpack rather than intraseasonal variability.

Launched in 2018, ICESat-2 improves significantly on the spatial coverage and temporal resolution of aerial Lidar snow depth estimates. While not a primary mission objective of ICESat-2, application of these high-resolution Lidar datasets to snow depth estimation across variable terrain would be a significant benefit to the water resource community. ICESat-2's onboard Advanced Topographic Laser Altimeter System (ATLAS) instrument is a six-beam laser altimeter that records the roundtrip times of photons and, using the satellite's position and attitude, calculates the elevation of the surface below (figure 1.2). The six beams are separated into three beam pairs that consist of a strong beam and a weak beam; the strong beams transmit four times as many photons per shot as the weak beams. The interpair distance is $\sim 3\text{km}$ and the intrapair distance is $\sim 90\text{m}$, enabling the computation of surface slopes in gentle terrain from a single pass and

more precise slope adjustments to ice sheets and sea ice elevation time series. The ground footprints of the shots are ~ 11 m in diameter and, at a shot frequency of 10,000 Hz, the shots allow for surface measurements every ~ 0.9 m along track (Markus et al., 2017). In polar regions the satellite track is repeated every 91 days, but below the polar latitudes the pointing geometry is tilted with each successive orbit to optimize spatial coverage in non-polar latitudes (Neuenschwander and Pitts, 2019).

After collection, data are preprocessed into various products made available to the public. The ATL03 dataset includes data for all individual photons, with the exception of background photons filtered using an elevation threshold and those flagged as clouds, and is a dense and unwieldy package. A typical ATL03 segment spanning the western US is about one gigabyte of data. To increase user accessibility, ATL03 data are further processed into level-3A data products which report elevations for a swath of photon returns. The level-3A products used herein are ATL08, which measures land and canopy elevations every 100m along track, and ATL06, which measures land ice elevations every 40m along track with 50% overlap between successive measurements. A typical ATL08 or ATL06 segment spanning the western US is only about 50 megabytes, or about one-twentieth the size of the ATL03 product.

Although ICESat-2's pointing geometry is adjusted so that reference ground tracks are offset outside the polar regions, meaning they do not collect repeat acquisitions of the same swath, many watersheds in middle and high latitudes are overlapped by multiple reference ground tracks during each 91-day repeat cycle. Since multiple reference ground tracks will intersect large mid-latitude watersheds and each satellite pass contains data from six beams, there is the potential for dense spatial coverage of

snow depth transects from ICESat-2. The satellite's laser altimeter is modeled to be accurate to within 40 cm for unvegetated or brushy terrain and within 2 m for complex, highly vegetated terrain (Neuenschwander and Pitts, 2019). Thus, there is the potential for highly accurate snow depth transects in sparsely vegetated mountain regions and first-order estimates of snow depth in vegetated terrain. These transects of snow depths could be used as inputs for snow energy balance models to better constrain measurements of snow depth and SWE across watersheds.

III. Methods

A. Study Areas

ICESat-2 data were gathered for both Reynolds Creek Experimental Watershed (EW) in southwestern Idaho and Wolverine Glacier on the Kenai Peninsula of Alaska. These two study sites were selected because they have extensive in situ observations from strikingly different climate regimes, which allows us to assess the detection threshold of our satellite-derived snow depths in different geographic areas and precipitation regimes. The areas of interest, described below, are fairly typical of watersheds in their geographic region and span an array of slopes, aspects, and, for Reynolds Creek EW, vegetation types and densities.

Reynolds Creek EW spans about 240 km² and slopes from the southern highlands to its outlet at the Snake River (figure 1.3); a full physical and biological description can be found in Seyfried et al. (2001). Reynolds Creek flows northwards from southern highlands near Slack Mountain to the Snake River Plain before joining the Snake River near Walters Ferry, Idaho. Located in the rain shadow of the Cascade Mountains, the in situ data for Reynolds Creek between 2010 and 2019 suggest the typical maximum snow

depth is 0.47 m at the SNOTEL site at 1700 m (SNOTEL data for Reynolds Creek site 2029). The watershed is an experimental watershed and has a long history of data collection and scientific endeavor (Hanson, 2001; Ilangakoon et al., 2016; Seyfried et al., 2018). Importantly, Reynolds Creek EW has a 2007 summertime Lidar data set that we use as our reference elevations for comparison with the ATL08 data (Idaho Lidar Consortium, 2007). Reynolds Creek EW is also the focus of US Agricultural Research Service snow model tuning and therefore contains a variety of snow courses and in situ instrumentation (Baffaut et al., 2020). While in situ observations may not be coincident in space or time with ICESat-2 transects, the data are valuable for validation of ICESat-2 derived snow depths and snow depth gradients.

Wolverine Glacier in Alaska is a USGS Benchmark Glacier. Wolverine Glacier also has a long data and scientific history as well as a long time series of mass balance fluxes (McNeil et al., 2019; O’Neel et al., 2019; Zeller et al., 2020). Rasterized elevations from the USGS Alaska Science Center’s Geodetic Data repository are used as our reference elevations for comparison with the ATL06 data (figure 1.4) (McNeil et al., 2019). Low in elevation and close to the north Pacific, Wolverine glacier is heavily influenced by the cool, moist, maritime climate typical of southeastern Alaska. The total ice-covered area is $\sim 15.6 \text{ km}^2$, and the equilibrium line lies at $\sim 1235 \text{ m}$ above mean sea level (Zeller et al., 2020). Located in a maritime polar setting, the in situ data from 2007-2017 suggest the typical maximum annual snow depth is 6.56 m (Baker et al., 2018).

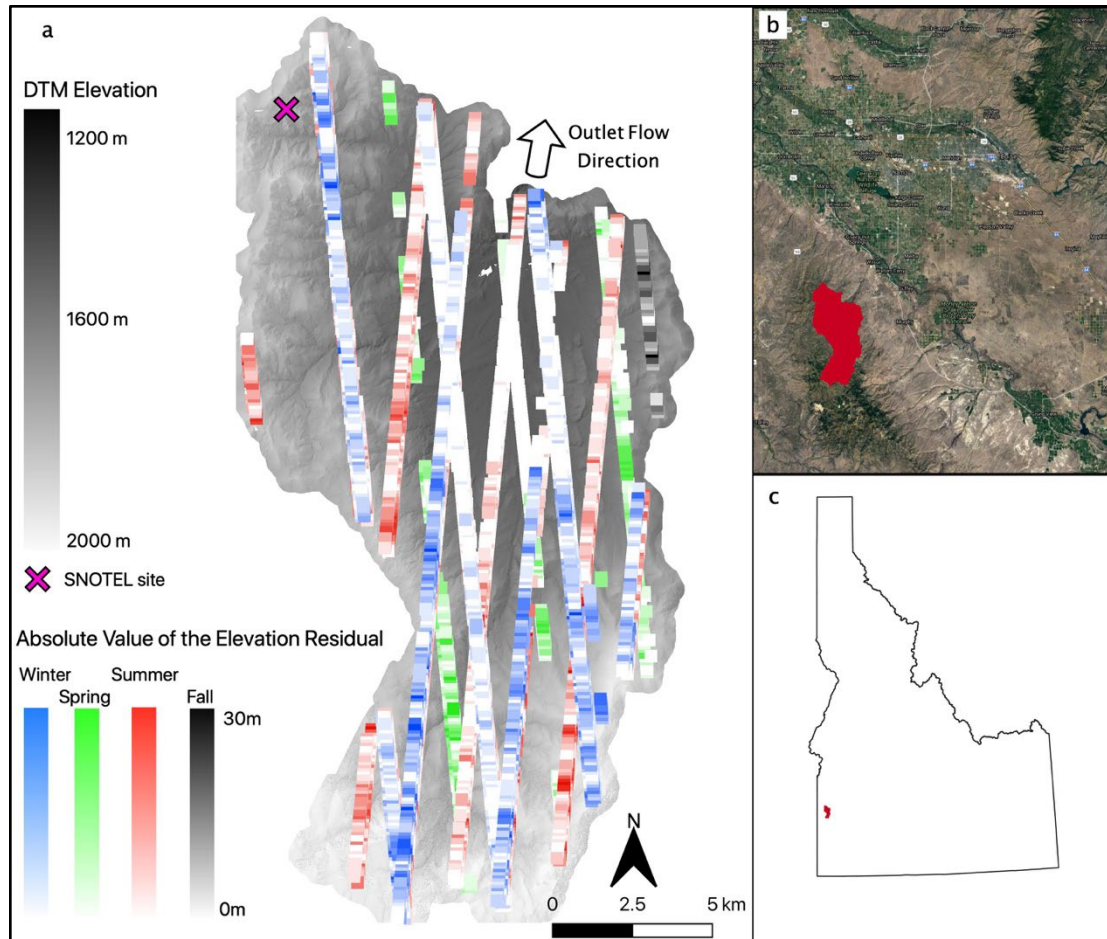


Figure 1.3. Reynolds Creek EW in southwestern Idaho (outlet coordinates 43.3241, -116.6858). The white triangles mark the locations of Reynolds Creek Critical Zone Observatory instruments. Elevations and seasons of 47 available ICESat-2 transects are shown in panel a. The base map is the digital terrain model (elevations in legend). The location of the watershed in relation to the Treasure Valley area is shown in panel b, and in relation to Idaho in panel c.

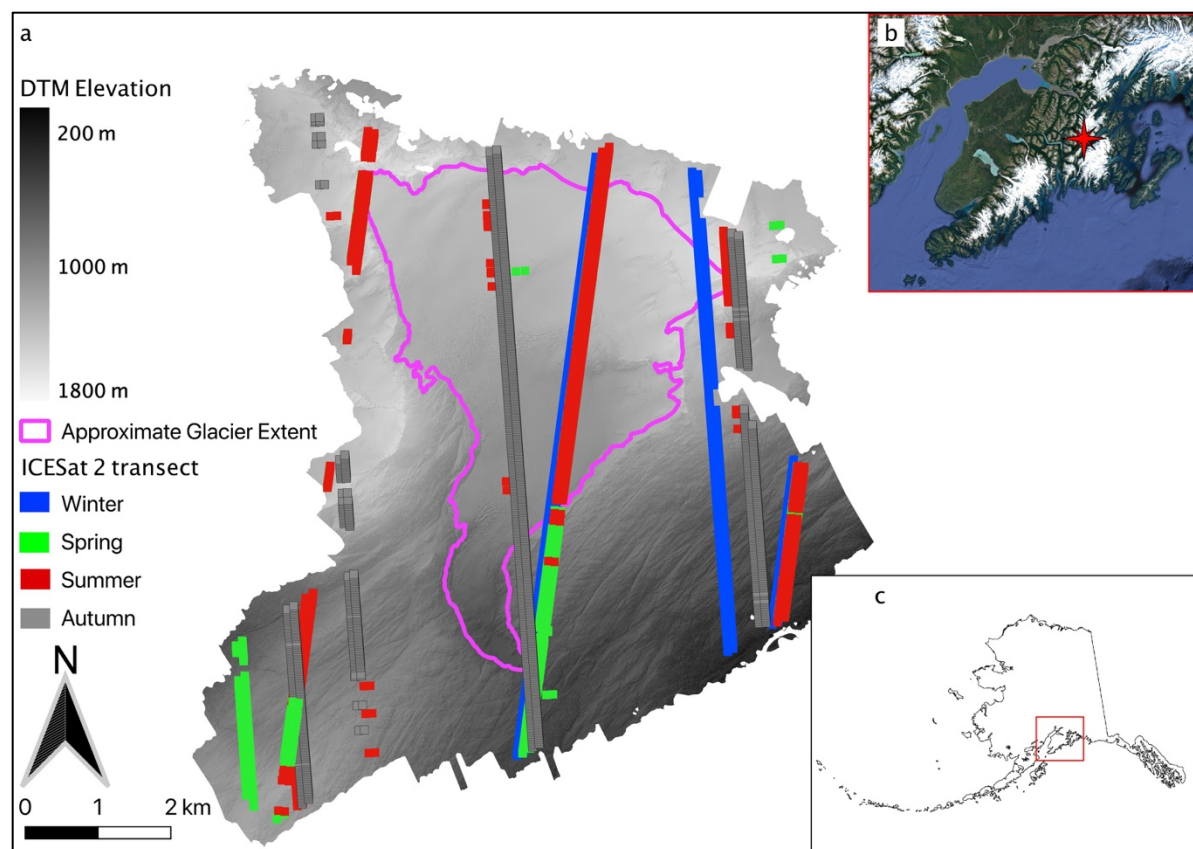


Figure 1.4. Wolverine glacier on the Kenai peninsula near Anchorage, AK, USA (outlet coordinates 60.383, -148.918). The locations and seasons of 42 available ICESat-2 transects are shown in panel a. The approximate glacier extent is from the most recent available GLIMS data. The base map is the digital terrain model from September 2018 Lidar data (elevations in legend). Wolverine glacier's location in relation to the Kenai Peninsula is shown in panel b, and to Alaska in panel c.

B. Reference Elevations

For low-slope, unglaciated polar regions, ground elevation time series can be differenced over time to construct time series of snow depth (Hall et al., 2001; Hislop et al., 2018). However, the non-repeat footprints of ICESat-2 below polar latitudes necessitates the use of independent, highly accurate reference (i.e., snow-off) datasets to construct time series of snow depth via elevation differencing.

For the Reynolds Creek EW, the reference elevation dataset contained >4 billion irregularly-spaced discrete Lidar point elevation observations. The Lidar data were down-

sampled to a 2 m-resolution raster grid to create a digital terrain model (DTM) that excludes vegetation cover (see `LASReadLastReturn.m`, Appendix A; Alexander, 2021). Reported errors from the Lidar data are presented in the Discussion section. The rasterized elevations were cropped to the outline of Reynolds Creek EW as defined using a surface flow path algorithm (see Watershed Delineation, Appendix A.E) with the Reynolds Creek's outlet into the Snake River as the outlet point. Finally, because the 2007 Lidar flight didn't cover the lowest ~ 20 km² of Reynolds Creek EW, the polygon was further clipped to only area covered by Lidar data.

A Lidar-derived 0.5 m-resolution DTM from early September 2018 was used as reference elevation dataset for Wolverine Glacier (McNeil et al., 2019). Wolverine Glacier's boundary was defined with the most recent Global Land Ice Measurement from Space (GLIMS) polygon available through the USGS (McNeil et al., 2019). Elevations that lay within the GLIMS polygon are subject to both movement of ice and accumulation or ablation of snow. We dealt with these shifting reference elevation issues with a process described in section 3.B.ii, below. We used the entire area covered by the Lidar dataset for coregistration, which includes some 20 km² of additional unglaciated area adjacent to Wolverine Glacier (figure 1.4).

C. Coregistration and Snow Depth Estimation

In order to estimate snow depths using ICESat-2 data, all ICESat-2 transects had to be precisely coregistered to their respective reference elevation DTMs. We considered the reference DTMs to be true elevations and individually coregistered each ICESat-2 transect to the DTMs. Both the ATL08 and ATL06 products include different elevations at each footprint within the transect; these different elevations are the mean, median, and

interpolated elevation values. In order to remain consistent, we considered only the mean elevation reported by ICESat-2, and we compared it to the mean reference elevation for the corresponding footprint within the transect. A gradient descent approach (see `coregister_icesat2.m` script included in Appendix A) was applied to efficiently identify and minimize the difference in elevation values between ICESat-2 and the reference DTMs. Each ICESat-2 segment was coregistered as a transect, not point-by-point, with the assumption that any error in coregistration was approximately uniform along the transect. The means were first shifted in the x (easting) and y (northing) directions until the minimized the root mean squared difference (RMSD) between ICESat-2 and reference elevations was found. An example ATL08 elevation transect is shown with the corresponding reference DTM elevation transect prior to and following horizontal coregistration in figure 1.5. The process was repeated for each snow-free (i.e., summer) ATL08 track in the Reynolds Creek EW. The method was refined, as described in the subsections below, for all Reynolds Creek EW ATL08 transects that potentially contained snow (i.e., October-June transects) so that only snow-free regions were used for coregistration. At Wolverine glacier, footprints inside the GLIMS polygon were excluded from the coregistration process because glacier mass loss and flow cause the glacier surface elevation to change over time and coregistration should only be performed using static terrain.

