

Master's thesis

Snow depth estimation in forested terrain using the ICESat-2 space laser

Version:

Thursday 2nd May, 2024

22:54

Simen Dalseid Aune

Geomorphology and Geomatics
60 ECTS credits

Department of Geosciences
Faculty of Mathematics and Natural Sciences

Spring 2024



Todo list

Don't touch the abstract until the end	i
Write acknowledgements	iii
Finish the background section. Describe what has been done, and why the research question is relevant. Why are forests challenging? Annual snow cover vs. forest. NMA LiDAR in fall.	2
Put literature study here. Describe point classification vs. photon classification? Geoid vs. ellipsoid? Other geodetic problems? Green laser -> Atmospheric window	4
Check some papers? -> Snow general -> Forest lidar. Literature review in a separate section? Needs to be several pages long for a thesis. Good intro on difficulty of measuring/modelling snow in forests.	4
Describe the instruments. Capabilities and limitations. Remember that examiners rarely know both data sources, so be thorough.	6
How sites were chosen. Delay between collections. Looking at cloud coverage (EUMETSAT...)	9
Add more criteria to the table	9
Describe somewhere how long winter lasts in Norway?	9
Describe snow cover	11
Add scale bar to images.	12
Photons vs. points. Main differences. Why so many DEMs? Max. min. returns - Difference and influence?	20
Show an example! Photon plot?	20
Cut the bulk of this section in favour of background/methods/res- ults. Talk about tools _relevant_ to your research questions.	22
Why was it used? See it as accompanying L1, not a separate thing.	22
Really?	22
Shorten most of the libraries into one. Keep xDEM and SlideRule. Generic libraries can be mentioned in processing?	23

Keep Pandas/GeoPandas, xDEM, SlideRule. Rest in context of postprocessing	23
Finish describing each processing step. Describe more what I actually did, rather than what is possible. Less abstract. Show results along the way. Flow chart?	25
You haven't yet said how you will be using this