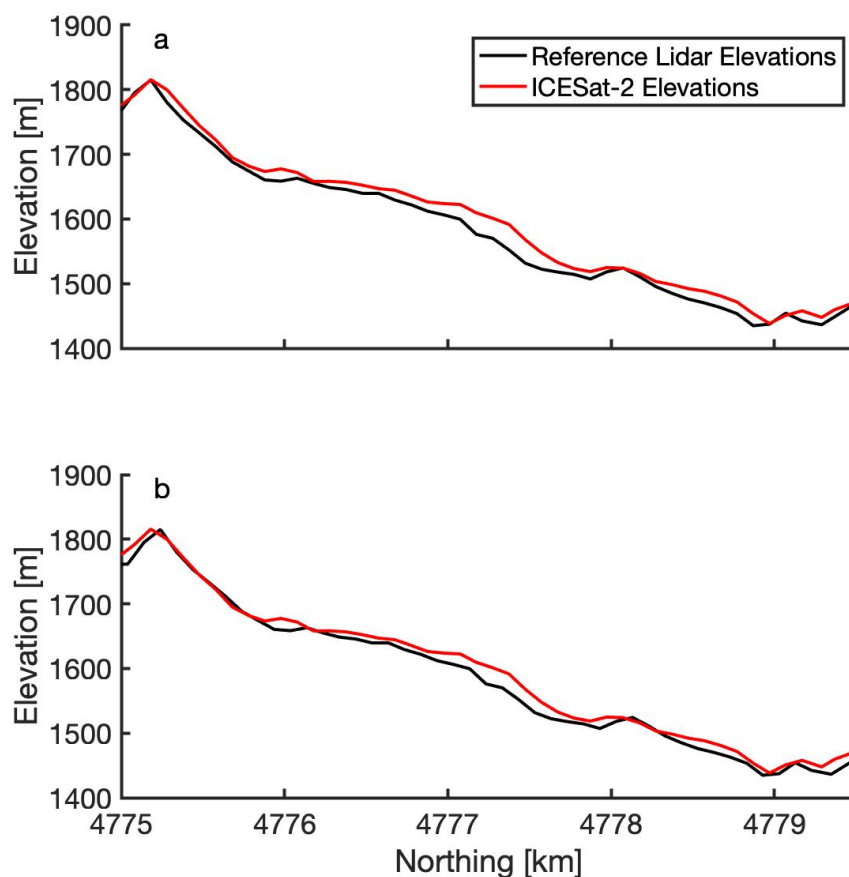


Figure 1.5. An example of the coregistration process for ATL08 transects for an August 26, 2019 ICESat-2 pass at Reynolds Creek EW. Panel (a) shows the raw ATL08 elevation transect in red and the reference DTM transect in black. Panel (b) shows the same elevation transects after horizontal coregistration of the ATL08 data. For this transect, the optimal northing shift was 58.4m and the optimal easting shift was -81.9m. Coordinates are referenced to UTM Zone 11N.

i. ATL08

For the ATL08 data product, photon return statistics (i.e., mean and median elevations, number of returns, etc.) were reported for 100 m-long by ~11 m-wide footprints oriented along the track of the beam transect. Elevations for the ground surface and for the canopy were reported separately.

During the winter, only snow-free surfaces should be used for coregistration. Here we used the ATL08 brightness flag to identify snow-free portions of each transect. The

brightness flag uses a radiometric approach to identify regions of snow and ice: if the average number of return photons is >3 per shot for the strong beam, then the brightness flag records that the surface below is highly reflective, which generally indicated snow or ice. For reference, the average number of return photons expected over normal terrain was one per shot for the strong beam and a quarter photon per shot for the weak beam (Neuenschwander & Pitts, 2019), with ~ 110 shots and therefore ~ 110 photons per strong beam footprint and ~ 30 photons per weak beam footprint. The brightness flag was used to select non-flagged (snow-free) areas for horizontal coregistration throughout the year for consistency, with the expectation that summer transects will contain zero brightness flags and the number and distribution of flagged regions will vary with snow cover throughout the rest of the year. We repeated the coregistration process for each of the 47 ATL08 transects that passed over Reynolds Creek EW.

The use of the brightness flag during registration should ensure that only snow-free regions are included in the coregistration process, enabling transect-specific horizontal and vertical coregistration. However, to account for the potential failure of the brightness flag, we used only summer transect observations to vertically coregister all transects. During the summer, coregistration should be highly accurate (i.e., the differences between reference and ICESat-2 elevations should be centered at zero meters) if both the reference DTM and the ICESat-2 transects include accurate estimates of snow-free, ground elevations and not vegetation. All transects were shifted vertically so that the summer difference median was forced to 0 m.

ii. ATL06

For the ATL06 data product, photon return statistics (i.e., mean and median elevations, number of returns, etc.) are reported for 40 m-long x ~11 m-wide footprint oriented along the track of the beam transect. These footprints overlap each other by 50%, so that ATL06 reports a mean elevation every 20 m along the transect. However, because the ATL06 data product is designed for land ice where most of the terrain will be highly reflective, there is no corresponding brightness flag that can be used for coregistration. Additionally, because ice can move both laterally and vertically, and because glaciers can gain or lose surface elevation due to mass change, glacierized regions cannot be used for coregistration. Therefore, we used only the transect elevations outside of the GLIMS polygon outline (i.e., outside the glacier extent) as static reference elevations to coregister transects.

Coregistration was only performed for transects with non-glacierized elevation residuals that exceeded the reported vertical uncertainties for ATL06 (i.e., transects with significantly different elevations than the reference DTM), which were indicative of poor horizontal and/or vertical coregistration. Figure 1.6 shows representative transects for a summer (figure 1.6a) and winter (figure 1.6b) over Wolverine glacier. For the summer transects, a gradient descent approach was used to identify the horizontal offsets required to minimize the root mean squared difference (RMSD) between ICESat-2 and reference elevations. Since there is no brightness flag to identify snow-free regions in the ATL06 dataset, we did not attempt to coregister non-summer (i.e., October-June) transects. As with the ATL08 data, all transects were vertically shifted using the median elevation difference from summer transects.

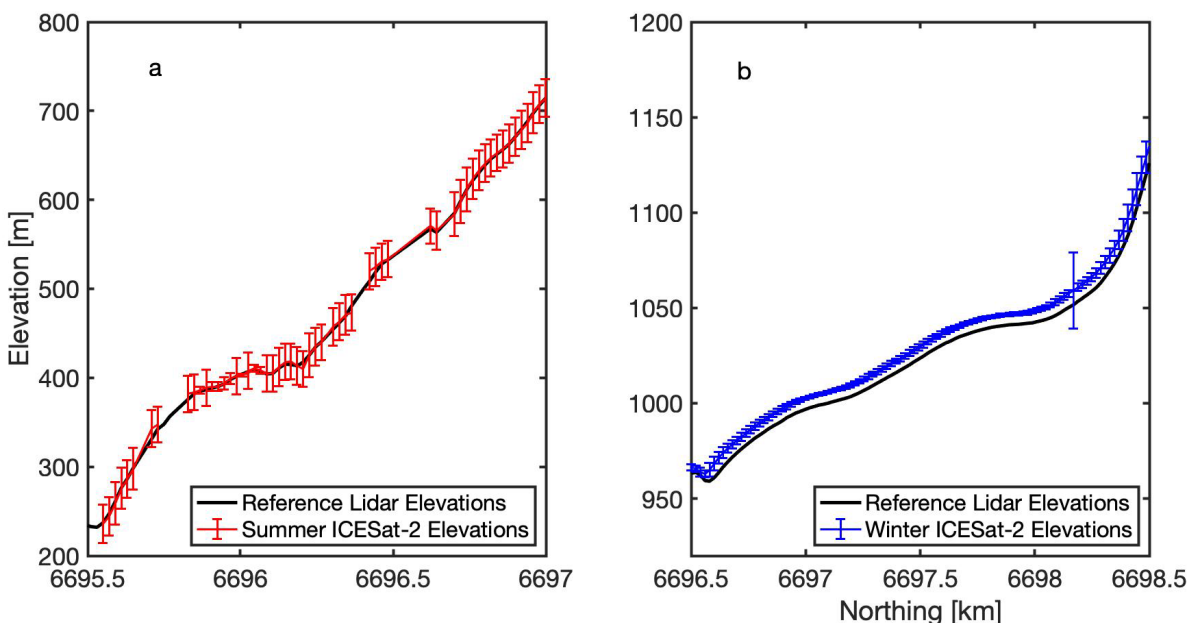


Figure 1.6. Two transects from Wolverine Glacier and corresponding reference elevations. A representative summer transect from 26 August 2019 is shown in (a); a representative winter transect from 26 February 2019 is shown in (b). The vertical error bars are the respective reported vertical geolocation errors.

Estimation of on-glacier snow depths from elevation residuals is complicated by the evolution of the glacier's surface over time. Inside of the GLIMS polygon, several geodetic DTMs for the glacier were used to approximate the annual end-of-melt-season (i.e. September) glacier surface elevation and assess uncertainties. Geodetic DTMs for Wolverine Glacier for 2008 and 2018 were obtained from the Benchmark Glacier data portal (McNeil et al., 2019). Each DTM was clipped to the GLIMS polygon and the elevations were bilinearly interpolated to the grid of the 2018 reference DTM. The end-of-melt-season glacier surface elevations for 2019 and 2020 were extrapolated assuming a constant rate of elevation change from 2008-2020, using the elevation change between the 2008 and 2018 DTMs (figure 1.7). The ATL08 vertical residuals presented in the following sections are with respect to the preceding end-of-melt-season elevations. These

residuals represent temporal changes in the elevation of the glacier surface elevation due to seasonal snow accumulation as well as summer snow and ice melt. Potential biases in these estimated due to intra-annual variability in glacier mass change and flow are discussed below.

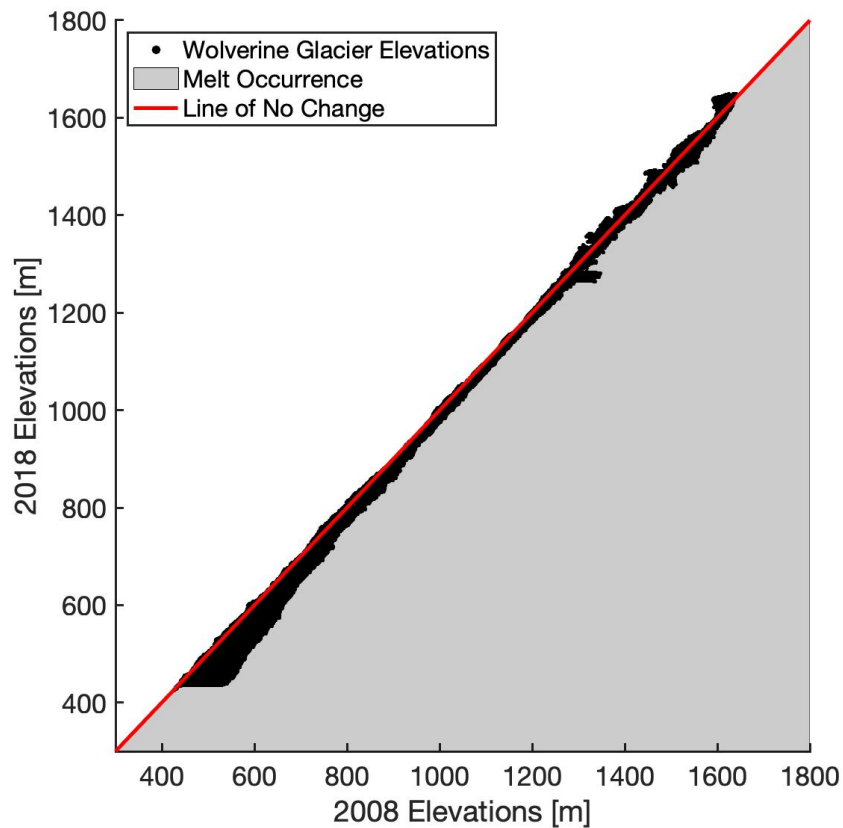


Figure 1.7. Scatterplot of Wolverine glacier elevations from 2008 (x-axis) and 2018 (y-axis). The red line represents a line of no elevation change over the period, and points below that line (the area shaded grey) indicate that a negative elevation change. The average annual rate of change in elevation between 2008 and 2018 is -1.1 m a^{-1} with a standard deviation of 0.03 m a^{-1} .

IV. Results

A. Reynolds Creek Experimental Watershed

For Reynolds Creek, our coregistration process yielded median \pm median of absolute deviation (MAD) easting and northing offsets of 6.12 ± 46.9 m and 33.5 ± 19.6 m, respectively. We find no obvious pattern in coregistration offsets in the easting and northing directions for either satellite orientation or beam strength (figure 1.8).

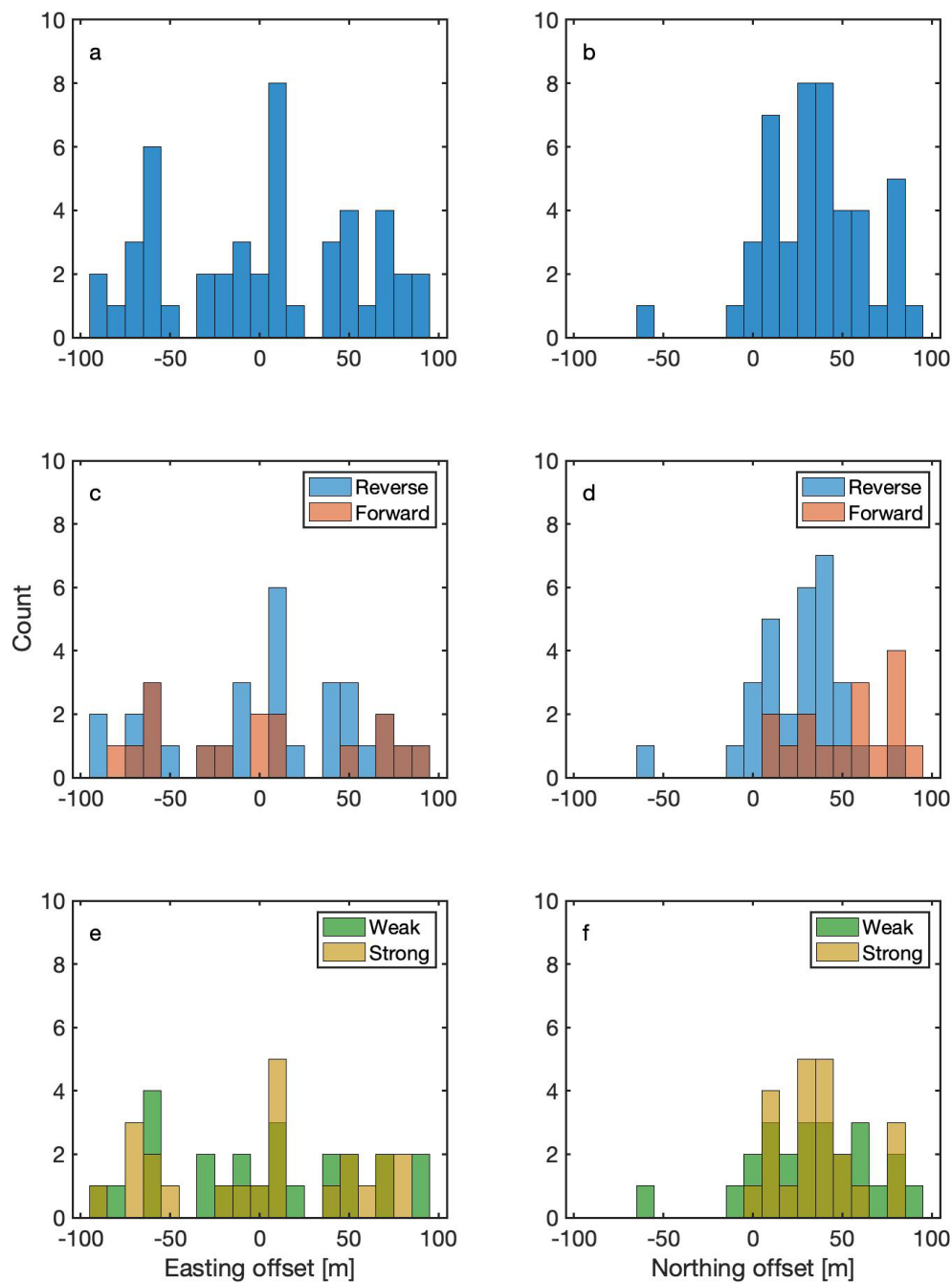


Figure 1.8. Histograms of coregistration offsets in the easting (left column; a, c, and e) and northing (right column; b, d, and f) directions extracted from snow-free regions for the Reynolds Creek EW transects. The top row (a, b) shows offsets for all 47 transects. The middle row (c, d) shows offsets segregated by the orientation of the satellite (reverse or forward), and the bottom row shows the offsets segregated by beam strength (strong or weak).

After horizontal coregistration, the summer transects had a median RMSD +/- MAD of 2.3 +/- 4.7 m (figure 1.9), suggesting a slight positive bias in ATL08 elevations with respect to the reference DTM. The data are not normally distributed, but exhibit right skew. Therefore, each ICESat-2 transect was shifted vertically downward 2.3 m to eliminate elevation bias. The final median \pm MAD elevation residuals for the summer (July – September), autumn (October – December), winter (January – March), and spring (April – June) are 0 ± 6.6 m, -0.7 ± 5.7 m, 3.1 ± 6.7 m, and 2.5 ± 7.1 m, respectively (figure 1.10). The distributions of elevation residuals for each season are statistically different from each other at $P = 0.05$ with the exception of spring and winter. Elevation differences are plotted as a function of vegetation height (figure 1.11), and we find no discernable relationship between vegetation height and residual values.

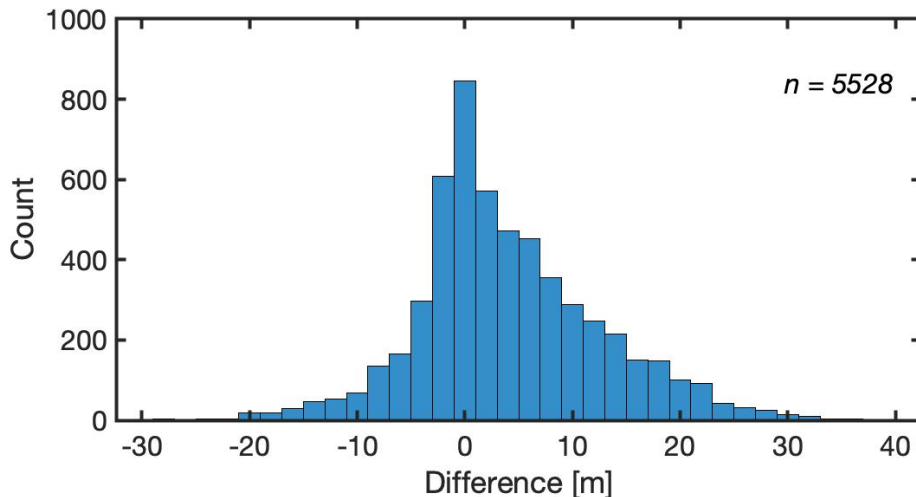


Figure 1.9. Histogram of all footprint elevation differences between ATL08 and reference DTM elevations at Reynolds Creek EW after horizontal coregistration. The histogram includes 5528 ATL08 mean footprint elevations from 2018-2020.

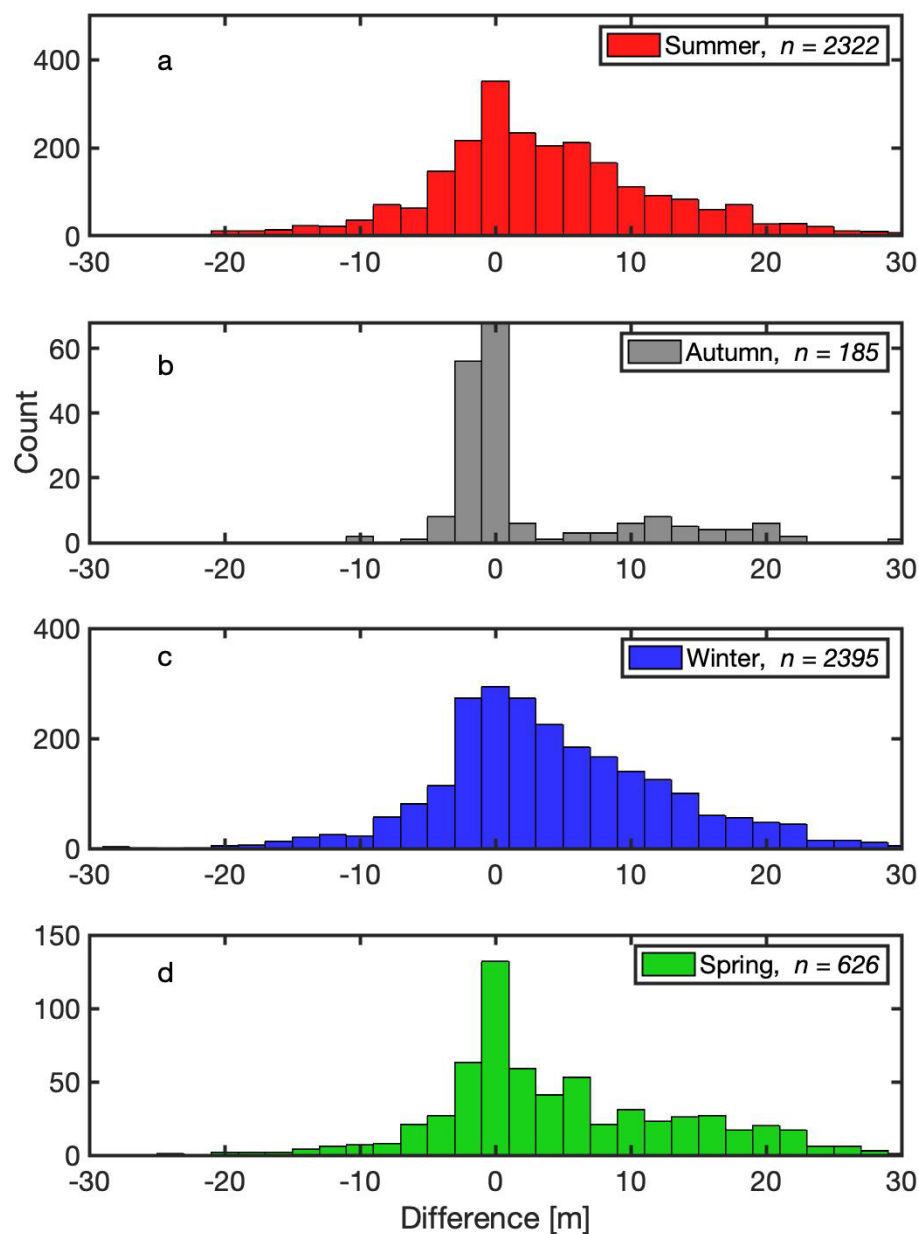


Figure 1.10. Histogram of elevation differences (residual values) after coregistration for 47 ICESat-2 transects totaling 5528 mean footprint elevations at Reynolds Creek EW in 2019 and 2020. Histograms are shown for summer (a), autumn (b), winter (c), and spring (d). Seasons are defined as summer (July – September), autumn (October – December), winter (January – March), and spring (April – June).

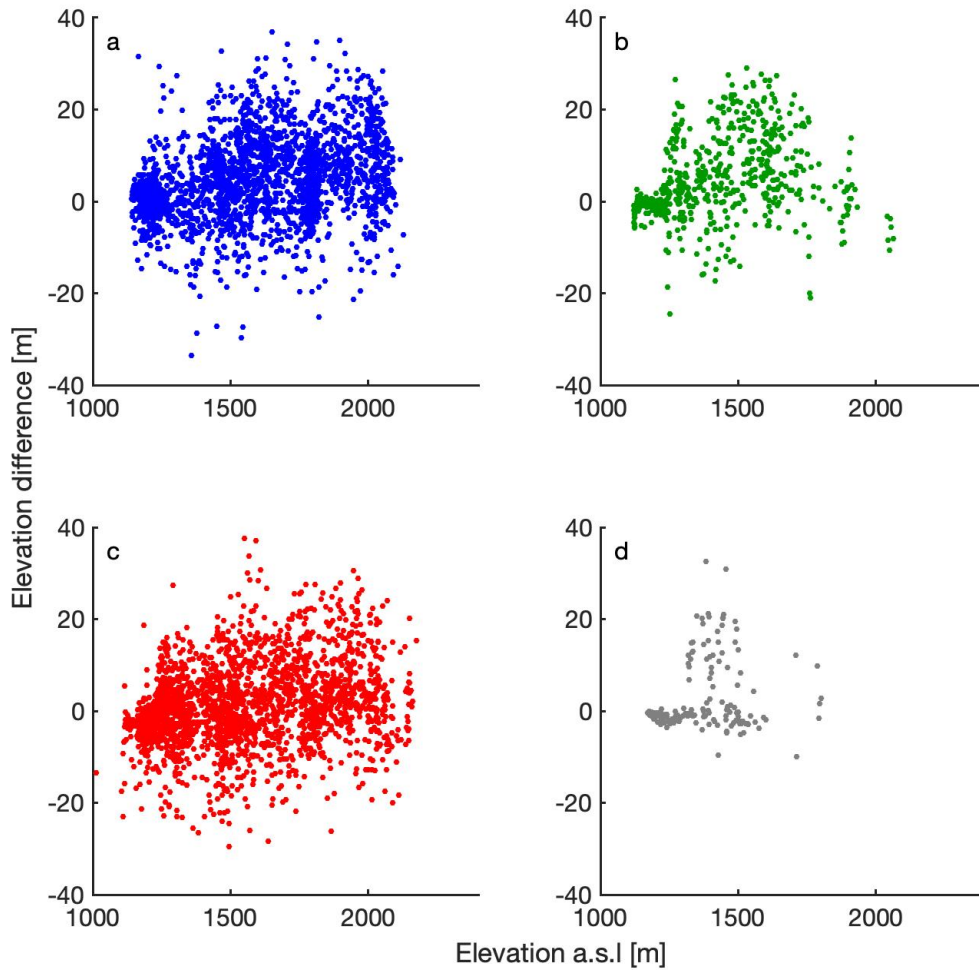


Figure 1.11. Plots of measured elevation difference as a function of reference elevation above sea level for ATL08 transects at Reynolds Creek EW for (a) winter, (b) spring, (c) summer, and (d) autumn.

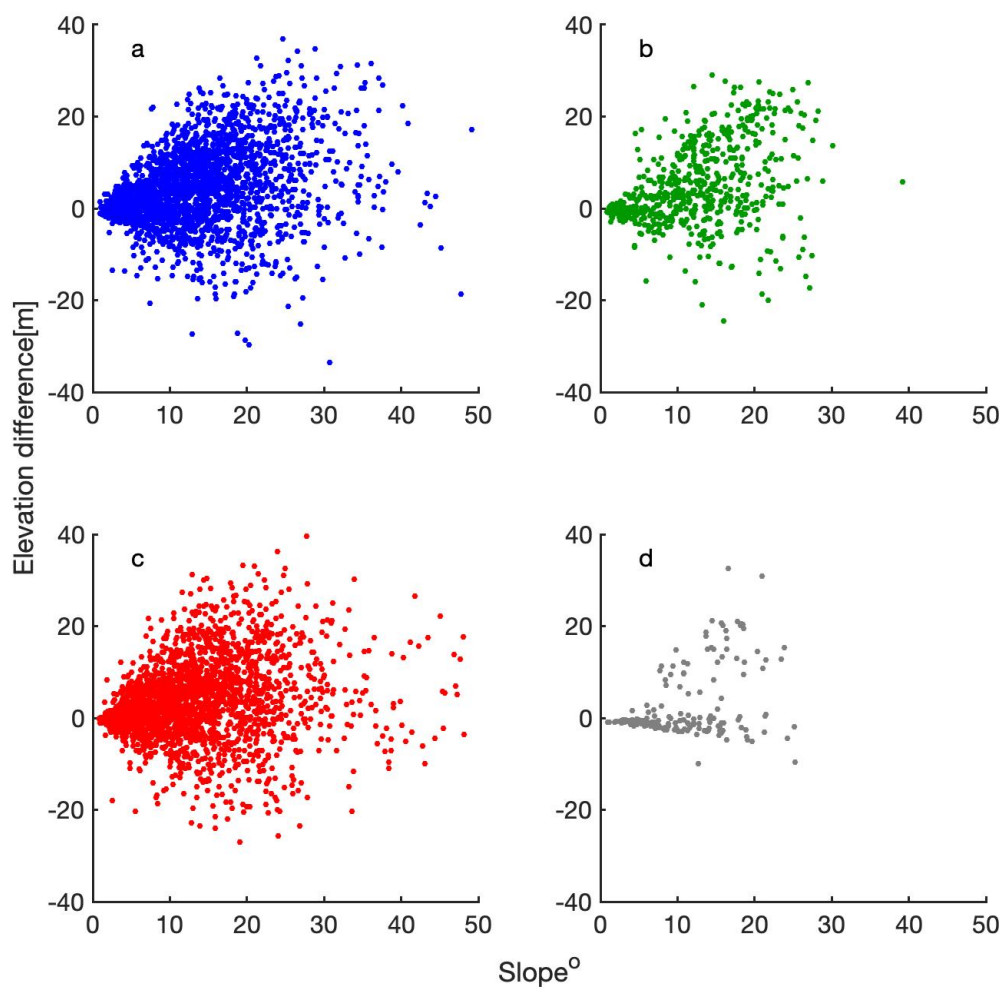


Figure 1.12. Plots of measured elevation difference as a function of slope for ATL08 transects at Reynolds Creek EW for (a) winter, (b) spring, (c) summer, and (d) autumn.

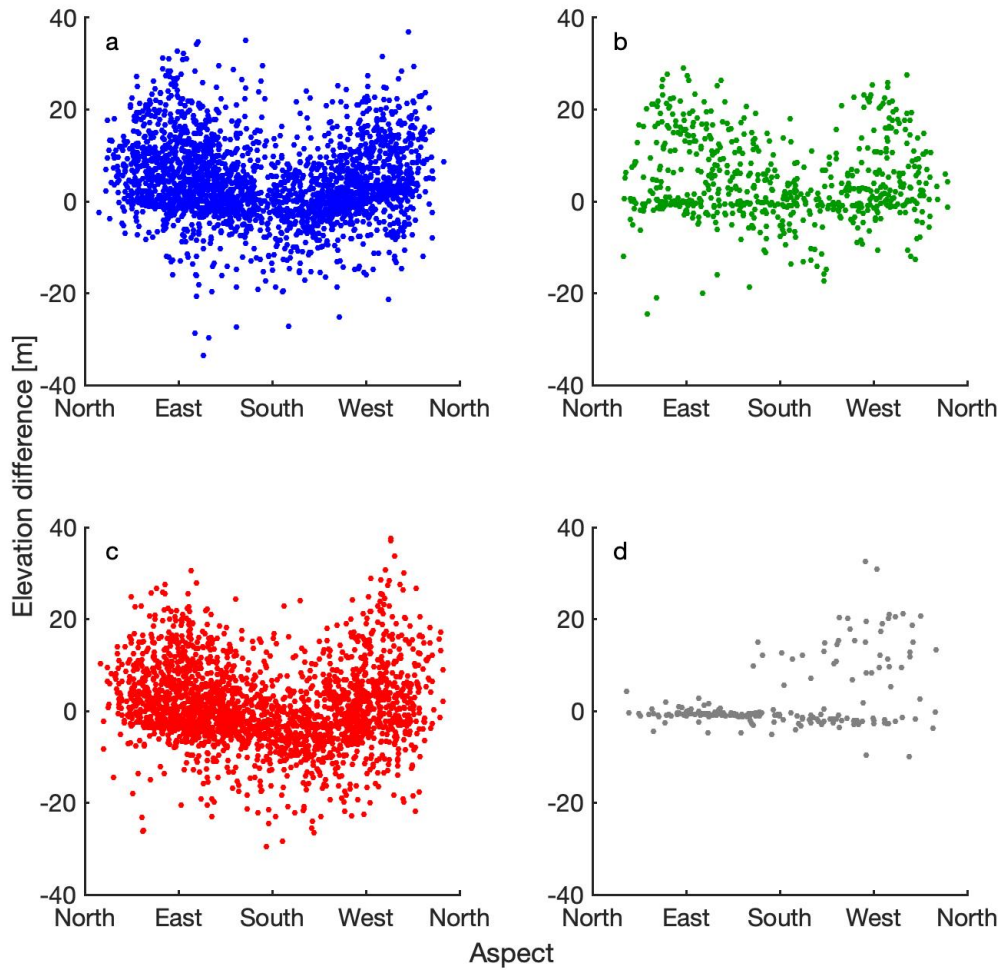


Figure 1.13. Plots of measured elevation difference as a function of aspect for ATL08 transects at Reynolds Creek EW for (a) winter, (b) spring, (c) summer, and (d) autumn.

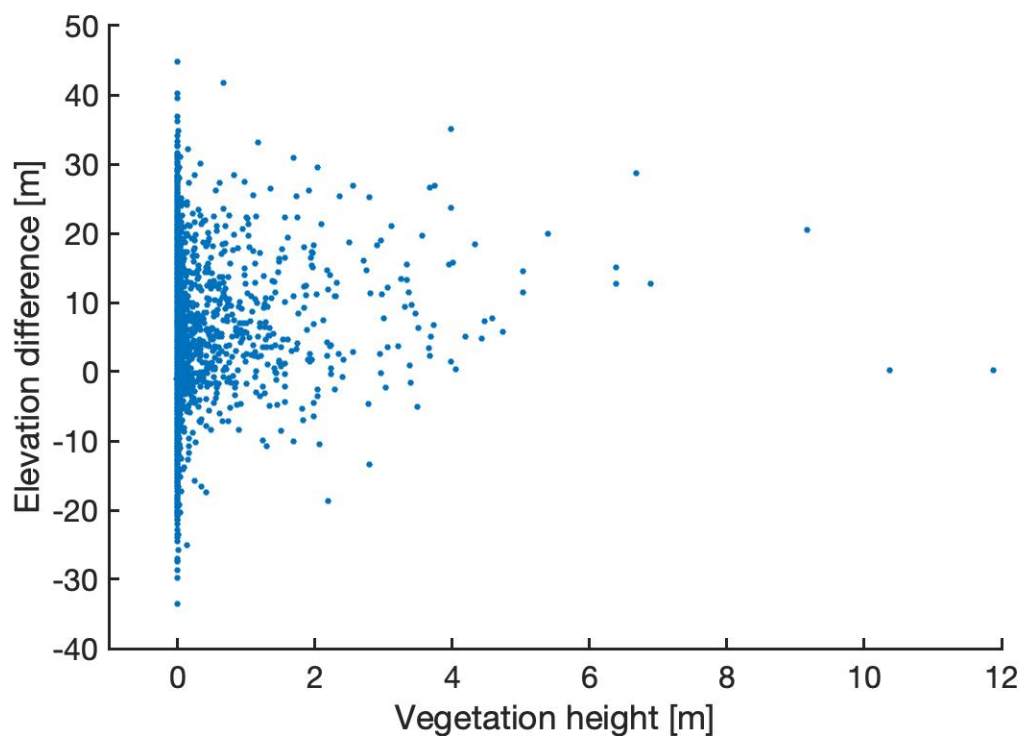


Figure 1.14. Histogram of elevation differences (residual values) as a function of vegetation height at Reynolds Creek EW for all 5528 sample points.

B. Wolverine Glacier

Of the 42 transects of the Wolverine Glacier watershed included in our study, 7 transects are horizontally coregistered, with median \pm MAD offset of 5.4 ± 2.8 m in the easting and 0.8 ± 0.6 m in the northing directions. To estimate the error associated with the lack of horizontal coregistration for non-summer transects, we compared the RMSD for summer transects without any horizontal coregistration to the coregistered RMSD. If the reference elevations fell within the vertical geolocation error of the ICESat-2 elevation reported at the same footprint, then the vertical offset used in calculating the RMSD was zero. For the seven horizontally-coregistered transects, coregistration decreased the median \pm MAD RMSD by 0.6 ± 0.2 m.

The seasonal distributions for Wolverine Glacier elevation residuals are shown in figure 1.12). The median \pm MAD are 0 ± 0.5 m, 1.5 ± 0.8 m, 5.5 ± 2.1 m, and 2.7 ± 2.1 m for summer, autumn, winter, and spring, respectively. Seasons were again defined as winter (January – March), spring (April – June), summer (July – September), and autumn (October – December). For Wolverine glacier, the distributions of elevation residuals for each season are statistically different at the $P = 0.05$ confidence level. For reference, the maximum annual snow depths retrieved from in situ snow pits and snow courses for the period 2007 – 2017 at Wolverine Glacier have a mean of 6.56 m and a standard deviation of 2.32 m (Baker et al., 2018).

The residual elevation distributions in figure 1.12 for winter and spring are strongly bimodal. Residual values for winter and spring are mapped in figure 1.13. Residual values are plotted against elevation in figure 1.14, slope in figure 1.15, and aspect in figure 1.16. Residual values are strongly positively correlated with elevation ($r^2 = 0.62$), with a linear polynomial slope of 0.005m/m (95% confidence interval [0.00405, 0.00595]).

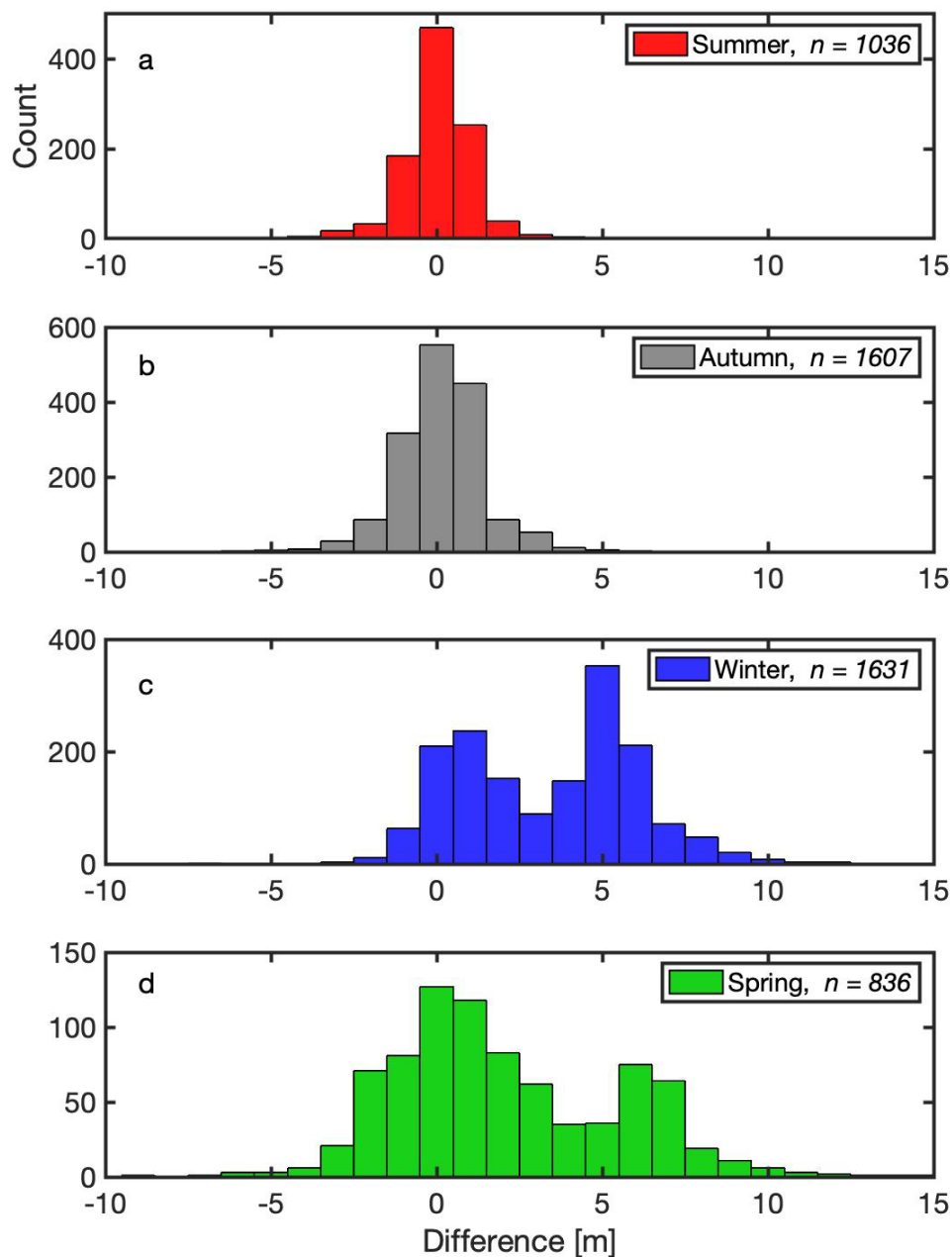


Figure 1.15. Histograms of elevation differences (residual values) after coregistration for 42 ATL06 transects totaling 5110 mean footprint elevations at Wolverine glacier in 2019 and 2020. The seasons are defined as winter (January – March), spring (April – June), summer (July – September), and autumn (October – December).

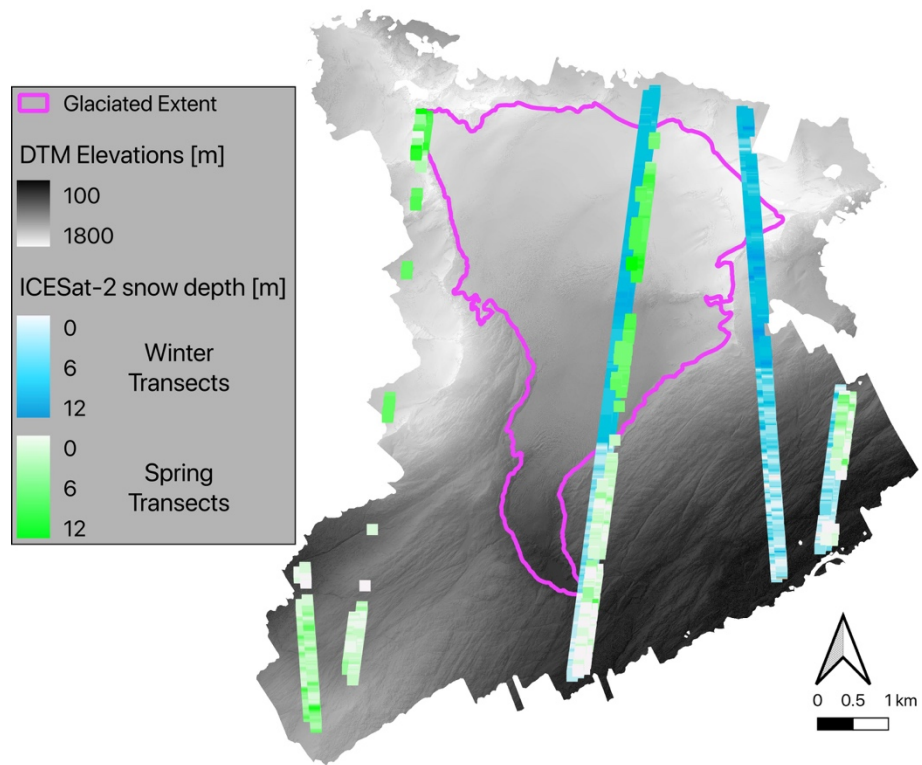


Figure 1.16. Estimated snow depths (residual values) at Wolverine glacier for winter (January-March) and spring (April – June) of 2018-2020. Residual values at each point along the transects are interpreted as estimated snow depths.

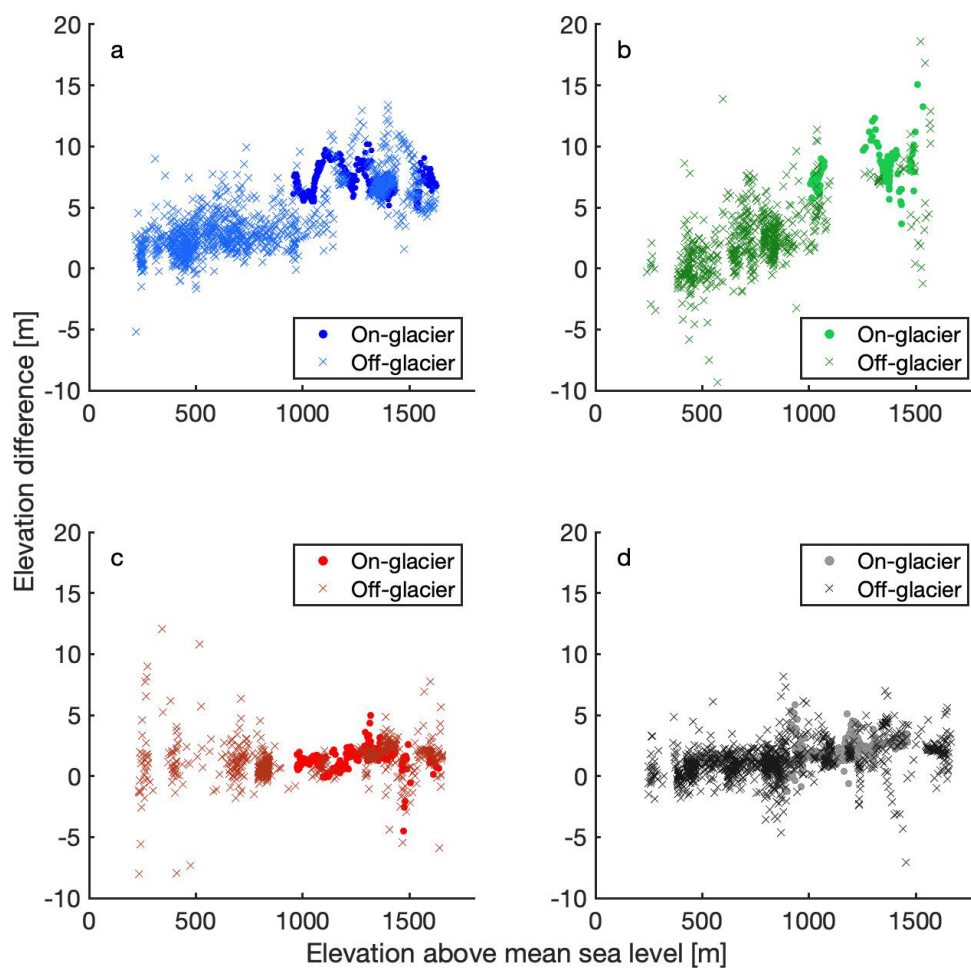


Figure 1.17. Plots of measured elevation difference as a function of reference elevation above sea level for ATL06 transects at Wolverine glacier for (a) winter, (b) spring, (c) summer, and (d) autumn.

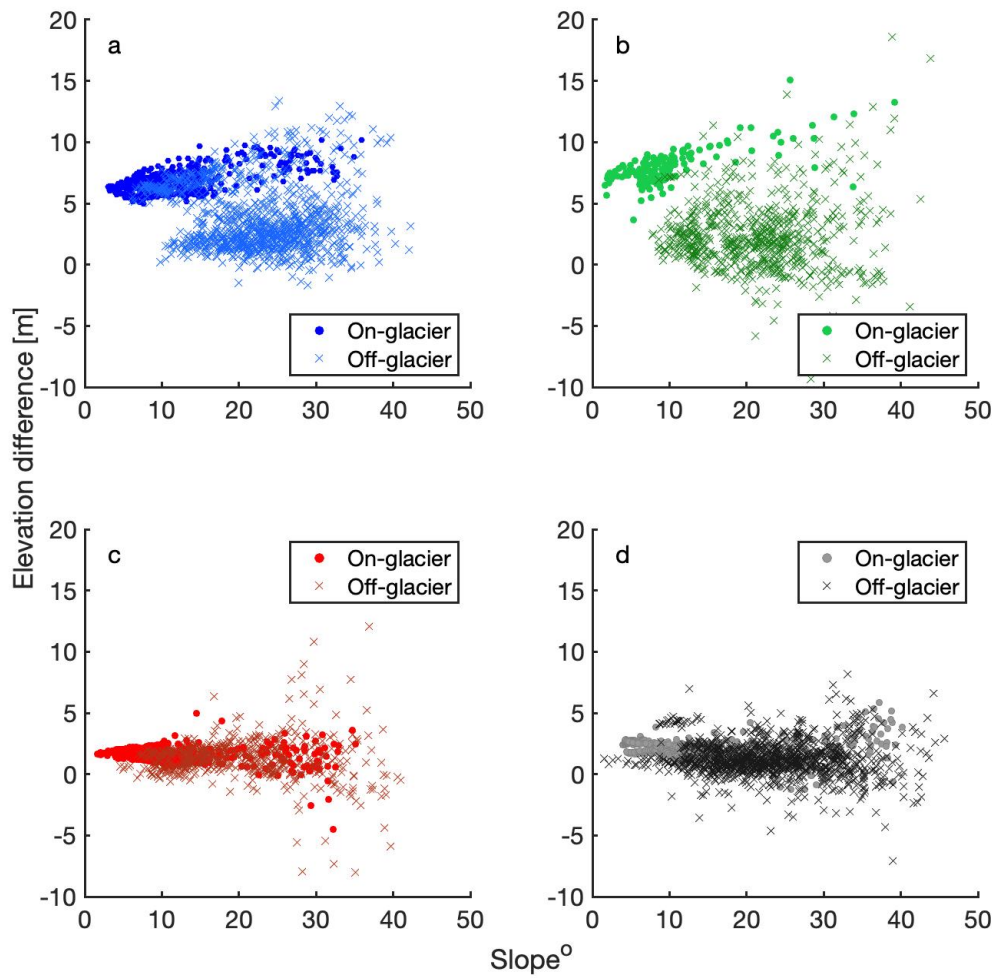


Figure 1.18. Plots of measured elevation difference as a function of slope for ATL06 transects at Wolverine glacier for (a) winter, (b) spring, (c) summer, and (d) autumn.

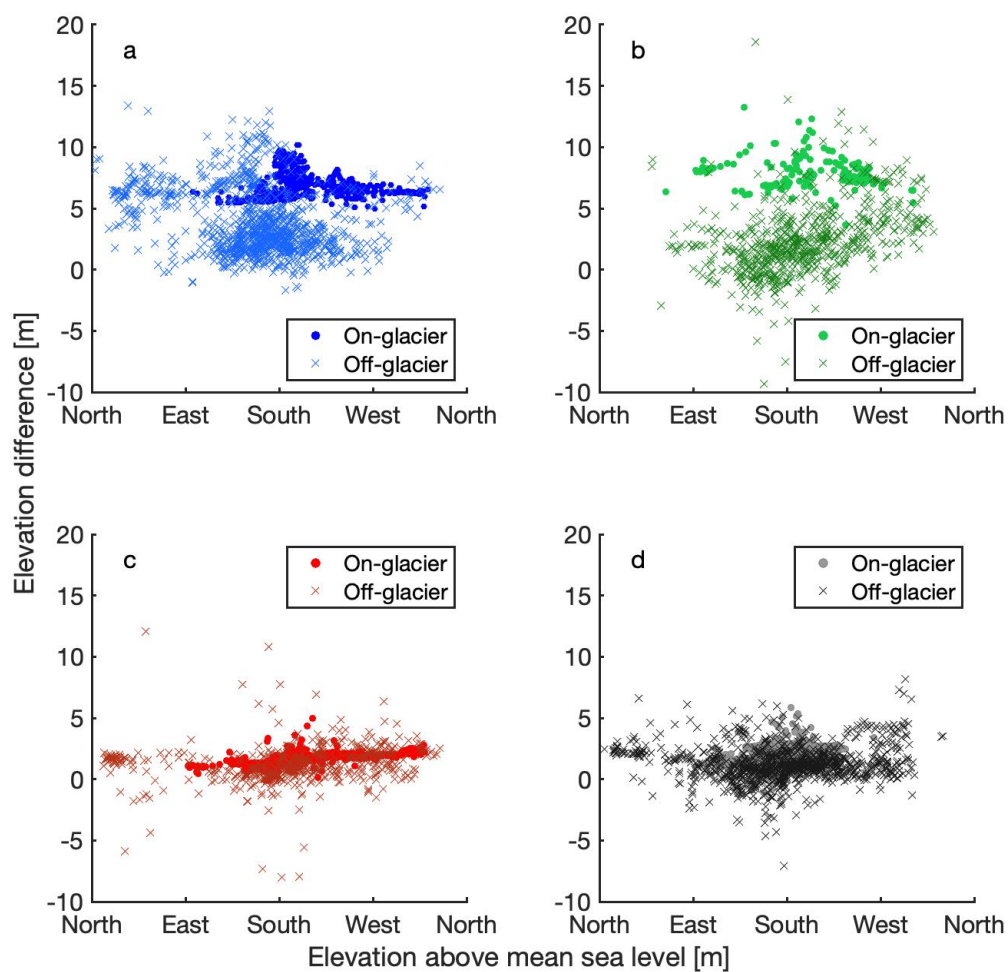


Figure 1.19. Plots of measured elevation difference as a function of aspect for ATL06 transects at Wolverine glacier for (a) winter, (b) spring, (c) summer, and (d) autumn.

V. Discussion

A. Error Quantification

The utility of ICESat-2 for snow depth estimation is dependent on uncertainties in both the ICESat-2 and reference elevation data as well as potential biases between the two datasets, as discussed below.

The first source of error comes from ICESat-2 measurements of terrain and canopy elevations. For the ATL08 data product, the estimated vertical accuracy is 0.40 m in sparsely vegetated regions and up to 2 m in densely vegetated regions (Neuenschwander and Pitts, 2019). For the ATL06 product, the vertical geolocation error variable is reported for each footprint. For the Wolverine Glacier transects, the median vertical geolocation error is 5.4 m but with considerable seasonal variability, as demonstrated in figure 1.6. The median vertical geolocation error reported is 5.6 m, 4.9 m, 4.7 m, and 12.1 m for summer, autumn, winter, and spring, respectively. Although assessment of controls on errors is outside the scope of our analysis, we hypothesize that the errors are smallest in winter due to strong returns from relatively smooth and highly reflective snow surfaces and peaks in spring when photons are reflected from both the surface and bottom of shallow supraglacial meltwater features.

Uncertainties in the reference DTMs also influence the vertical residuals. The Reynolds Creek EW Lidar data used herein have a reported vertical error of 0.25 m, although the Lidar returns are averaged into 2m raster cells during the downsampling process. The resulting uncertainty depends on the number of returns in each grid cell but is often much greater than 100. The expected uncertainty in each grid cell, therefore, is on the order of 0.025 m. The Wolverine glacier DTM sourced from the USGS has a reported

vertical error of 0.10 m. These DTM errors shrink proportionally to the root of the number of observations included in ICESat-2 footprint. For Reynolds Creek EW, there are 275 raster cell elevations included in each footprint. At Wolverine glacier, where both the ATL06 footprint and the raster cell resolution are smaller, there are 1600 raster cell elevations in each footprint. The expected ATL08 uncertainties and DTM uncertainties sum to a vertical error on elevation residuals of 0.10 m in sparsely vegetated terrain and .26 m in densely vegetated terrain. However, because the mean DTM elevation for each footprint inherently uniformly weights elevations over the entire footprint but the ICESat-2 photon returns are unlikely to adhere to a uniform distribution, spatial averaging increases the uncertainty of elevation residuals by an unquantifiable amount.

There are also unique, terrain-specific sources of errors for Reynolds Creek and Wolverine Glacier. Vegetation varies throughout the Reynolds Creek watershed. In order to accurately estimate snow depths from elevation residuals, vegetation must be accurately removed from both the reference DTM and the ATL08 ground heights. A comparison of the elevation residuals to the difference between the first and last returns from the reference Lidar data, which approximates vegetation heights, suggests that inaccurate removal of vegetation from ATL08 ground heights cannot explain the large spread in our elevation residuals. However, a more rigorous comparison is needed to fully assess the influence of vegetation on both elevation datasets.

Although Wolverine Glacier does not suffer from uncertainties introduced by vegetation, glacier movement and mass change can potentially bias elevation residuals and their interpretation. The reference DTM elevations inside of the GLIMS polygon are subject to change from translation of ice (lateral flow), emergence or submergence of ice

(vertical flow), snow accumulation, and ablation of snow, firn, and ice. We attempt to account for glacier movement and mass change through the construction of 2019 and 2020 end-of-melt-season DTMs using the decadal trends in elevation change for the glacier. Our extrapolation approach is computationally efficient but physically crude. Long-term mass balance estimates for the glacier suggest a constant rate of mass loss over decadal time scales since the late 1980s but with considerable inter-annual variability (O'Neel et al., 2019). Inter-annual variations in the glacier's mass balance can cause deviations in the rate of surface lowering due to firn compaction and ice melt relative to long-term trends. Firn compaction and vertical ice flow cause surface lowering at rates of up to 4.12 m a^{-1} in the glacier's accumulation zone (Zeller and McGrath, personal communication). As such, winter ATL06 elevation residuals calculated with respect to the end-of-melt-season DTM may under-estimate snow depths by several meters at high elevations. The glacier's steep mass balance gradient also results in large rates of upwards ice flow in the ablation zone, with emergence velocities as high as 5.5 m a^{-1} of at 800 m a.s.l. (Zeller and McGrath, personal communication). Thus, winter ATL06 elevation residuals calculated with respect to the end-of-melt-season DTM may over-estimate snow depths by several meters at low elevations. Failure to include intra-annual glacier surface elevation changes results in an apparent flattening of the winter snow accumulation gradient, as illustrated from ~800-1600 m a.s.l. in figure 1.14a.

B. Reynolds Creek Experimental Watershed and the ATL08 Product

Mean maximum annual snow depths from the in situ SNOTEL site at 1700 m a.s.l. for the period 2010 – 2019 are 0.47 m and have a standard deviation of 0.24 m. Although the ATL08 snow depths inferred from the elevation residuals during the winter

and spring cannot be directly compared to the SNOTEL data because they are neither coincident in time or space, it is problematic that the residuals are nearly an order of magnitude greater than maximum in situ snow depths (3.1 m and 2.5 m, respectively). Furthermore, the median absolute deviations for each season at Reynolds Creek EW are ~7 m, such that snow depths cannot be confidently inferred from the individual point elevations.

The large spread in the elevation residuals is likely in part due to the need for precise coregistration to a reference DTM before they can be used to calculate snow depths. In theory, snow-free regions identified using the ATL08 brightness flag can be used for horizontal and vertical coregistration, yielding precise estimates of snow depth for each transect. We find, however, that the brightness flag fails to distinguish snow-free and snow-covered regions. For example, the SNOTEL site for Reynolds Creek (site 2029), which is in the southern highlands of the watershed, recorded 0.28 m of snow on 26 January 2020 (Table 1.1). Landsat-8 images from before and after 26 January 2020 both show snow cover across the majority of Reynolds Creek EW, supporting the widespread presence of snow. However, for the 632 ATL08 footprints that were recorded for the six beam transects on 26 January 2020, there was not a single brightness-flagged point. This indicates that the ATL08 brightness flag severely and systematically underreports snow- or ice-covered areas. In fact, none of the 2580 individual footprints from typical snow accumulation months (November to March for Reynolds Creek EW) included a brightness flag indicative of snow cover. Based on preliminary radiometry work done by Neuenschwander et al. (2020), the brightness flag should be adjusted from >3 photons per shot for the strong beam down to >1 photon per shot for highly reflective

ground conditions (i.e., snow and ice cover) for better performance. Given the failure of the brightness flag, we attribute our large and highly variable elevation residuals to coregistration difficulties and recommend further work to refine the brightness flag before the ATL08 data can be used to accurately map snow depths in non-glacierized mountainous watersheds.

Table 1.1. Reynolds Creek SNOTEL Site 2029 data for the month of January 2020.

Date	Snow Depth (cm)	Snow Water Equivalent (cm)
1 Jan 2020	25	8
2 Jan 2020	25	8
3 Jan 2020	25	8
4 Jan 2020	27	8
5 Jan 2020	27	10
6 Jan 2020	25	10
7 Jan 2020	25	8
8 Jan 2020	27	8
9 Jan 2020	30	10
10 Jan 2020	30	10
11 Jan 2020	33	13
12 Jan 2020	33	13
13 Jan 2020	38	13
14 Jan 2020	41	13
15 Jan 2020	46	15
16 Jan 2020	46	15
17 Jan 2020	43	15
18 Jan 2020	43	15
19 Jan 2020	41	13
20 Jan 2020	36	13
21 Jan 2020	36	13
22 Jan 2020	36	10
23 Jan 2020	30	10
24 Jan 2020	33	10
25 Jan 2020	33	8
26 Jan 2020	30	8
27 Jan 2020	25	8
28 Jan 2020	25	8
29 Jan 2020	25	8
30 Jan 2020	30	10
31 Jan 2020	30	10

C. Wolverine Glacier and the ATL06 Product

In general, we find that the ATL06 data product is adequately coregistered. Transects from July, August, and September -- the summer months -- have a small proportion of footprint elevations out of agreement with the reference elevations once the vertical geolocation errors are considered. Summer transects have a median of 3.9% of footprint elevations that differ from the reference elevations by more than the reported vertical error. Disagreement between the ATL06 and reference elevations is greater throughout the rest of the year – autumn, winter, and spring transects have a median of 26% of footprint elevations out of agreement – due to the presence of snow. Snow increases surface elevations and, we hypothesize, decreases the reported vertical error because it is highly reflective and has a smoother macrosurface than rock outcroppings or glacial ice (Wiscombe & Warren, 1980; Filhol & Sturm, 2019), which increases the number of return photons per shot and compresses the distribution of photon elevations in each footprint. Figure 1.6b demonstrates the elevation bias and reduced uncertainty observed outside the summer months. Future work could focus on examining the total vertical geolocation error as a function of days since the last fresh snowfall event to test the aforementioned hypothesis.

The bimodal winter and spring distributions apparent in figure 1.12 appear to be forced by differences in snow accumulation as a function of elevation. Figure 1.13 plots elevation residuals as a function of elevation. There is a positive relationship between elevation residuals and elevation with an average slope of 0.005m/m, which is consistent with previously observed snow depth gradients for Wolverine Glacier despite the strong dependence of high elevation snow depths on wind redistribution and terrain effects

(McGrath et al., 2018). We do not find relationships between elevation residuals and aspect or slope (figure 1.15-1.16). However, as previously discussed in the section on error quantification, our elevation residuals likely under-estimate snow depths at high elevations and over-estimate snow depths low on the glacier because we do not account for intra-annual surface elevation change due to emergence and submergence of ice. In short, while the methodology described in this paper yields first-order estimates of snow depths for glaciated areas covered by the ATL06 data product, further refinement and greater quantitative error assessment is needed before the ATL06 data can be used for precise snow depth estimation.

VI. Conclusions

Seasonal snow depths in remote, mountainous terrain are difficult to measure and often have inconsistent spatial and temporal coverage. Here we explore whether the ICESat-2 level-3a data products can be used to efficiently and accurately estimate snow depths. We find that largely due to coregistration difficulties, the ATL08 product cannot be used to map spatial variations in snow depths in vegetated mountainous watersheds. However, further refinement of the brightness flag included in the ATL08 product could reduce vertical uncertainties associated with imprecise coregistration and potentially enable snow depth estimation.

In contrast, the ATL06 product provides reasonable first-order estimates of snow depth in glacierized terrain when paired with annual end-of-melt-season DTMs. More precise estimates of annual snow accumulation is possible for glaciers with either well-constrained or relatively small submergence and emergence velocities. Based on our analysis, we suggest that the ATL06 data are used to examine the snow depths on a large

scale, and to refine the relationships between snow depths and terrain parameters such as slope, aspect, and elevation to snow depths in glacierized mountainous watersheds.

REFERENCES

- Alexander, Cici (2021). LASRead
(<https://www.mathworks.com/matlabcentral/fileexchange/21434-lasread>),
MATLAB Central File Exchange. Retrieved January 20, 2021.
- Anderton, S. P., White, S. M., & Alvera, B. (2004). Evaluation of spatial variability in snow water equivalent for a high mountain catchment. *Hydrological Processes*, 18(3), 435-453.
- Archer, D. R., Forsythe, N., Fowler, H. J., & Shah, S. M. (2010). Sustainability of water resources management in the Indus Basin under changing climatic and socio economic conditions. *Hydrology and Earth System Sciences*.
- Baffaut, C., Baker, J. M., Biederman, J. A., Bosch, D. D., Brooks, E. S., Buda, A. R., ... & Yasarer, L. M. (2020). Comparative analysis of water budgets across the US Long-Term Agroecosystem Research network. *Journal of Hydrology*, 125021.
- Baker, E. H., McNeil, C. J., Sass, L. C., & O'Neel, S. (2018). Point Raw Glaciological Data: Ablation Stake, Snow Pit, and Probed Snow Depth Data on USGS Benchmark Glaciers, 1981-2016.
- Barnett, Tim P., Jennifer C. Adam, and Dennis P. Lettenmaier. (2005) "Potential impacts of a warming climate on water availability in snow-dominated regions." *Nature* 438.7066: 303.
- Beniston, M., Keller, F., Koffi, B., & Goyette, S. (2003). Estimates of snow accumulation and volume in the Swiss Alps under changing climatic conditions. *Estimated and Applied Climatology*, 76(3-4), 125-140.
- Brandt, W. T., Bormann, K. J., Cannon, F., Deems, J. S., Painter, T. H., Steinhoff, D. F., & Dozier, J. (2020). Quantifying the Spatial Variability of a Snowstorm Using Differential Airborne Lidar. *Water Resources Research*, 56(3), e2019WR025331.

- Brucker, L., Hiemstra, C., Marshall, H. P., Elder, K., De Roo, R., Mousavi, M., ... & Gelvin, A. (2018, July). Nasa Snowex'17 in SITU Measurements and Ground-Based Remote Sensing. In IGARSS 2018-2018 IEEE International Geoscience and Remote Sensing Symposium (pp. 6266-6268). *IEEE*.
- Fiebrich, C. A. (2009). History of surface weather observations in the United States. *Earth-Science Reviews*, 93(3-4), 77-84.
- Filhol, S., & Sturm, M. (2019). The smoothing of landscapes during snowfall with no wind. *Journal of Glaciology*, 65(250), 173-187.
- Grünewald, T., Schirmer, M., Mott, R., & Lehning, M. (2010). Spatial and temporal variability of snow depth and SWE in a small mountain catchment. *The Cryosphere*, 4, 215-225.
- Hall, D. K., Riggs, G. A., Salomonson, V. V., DiGirolamo, N. E., & Bayr, K. J. (2002). MODIS snow-cover products. *Remote sensing of Environment*, 83(1-2), 181-194.
- Hanson, C. L. (2001). Long-term precipitation database, reynolds creek experimental watershed, Idaho, United States. *Water Resources Research*, 37(11), 2831-2834.
- Hislop, S., Jones, S., Soto-Berelov, M., Skidmore, A., Haywood, A., & Nguyen, T. H. (2018). Using landsat spectral indices in time-series to assess wildfire disturbance and recovery. *Remote sensing*, 10(3), 460.
- Huss, M., & Hock, R. (2018). Global-scale hydrological response to future glacier mass loss. *Nature Climate Change*, 8(2), 135-140.
- Idaho Lidar Consortium. 2007. <https://www.idahoLidar.org/>
- Ilangakoon, N., Glenn, N. F., Spaete, L. P., Dashti, H., & Li, A. (2016). 2014 Lidar-Derived 1m Digital Elevation Model Data Set for Reynolds Creek Experimental Watershed, Southwestern Idaho.
- Itten, K. I. (1975). Approaches to digital snow mapping with Landsat-1 data. NASA Goddard Space Flight Center Operational Applications of Satellite.

- Kelly, S. A., & Belmont, P. (2018). High resolution monitoring of river bluff erosion reveals failure mechanisms and geomorphically effective flows. *Water*, 10(4), 394.
- Kim, E., Gatebe, C., Hall, D., Newlin, J., Misakonis, A., Elder, K., ... & Crawford, C. (2017, July). NASA's SnowEx campaign: Observing seasonal snow in a forested environment. In 2017 IEEE International Geoscience and Remote Sensing Symposium (IGARSS) (pp. 1388-1390). *IEEE*.
- Lievens, H., Demuzere, M., Marshall, H. P., Reichle, R. H., Brucker, L., Brangers, I., ... & Jonas, T. (2019). Snow depth variability in the Northern Hemisphere mountains observed from space. *Nature communications*, 10(1), 1-12.
- Magruder, L., Neumann, T., Fricker, H.A., Farrell, S.L., Brunt, M.K., Gardner, A., ... & Yang, Y. (2019), New Earth orbiter provides a sharper look at a changing planet, *Eos*, 100, <https://doi.org/10.1029/2019EO133233>.
- Markus, T., Neumann, T., Martino, A., Abdalati, W., Brunt, K., Csatho, B., ... & Jasinski, M. (2017). The Ice, Cloud, and land Elevation Satellite-2 (ICESat-2): science requirements, concept, and implementation. *Remote Sensing of Environment*, 190, 260-273.
- McGrath, D., Sass, L., O'Neel, S., McNeil, C., Candela, S. G., Baker, E. H., & Marshall, H. P. (2018). Interannual snow accumulation variability on glaciers derived from repeat, spatially extensive ground-penetrating radar surveys. *The Cryosphere*, 12(11), 3617-3633.
- McNeil, C. J., Florentine, C. E., Bright, V. A., Fahey, M. J., Evan, E. M. C. F. L., Thoms, E., ... & Baker, E. H. (2019). Geodetic Data for USGS Benchmark Glaciers: Orthophotos, Digital Elevation Models, and Glacier Boundaries (ver 1.0, September 2019): U.S. Geological Survey data release, <https://doi.org/10.5066/P9R8BP3K>.

- Mishra, S. K., Hayse, J., Veselka, T., Yan, E., Kayastha, R. B., LaGory, K., ... & Steiner, N. (2018). An integrated assessment approach for estimating the economic impacts of climate change on River systems: An application to hydropower and fisheries in a Himalayan River, Trishuli. *Environmental Science & Policy*, 87, 102-111.
- National Resources Conservation Service (2012). Part 622 Snow Survey and Water Supply Forecasting. In L. Owens and S. Self (eds.), *National Engineering Handbook*. pp 622.0100 – 622.0110. Washington, DC: United States Department of Agriculture.
- Neuenschwander, Amy, and Katherine Pitts. (2019). "The ATL08 land and vegetation product for the ICESat-2 Mission." *Remote sensing of environment* 221: 247-259.
- Neuenschwander, A., Guenther, E., White, J. C., Duncanson, L., & Montesano, P. (2020). Validation of ICESat-2 terrain and canopy heights in boreal forests. *Remote Sensing of Environment*, 251, 112110.
- O'Neel, S., Hood, E., Bidlack, A. L., Fleming, S. W., Arimitsu, M. L., Arendt, A., ... & Hayward, G. D. (2015). Icefield-to-ocean linkages across the northern Pacific coastal temperate rainforest ecosystem. *BioScience*, 65(5), 499-512.
- O'Neel, S., McNeil, C., Sass, L. C., Florentine, C., Baker, E. H., Peitzsch, E., ... & Fagre, D. (2019). Reanalysis of the US Geological Survey Benchmark Glaciers: long-term insight into climate forcing of glacier mass balance. *Journal of Glaciology*, 65(253), 850-866.
- Orheim, O., & Lucchitta, B. K. (1987). Snow and ice studies by Thematic Mapper and multispectral scanner Landsat images. *Annals of Glaciology*, 9, 109-118.
- Pettinato, S., Santi, E., Brogioni, M., Paloscia, S., Pampaloni, P., Palchetti, E., ... & Xiong, C. (2011, July). The potential of Cosmo-Skymed SAR images in mapping snow cover and snow water equivalent. In 2011 IEEE International Geoscience and Remote Sensing Symposium (pp. 2733-2736). IEEE.

- Pflug, J. M., & Lundquist, J. D. (2020). Inferring distributed snow depth by leveraging snow pattern repeatability: Investigation using 47 lidar observations in the Tuolumne watershed, Sierra Nevada, California. *Water Resources Research*, 56(9), e2020WR027243.
- Rango, A., Martinec, J., Chang, A. T., Foster, J. L., & Van Katwijk, V. (1989). Average areal water equivalent of snow in a mountain basin using microwave and visible satellite data. *IEEE Transactions on Geoscience and Remote Sensing*, 27(6), 740-745.
- Rebold, T. W., Luthcke, S. B., Thomas, T., & Nicholas, J. B. (2019). ICESat-2 Surface Sampling Performance: Orbit Design and Calibration and Validation of Reference Ground-Track Pointing Control. AGUFM, 2019, C31C-1530.
- Rott, H., Heidinger, M., Nagler, T., Cline, D., & Yueh, S. (2009, July). Retrieval of snow parameters from Ku-band and X-band radar backscatter measurements. In 2009 IEEE International Geoscience and Remote Sensing Symposium (Vol. 2, pp. II-144). IEEE.
- Seyfried, M., Harris, R., Marks, D., & Jacob, B. (2001). Geographic database, Reynolds Creek Experimental Watershed, Idaho, United States. *Water Resources Research*, 37(11), 2825-2829.
- Seyfried, M., Lohse, K., Marks, D., Flerchinger, G., Pierson, F., & Holbrook, W. S. (2018). Reynolds creek experimental watershed and critical zone observatory. *Vadose Zone Journal*, 17(1).
- Smith, B., Fricker, H. A., Holschuh, N., Gardner, A. S., Adusumilli, S., Brunt, K. M., ... & Siegfried, M. R. (2019). Land ice height-retrieval algorithm for NASA's ICESat-2 photon-counting laser altimeter. *Remote Sensing of Environment*, 233, 111352.
- Wiscombe, W. J., & Warren, S. G. (1980). A model for the spectral albedo of snow. I: Pure snow. *Journal of Atmospheric Sciences*, 37(12), 2712-2733.

Zeller, L., McGrath, D., O'Neel, S., & Sass, L. (2020, November). Coupling radar and repeat geodetic observations to constrain vertical ice velocities of Wolverine Glacier, Alaska. In 18th International Conference on Ground Penetrating Radar (pp. 105-108). Society of Exploration Geophysicists.

APPENDIX

Code

Each of the following scripts and live scripts were written by the author of this paper, Colten Elkin, and were written in Matlab R2020b under an academic use license.

For more information or for a complete user guide contact the author at

coltenelkin@u.boisestate.edu.

A. Coregister_icesat2.m

```

function rmsez = coregister_icesat2(icesat2, elevations, R2, A)
% Function COREGISTER_ICESAT2 coregisters icesat-2 data with a
corresponding digital
% terrain model
% INPUTS: icesat2 = a csv file with icesat 2 elevations created using the
%          h5 to csv jupyter notebook
%          elevations = the matrix created using geotiffread()
%          R2 = the cell map refernce created as the second output in
%          geotiffread()
%          A = a [2 1] vector that serves as the spatial offsets in
%          the x and y directions (meters)
% OUTPUTS: rmsez = the root mean squared difference between the icesat-2
%          elevations and their corresponding (offset) DTM
%          elevations

% Created 19 October 2020 by Colten Elkin (coltenelkin@u.boisestate.edu)
% last modified 30 Jan 2021

% most recent update: added ATL08 brightness flag lines

elevations(elevations < -10) = nan; % throw out trash data
elevations(elevations > 10000) = nan; % more trash takeout

T = readtable(icesat2);

zmod = T.Elevation(1:end-1); % save the 'model' elevations (icesat-2 elevations)

east = T.Easting(:); % pull out the easting values
north = T.Northing(:); % pull out the northings
footwidth = 11; % approx. width of icesat2 shot footprint in meters

% for ATL08 files only use snow-free data (brightness flag negative)
if strcmp(icesat2(end-44:end-40), 'ATL08') == 1 % ATL08 commands
    bright = T.Brightness_Flag;

```

```

    ib = find(bright == 0);
    easts = easts(ib);
    norths = norths(ib);
    zmod = zmod(ib);
end

theta = zeros(size(norths)); % initialize empty matrices
xs = {};
ys = {};
xpoly = nan([1,5]);
ypoly = nan([1,5]);

for r = 1:length(theta)-1
    if strcmp(icesat2(end-44:end-40), 'ATL08') == 1 % ATL08 commands
        theta(r) = abs(atan((norths(r+1) - norths(r))/(easts(r+1) - easts(r)))); % trig to
get angle theta along-track

        % get the x and y vectors to form the polygon
        xpoly(1) = easts(r) + (footwidth/2) - footwidth/2*cos((pi/2) - theta(r)); %
calculate the 4 corners in the x direction
        xpoly(2) = easts(r) + (footwidth/2) + footwidth/2*cos((pi/2) - theta(r));
        xpoly(3) = easts(r) - (footwidth/2) + footwidth/2*cos((pi/2) - theta(r));
        xpoly(4) = easts(r) - (footwidth/2) - footwidth/2*cos((pi/2) - theta(r));
        xpoly = xpoly+A(1); % adjust by the easting offset
        xs{r} = [xpoly(1), xpoly(2), xpoly(3), xpoly(4), xpoly(1)]; % save the
corners as a vector in the x-es cell array

        ypoly(1) = norths(r) - 50 - footwidth/2*cos((pi/2) - theta(r)); % calculate the
4 corners in the y direction
        ypoly(2) = norths(r) - 50 + footwidth/2*cos((pi/2) - theta(r));
        ypoly(3) = norths(r) + 50 + footwidth/2*cos((pi/2) - theta(r));
        ypoly(4) = norths(r) + 50 - footwidth/2*cos((pi/2) - theta(r));
        ypoly = ypoly+A(2); % adjust by the nothing offset
        ys{r} = [ypoly(1), ypoly(2), ypoly(3), ypoly(4), ypoly(1)]; % save the
corners as a vector in the y-s cell array

    elseif strcmp(icesat2(end-44:end-40), 'ATL06') == 1 % ATL06 commands
        theta(r) = abs(atan((norths(r+1) - norths(r))/(easts(r+1) - easts(r)))); % trig to
get angle theta along-track

        % get the x and y vectors to form the polygon
        xpoly(1) = easts(r) + (footwidth/2) - footwidth/2*cos((pi/2) - theta(r)); %
calculate the 4 corners in the x direction
        xpoly(2) = easts(r) + (footwidth/2) + footwidth/2*cos((pi/2) - theta(r));

```

```

    xpoly(3) = easts(r) - (footwidth/2) + footwidth/2*cos((pi/2) - theta(r));
    xpoly(4) = easts(r) - (footwidth/2) - footwidth/2*cos((pi/2) - theta(r));
    xpoly = xpoly+A(1); % adjust by the easting offset
    xs{r} = [xpoly(1), xpoly(2), xpoly(3), xpoly(4), xpoly(1)]; % save the
corners as a vector in the x-es cell array

    ypoly(1) = norths(r) - 20 - footwidth/2*cos((pi/2) - theta(r)); % calculate the
4 corners in the y direction
    ypoly(2) = norths(r) - 20 + footwidth/2*cos((pi/2) - theta(r));
    ypoly(3) = norths(r) + 20 + footwidth/2*cos((pi/2) - theta(r));
    ypoly(4) = norths(r) + 20 - footwidth/2*cos((pi/2) - theta(r));
    ypoly = ypoly+A(2); % adjust by the nothing offset
    ys{r} = [ypoly(1), ypoly(2), ypoly(3), ypoly(4), ypoly(1)]; % save the
corners as a vector in the y-s cell array
end
end

    x =
R2.XWorldLimits(1)+0.5*R2.CellExtentInWorldX:R2.CellExtentInWorldX:R2.XWorld
Limits(end)-0.5*R2.CellExtentInWorldX; % get a vector of x coords
    y =
R2.YWorldLimits(1)+0.5*R2.CellExtentInWorldY:R2.CellExtentInWorldY:R2.YWorld
Limits(end)-0.5*R2.CellExtentInWorldY; % get a vector of y coords

[xgrid, ygrid] = meshgrid(x, y); % create grids of each of the x and y coords
elevation_report = zeros([1, length(xs)]);

for t = 1:length(xs)
    xv = xs{t}; % bounding box x vector
    yv = ys{t}; % bounding box y vector

    % first trimming
    in = inpolygon(xgrid, ygrid, xv, yv); % get logical array of in values
    pointsinx = xgrid(in); % save x locations
    pointsiny = ygrid(in); % save y locations
    in2 = flip(in); % create a flipped in-grid (need row, column instead of column,
row)
    elevationsin = elevations(in2); % save elevations
    elevation_report(t) = nanmean(elevationsin);
end
ztruth = elevation_report(:);

differences = zmod - ztruth;

```

```

differences(differences > 80) = nan;
differences(differences < -80) = nan;

rmsez = sqrt(nanmean((differences).^2));

```

B. Icesat2_residuals.m

```

function [rmsez, residuals, differences] = icesat2_residuals(icesat2, elevations,
R2, Abest, offset)
% Function ICESAT2_RESIDUALS calculates residuals of a given ICESat-2
track
% INPUTS: icesat2 = a csv file with icesat 2 elevations created using the
%          h5 to csv jupyter notebook
%          elevations = the matrix created using geotiffread()
%          R2 = the cell map reference created as the second output in
%          geotiffread()
%          Abest = a [2 1] vector that serves as the best spatial offsets in
%          the easting and northing directions (meters)
%          offset = the offset in elevation values
%          pulled from the function value using the fminsearch approach
% OUTPUTS: rmsez = the root mean squared difference in elevation values
%          before offset
%          residuals = the calculated residuals (vector) between the icesat-2
%          elevations and their corresponding (offset) DTM
%          elevations
%          differences = the vector of point-by-point elevation differences
%          before elevation offset

% Created 30 October 2020 by Colten Elkin (coltenelkin@u.boisestate.edu)
% last modified 30 January 2021

% most recent update: added two lines to deal with various no data values
% used in DTM storage (-9999; 4.02e38, etc.);

elevations(elevations < -10) = nan; % throw out trash data
elevations(elevations > 10000) = nan; % more trash takeout

A = Abest; % save as the name that's in the loop below

T = readtable(icesat2); % read in the first icesat-2 at108 csv file

zmod = T.Elevation(1:end-1); % save the 'model' elevations (icesat-2 elevations)

east = T.Easting(:); % pull out the easting values

```



```

norths = T.Northing(:); % pull out the northings
footwidth = 11; % approx. width of icesat2 shot footprint in meters

theta = zeros(size(norths)); % initialize empty matrices
xs = {};
ys = {};
xpoly = nan([1,5]);
ypoly = nan([1,5]);

for r = 1:length(theta)-1
    if strcmp(icesat2(end-44:end-40), 'ATL08') == 1 % ATL08 commands
        theta(r) = abs(atan((norths(r+1) - norths(r))/(easts(r+1) - easts(r)))); % trig to
get angle theta along-track

        % get the x and y vectors to form the polygon
        xpoly(1) = easts(r) + (footwidth/2) - footwidth/2*cos((pi/2) - theta(r)); %
calculate the 4 corners in the x direction
        xpoly(2) = easts(r) + (footwidth/2) + footwidth/2*cos((pi/2) - theta(r));
        xpoly(3) = easts(r) - (footwidth/2) + footwidth/2*cos((pi/2) - theta(r));
        xpoly(4) = easts(r) - (footwidth/2) - footwidth/2*cos((pi/2) - theta(r));
        xpoly = xpoly+A(1); % adjust by the easting offset
        xs{r} = [xpoly(1), xpoly(2), xpoly(3), xpoly(4), xpoly(1)]; % save the
corners as a vector in the x-es cell array

        ypoly(1) = norths(r) - 50 - footwidth/2*cos((pi/2) - theta(r)); % calculate the
4 corners in the y direction
        ypoly(2) = norths(r) - 50 + footwidth/2*cos((pi/2) - theta(r));
        ypoly(3) = norths(r) + 50 + footwidth/2*cos((pi/2) - theta(r));
        ypoly(4) = norths(r) + 50 - footwidth/2*cos((pi/2) - theta(r));
        ypoly = ypoly+A(2); % adjust by the nothing offset
        ys{r} = [ypoly(1), ypoly(2), ypoly(3), ypoly(4), ypoly(1)]; % save the
corners as a vector in the y-s cell array

    elseif strcmp(icesat2(end-44:end-40), 'ATL06') == 1 % ATL06 commands
        theta(r) = abs(atan((norths(r+1) - norths(r))/(easts(r+1) - easts(r)))); % trig to
get angle theta along-track

        % get the x and y vectors to form the polygon
        xpoly(1) = easts(r) + (footwidth/2) - footwidth/2*cos((pi/2) - theta(r)); %
calculate the 4 corners in the x direction
        xpoly(2) = easts(r) + (footwidth/2) + footwidth/2*cos((pi/2) - theta(r));
        xpoly(3) = easts(r) - (footwidth/2) + footwidth/2*cos((pi/2) - theta(r));
        xpoly(4) = easts(r) - (footwidth/2) - footwidth/2*cos((pi/2) - theta(r));
        xpoly = xpoly+A(1); % adjust by the easting offset

```

```

        xs{r} = [xpoly(1), xpoly(2), xpoly(3), xpoly(4), xpoly(1)]; % save the
        corners as a vector in the x-es cell array

        ypoly(1) = norths(r) - 20 - footwidth/2*cos((pi/2) - theta(r)); % calculate the
4 corners in the y direction
        ypoly(2) = norths(r) - 20 + footwidth/2*cos((pi/2) - theta(r));
        ypoly(3) = norths(r) + 20 + footwidth/2*cos((pi/2) - theta(r));
        ypoly(4) = norths(r) + 20 - footwidth/2*cos((pi/2) - theta(r));
        ypoly = ypoly+A(2); % adjust by the nothing offset
        ys{r} = [ypoly(1), ypoly(2), ypoly(3), ypoly(4), ypoly(1)]; % save the
        corners as a vector in the y-s cell array
    end
end

x =
R2.XWorldLimits(1)+0.5*R2.CellExtentInWorldX:R2.CellExtentInWorldX:R2.XWorld
Limits(end)-0.5*R2.CellExtentInWorldX; % get a vector of x coords
y =
R2.YWorldLimits(1)+0.5*R2.CellExtentInWorldY:R2.CellExtentInWorldY:R2.YWorld
Limits(end)-0.5*R2.CellExtentInWorldY; % get a vector of y coords

[xgrid, ygrid] = meshgrid(x, y); % create grids of each of the x and y coords
elevation_report = zeros([1, length(xs)]);

for t = 1:length(xs)
    xv = xs{t}; % bounding box x vector
    yv = ys{t}; % bounding box y vector

    % first trimming
    in = inpolygon(xgrid, ygrid, xv, yv); % get logical array of in values
    pointsinx = xgrid(in); % save x locations
    pointsiny = ygrid(in); % save y locations
    in2 = flip(in); % create a flipped in-grid (need row, column instead of column,
row)
    elevationsin = elevations(in2); % save elevations
    elevation_report(t) = nanmean(elevationsin);
end
ztruth = elevation_report(:); % create column vector

differences = zmod-ztruth; % calculate differences
differences(differences > 80) = nan; % toss bad points
differences(differences < -80) = nan; % toss bad points

```

```
rmsez = sqrt(nanmean((differences).^2)); % calculate rmsez
residuals = (zmod+offset)-ztruth; % calculate residuals
```

C. Downsampling.mlx

```
% clean downsampling live script. The goal is to read in csv files
% output by the las2csv.mlx live script, downsample those data,
% and turn them into an elevation raster at [binsize]-meter resolution

% created 12 September 2020 by C. Elkin

% last modified 3 October 2020. Code seems to be at reasonable
% efficiency. Got estimated time for RC raster creation down from
% ~86 days to ~4 hours

% most recent change: lines at bottom before export that fill in vertical
% and horizontal NaN striping

% this is the user input section.

% read in the desired folder with txt extensions. * is the wildcard operator
infolder =
'/Users/glaciologygroup/Desktop/elkin/ms_code/LASRead/output_test/rc_LAS/BIN*.txt';
% path to files

% mac (/) or PC (\) directory divider
divider = '/'; % input which divider your OS uses to separate directories

% set bin size (in meters, integer >= 1)
binsize = 2;

% path to a shapefile with the bounds of the watershed (UTM Coordinates)
shapepath =
'/Users/glaciologygroup/Desktop/elkin/ms_code/RC_shapefile/reynoldsck_utm.shp';
```

```

% desired output name for the final raster and csv products (same base name
% with different file extensions)
tiffname = 'RC_Downsampling_lastreturn.tif';
csvname = 'RC_downsampling_lastreturn.csv';

% the CRS EPSG Code for the projection of the output tiff.
% 32611 is the code for UTM zone 11 N
crs_code = 32611;

% this subsection sets the files for the outermost loop
files = dir(infolder); % call all of the files in the input folder name
numfiles = length({files.name}); % get the number of files in the directory for
looping
filenames = {files.name}; % create a cell array of just the file names
folder = {files.folder}; % create a cell array of just the path
folder = folder(1); % pull out the first path (each of the paths should be identical)
folder = folder{:}; % turn the cell variable into a character array

% this subsection sets up the initial grid for the watershed. It also sets
% up a NaNs array to store the eventual elevation values
shp = shaperead(shapepath); % read in the vectorized shapefile for the watershed
xlower = floor(nanmin(shp.BoundingBox(:, 1))); % get lower x value from the
read shapefile
xupper = ceil(nanmax(shp.BoundingBox(:, 1))) + binsize; % get upper x value
from the read shapefile
ylower = floor(nanmin(shp.BoundingBox(:, 2))); % get lower y value from the
read shapefile
yupper = ceil(nanmax(shp.BoundingBox(:, 2))) + binsize; % get upper y value
from the read shapefile

xbins = xlower:binsize:xupper; % set x bins for the watershed
ybins = ylower:binsize:yupper; % set y bins for the watershed

elevation_raster = nan([length(xbins), length(ybins)]); % set the elevation array

% this subsection is the looping section that reads in the files, averages
% the elevation values that fall within a given grid cell, and repeats for
% each grid cell.

```

```

tic % time the loop

for a = 1:numfiles
    filecall = filenames(a); % pull the a-th filename
    filecall = filecall{:}; % turn the cell variable into a character array
    infile = [folder, divider, filecall]; % create the character array pointing the loop
to which file to load

    elevationdata = readtable(infile); % read in the data
    elevations = table2array(elevationdata); % turn the table data to array data

    minx_in_file = nanmin(elevations(:,1)); % find the smallest x value in the
currently open bin
    maxx_in_file = nanmax(elevations(:,1)); % find the largest x value in the
currently open bin
    x_start_and_stop = find(xbins>=minx_in_file & xbins<ceil(maxx_in_file)); %
find bins that have data for this file

    test_count_z = 0; % count the number of elevations that get stored

    for i = 1:(length(x_start_and_stop)-1)

        elevations = elevationdata; % we only want to load the file in once, and this
keeps us from doing
        % it over and over again a bit further down

        elevations = table2array(elevations); % make the data an array instead of a
table
        Ix = find(elevations(:,1)>=xbins(x_start_and_stop(i)) &
elevations(:,1)<xbins(x_start_and_stop(i+1))); % find the x data inside the bin

        miny_in_file = nanmin(elevations(:,2));
        maxy_in_file = nanmax(elevations(:,2));
        y_start_and_stop = find(ybins>=miny_in_file & ybins<ceil(maxy_in_file));

        elevationsx = elevations(Ix,:); % subset to just the x indices of interest

        for j = 1:(length(y_start_and_stop)-1)
            elevations = elevationsx; % reset the elevations array to the x indices of
interest
            Iy = find(elevations(:,2)>=ybins(y_start_and_stop(j)) &
elevations(:,2)<ybins(y_start_and_stop(j+1))); % of the remaining values in
            % 'elevations', find the y indices falling within the y bin

```

```

elevations = elevations(Iy, :); % save only those data in the y bounds

zreport = nanmean(elevations(:,3)); % average and save the remaining
elevation values

if ~isnan(zreport) % only report if the zreport value is NOT a NaN
    test_count_z = test_count_z+1;
    if isnan(elevation_raster(x_start_and_stop(i),y_start_and_stop(j))) %
only report if the elevation_raster value IS a NaN
        elevation_raster(x_start_and_stop(i),y_start_and_stop(j)) = zreport;
% report the elevation into the raster
    end
end
end
end
end
% print the number of elevations recorded in boxes
fprintf("For file %3.0f the z count value was %3.0f.\n\n", a, test_count_z)

% save to be safe every 50 iterations through the loop
if isinteger(a/50)
    writematrix(elevation_raster, 'RC_downsampled.csv'); % save the elevations
as a csv every 50 files
    fprintf('\n The matrix has been saved as a csv file. \n')
end
end
% rotate the matrix 90 degrees. Somewhere in my loop I have my Xs and Ys
% confused and I don't want to try and figure out how to switch the
% scanning
true_elevations = rot90(elevation_raster);

% fill in the gaps in the matrix (interpolate in case of stripes of no data/NaN
values in final raster
[numrow, numcol] = size(true_elevations); % get number of rows and columns
for indrow = 2:(numrow-1)
    for indcol = 2:(numcol-1)
        if isnan(true_elevations(indrow, indcol)) % loop through and replace
horizontal nan bars with average from cell above and below
            true_elevations(indrow, indcol) = ((true_elevations(indrow-1, indcol) +
true_elevations(indrow+1, indcol))/2);
        end
        if isnan(true_elevations(indrow, indcol)) % now replace vertical nan bars
with average from cell left and cell right
            true_elevations(indrow, indcol) = ((true_elevations(indrow, indcol-1) +
true_elevations(indrow, indcol+1))/2);
        end
    end
end
end

```

end

```
% at the end of the loop save the elevation matrix as a csv file
writematrix(true_elevations, csvname);
```

```
toc % report the total elapsed time
Elapsed time is 23682.340072 seconds.
```

```
% export the matrix as a geotiff
R = maprefcells(); % create a map cells reference object
R.XWorldLimits = [xlower xupper]; % modify R to suit our needs using known
vars
R.YWorldLimits = [ylower yupper];
R.RasterSize = size(true_elevations);
R.ColumnsStartFrom = 'north';

geotiffwrite(tiffname,true_elevations,R,"CoordRefSysCode",crs_code)
```

D. LASReadLastReturn.m

```
function outfile = LASReadLastReturn(infile, outfile)
% LASREAD reads in files in LAS 1.1 format and outputs comma delimited text
files
% note: In order to make the program work for LAS 1.2 format go to
% line 40 and change it to "if VersionMajor ~= 1"
%
% INPUT
% infile: input file name in LAS 1.1 format
% (for example, 'myinfile.las')
```

```

% outfile: output file name in text format
%         (for example, 'myoutfile.txt')
% nFields: default value of 1 outputs X, Y and Z coordinates of the
%         point - [X Y Z].
%         A value of 2 gives Intensity as an additional attribute - [X Y Z I].
%         A value of 3 gives the Return number and the Number of returns
%         in addition to the above - [X Y Z I R N].
%
% OUTPUT
% outfile: the output matrix
%
% EXAMPLE
% A = LASRead ('infile.las', 'outfile.txt', 3)
%
% Cici Alexander
% September 2008 (updated 26.09.2008)
%
% Modifications 4 October 2020 by Colten Elkin:
% changed bottom few lines of script to only save last return.
% The object is to save only terrain (not canopy) elevations
% in order to create a digital terrain model using the downsampling.mlx
% live script. Contact coltenelkin@u.boisestate.edu for info.

% Open the file
fid =fopen(infile);

% Check whether the file is valid
if fid == -1
    error('Error opening file')
end

% Check whether the LAS format is 1.1
fseek(fid, 24, 'bof');
VersionMajor = fread(fid,1,'uchar');
VersionMinor = fread(fid,1,'uchar');
if VersionMajor ~= 1 %|| VersionMinor ~= 1
    error('LAS format is not 1.1')
end

% Read in the offset to point data
fseek(fid, 96, 'bof');
OffsetToPointData = fread(fid,1,'uint32');

% Read in the scale factors and offsets required to calculate the coordinates
fseek(fid, 131, 'bof');
XScaleFactor = fread(fid,1,'double');

```



```

YScaleFactor = fread(fid,1,'double');
ZScaleFactor = fread(fid,1,'double');
XOffset = fread(fid,1,'double');
YOffset = fread(fid,1,'double');
ZOffset = fread(fid,1,'double');

% The number of bytes from the beginning of the file to the first point record
% data field is used to access the attributes of the point data
%
c = OffsetToPointData;

% Read in the X coordinates of the points
%
% Reads in the X coordinates of the points making use of the
% XScaleFactor and XOffset values in the header.
fseek(fid, c, 'bof');
X1=fread(fid,inf,'int32',24);
X=X1*XScaleFactor+XOffset;
X = X(:); % turn vectors into column vectors for matrix generation below

% Read in the Y coordinates of the points
fseek(fid, c+4, 'bof');
Y1=fread(fid,inf,'int32',24);
Y=Y1*YScaleFactor+YOffset;
Y = Y(:); % column vector

% Read in the Z coordinates of the points
fseek(fid, c+8, 'bof');
Z1=fread(fid,inf,'int32',24);
Z=Z1*ZScaleFactor+ZOffset;
Z = Z(:); % column vector

% Read in the Intensity values of the points
fseek(fid, c+12, 'bof');
Int=fread(fid,inf,'uint16',26);
Int = Int(:); % column vector

% Read in the Return Number of the points. The first return will have a
% return number of one, the second, two, etc.
fseek(fid, c+14, 'bof');
Rnum=fread(fid,inf,'bit3',221);
Rnum = Rnum(:); % column vector
Rnum = round(Rnum); % turn into integers

```

```

% Read in the Number of Returns for a given pulse.
fseek(fid, c+14, 'bof');
fread(fid,1,'bit3');
Num=fread(fid,inf,'bit3',221);
Num = Num(:); % column vector
Num = round(Num); % turn into integers

counter = 1:length(X); % create indexing column for cutdown purposes
counter = counter(:); % column vector

matrix = [counter, X, Y, Z, Int, Rnum, Num]; % create matrix from values

% use logical indexing to eliminate rows where Rnum does not equal num
% (i.e. where the return is not the last return)

cutdown = matrix(matrix(:,6) == matrix(:,7)); % rows with wanted data

matrix = matrix(cutdown, [2 3 4]); % shrink matrix down to just X, Y, and Z of
last return

X = matrix(:,1); % create new X column vector with only wanted data
Y = matrix(:,2); % new Y column vector
Z = matrix(:,3); % new Z column vector

% Write out the file with X, Y and Z coordinates, representing only last
% returns

outfileheader = ['X' 'Y' 'Z'];
outfile = [X Y Z];

dlmwrite(outfilename,outfileheader);
dlmwrite(outfilename,outfile, '-append','precision','%0.2f','newline','pc');

```

E. Watershed Delineation

<https://giscrack.com/automatically-delineate-a-watershed-in-qgis/>

F. write_icesat2_csv.m

```

function write_icesat2_csv(inputdir, outputdir, shapefile)
% function WRITE_ICESAT2_CSV reads in an h5 file and outputs csv files of

```

```

% individual beams and their pertinent data
% INPUTS: inputdir = directory pointing to .h5 files (ends with '/') (string)
%         outputdir = directory where you want to save csv files (also ends
%                 with '/') (string)
%         shapefile = directory and name of shapefile of the region of
%                 interest that serves for clipping the icesat-2 data
%                 down (string)

% created 21 December 2020 by Colten Elkin (coltenelkin@u.boisestate.edu)
% requires matlab function deg2utm.m (available here:
% https://www.mathworks.com/matlabcentral/fileexchange/10915-deg2utm )
% also requires that path to deg2utm is added in the add path line below:
addpath('/Users/glaciologygroup/Desktop/elkin/ms_code/matlab_scripts'); % add
path for calling deg2utm later

% last edited: 25 January 2021
% most recent update: made to work with both ATL08 and ATL06

if(~isfolder(outputdir)) % create the output directory if it doesnt already exist
    mkdir(outputdir)
end

watershed = shaperead(shapefile);
h5files = dir([inputdir,'*.h5']); % pull out the h5 files
beams = {'gt1r', 'gt1l', 'gt2r', 'gt2l', 'gt3r', 'gt3l'}; % list of icesat2 beams for inner
loop
for filecounter = 1:length(h5files) % loop through icesat2 files
    % check to see whether it's ATL08 or ATL06
    if strcmp(h5files(filecounter).name(1:5), 'ATL08') == 1 % ATL08 commands
        h5test = [h5files(filecounter).folder,'/',h5files(filecounter).name]; % get
string pointing to n-th h5 file

        for beamcount = 1:length(beams)
            beam = beams{beamcount}; % set beam
            % read in data
            terrain = h5read(h5test, ['/',beam,'/land_segments/terrain/h_te_mean']); %
read terrain means
            lat = h5read(h5test, ['/',beam,'/land_segments/latitude']); % read lats
            lon = h5read(h5test, ['/',beam,'/land_segments/longitude']); % read lons
            bright = h5read(h5test, ['/',beam,'/land_segments/brightness_flag']); %
read in brightness of shot
            std = h5read(h5test, ['/',beam,'/land_segments/terrain/h_te_std']); %
standard deviation
            can = h5read(h5test, ['/',beam,'/land_segments/canopy/h_canopy']); %
canopy elevation

```

```

% crop data to area of interest
% first, crude cropping
lonlims = watershed.BoundingBox(:, 1); % save upper and lower
longitudes of the watershed
latlims = watershed.BoundingBox(:, 2); % save upper and lower latitudes
of the watershed

% note: this trimming process is ~4x faster than just using inpolygon
Ix = find(lon > min(lonlims) & lon < max(lonlims)); % find longitudes
between limits
lon = lon(Ix); % cut down based on longitudes
lat = lat(Ix);
terrain = terrain(Ix);
bright = bright(Ix);
std = std(Ix);
can = can(Ix);

Iy = find(lat > min(latlims) & lat < max(latlims)); % find lats between
limits
lat = lat(Iy); % cut down based on latitudes
lon = lon(Iy);
terrain = terrain(Iy);
bright = bright(Iy);
std = std(Iy);
can = can(Iy);

% now do a final trip to the actual watershed bounds
wats = [watershed.X', watershed.Y']; % save coordinate tuples from the
watershed shapefile
Ifinal = inpolygon(lon, lat, wats(:,1), wats(:,2));

lat = lat(Ifinal); % save data as vectors after final clipping

if ~isempty(lat) % continue if data is inside the region of interest
lon = lon(Ifinal); % continue saving data
terrain = terrain(Ifinal);
bright = bright(Ifinal);
std = std(Ifinal);
can = can(Ifinal);
can(can > 1000) = nan; % change canopy elevation no data value to nan

% use deg2utm script to write easting and northing values
[easts, norths] = deg2utm(lat, lon);

```

```

    % create a final structure with all of the data
    s = struct(); % set structure for fields
    s.Latitude = lat; % set lats
    s.Longitude = lon; % set lons
    s.Elevation = terrain; % fill additional fields
    s.Elevation(s.Elevation >= 10e20) = NaN; % set Nans (icesat2 default
nan value is 4.028e38)
    s.Canopy = can;
    s.std = std;
    s.Easting = easts;
    s.Northing = norths;
    s.Brightness_Flag = bright;

    table = struct2table(s); % convert to a table
    h5filename = h5test(end-38:end-3); % save h5 filename
    outputname = [outputdir, h5filename, '_', beam, '.csv']; % save full
filename
        writetable(table, outputname)
    end
end
% if not ATL08, ATL06?
elseif strcmp(h5files(filecounter).name(1:5), 'ATL06') == 1 % enter ATL06
commands
    h5test = [h5files(filecounter).folder, '/', h5files(filecounter).name]; % get
string pointing to n-th h5 file
    for beamcount = 1:length(beams)
        beam = beams{beamcount}; % set beam
        % read in data
        terrain = h5read(h5test, [ '/', beam, '/land_ice_segments/h_li']); % read
terrain means
        lat = h5read(h5test, [ '/', beam, '/land_ice_segments/latitude']); % read lats
        lon = h5read(h5test, [ '/', beam, '/land_ice_segments/longitude']); % read
lons
        bright = h5read(h5test, [ '/', beam, '/land_ice_segments/sigma_geo_h']); %
read in vertical geolocation error
        std = h5read(h5test, [ '/', beam, '/land_ice_segments/h_li_sigma']); %
standard deviation

        % crop data to area of interest
        % first, crude cropping
        lonlims = watershed.BoundingBox(:, 1); % save upper and lower
longitudes of the watershed
        latlims = watershed.BoundingBox(:, 2); % save upper and lower latitudes
of the watershed

```

```

% note: this trimming process is ~4x faster than just using inpolygon
Ix = find(lon > min(lonlims) & lon < max(lonlims)); % find longitudes
between limits
lon = lon(Ix); % cut down based on longitudes
lat = lat(Ix);
terrain = terrain(Ix);
bright = bright(Ix);
std = std(Ix);

Iy = find(lat > min(latlims) & lat < max(latlims)); % find lats between
limits
lat = lat(Iy); % cut down based on latitudes
lon = lon(Iy);
terrain = terrain(Iy);
bright = bright(Iy);
std = std(Iy);

% now do a final trip to the actual watershed bounds
wats = [watershed.X', watershed.Y']; % save coordinate tuples from the
waterhsed shapefile
Ifinal = inpolygon(lon, lat, wats(:,1), wats(:,2));

lat = lat(Ifinal); % save data as vectors after final clipping

if ~isempty(lat) % continue if data is inside the region of interest
lon = lon(Ifinal); % conitnue saving data
terrain = terrain(Ifinal);
bright = bright(Ifinal);
std = std(Ifinal);

% use deg2utm script to write easting and northing values
[easts, norths] = deg2utm(lat, lon);

% create a final structure with all of the data
s = struct(); % set structure for fields
s.Latitude = lat; % set lats
s.Longitude = lon; % set lons
s.Elevation = terrain; % fill additional fields
s.Elevation(s.Elevation >= 10e20) = NaN; % set Nans (icesat2 default
nan value is 4.028e38)
s.std = std;
s.Easting = easts;
s.Northing = norths;
s.Vert_Geo_error = bright;

```

```
        table = struct2table(s); % convert to a table
        h5filename = h5test(end-38:end-3); % save h5 filename
        outputname = [outputdir, h5filename, '_', beam, '.csv']; % save full
filename
        writetable(table, outputname)
        end
    end
end
end
```

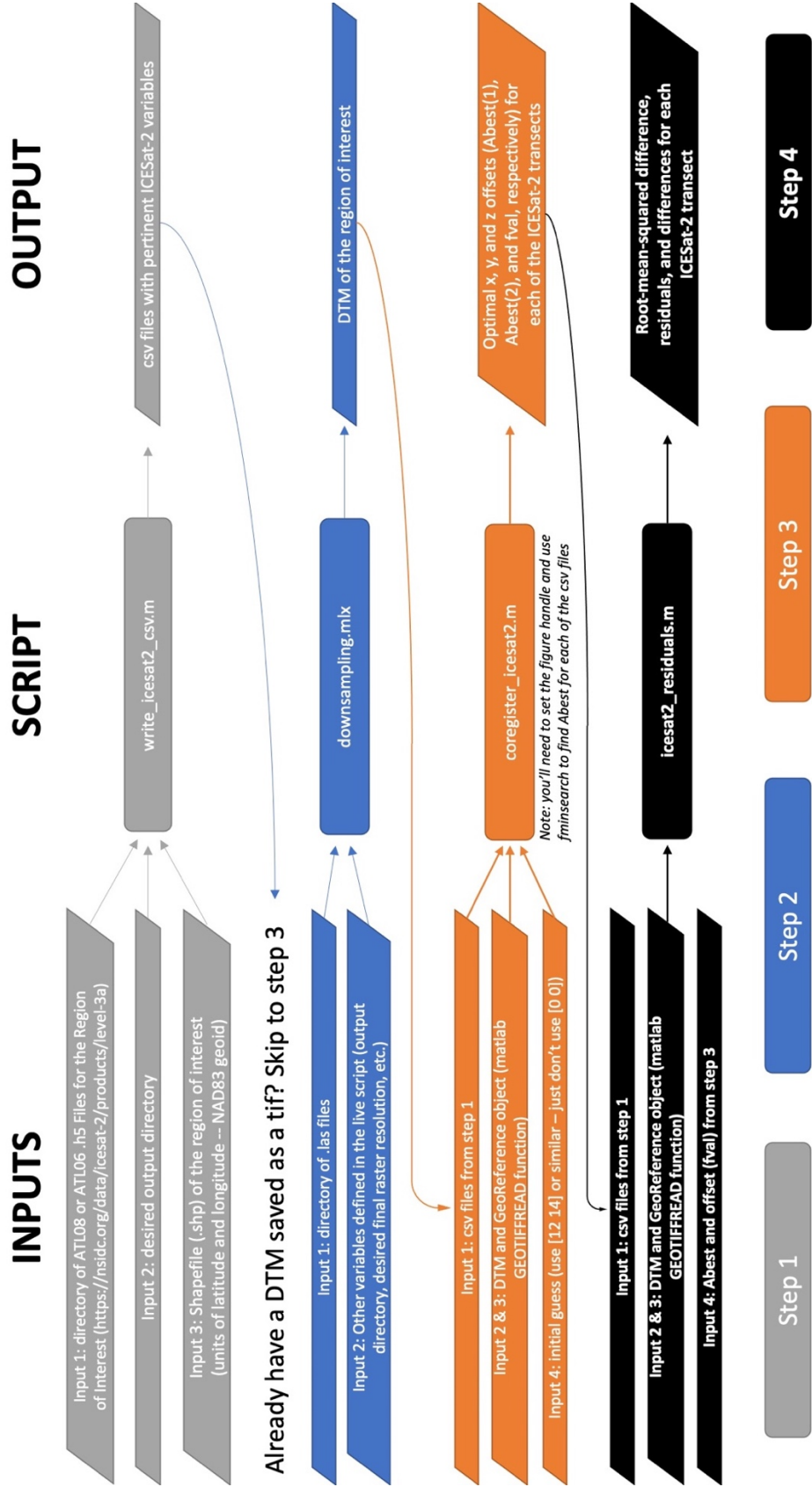


Figure A.1. A simplified flowchart of code steps used for coregistration of ICESat-2 transects. For a complete user guide, please contact the author Colfen Elkin at colfenelkin@u.boisestate.edu.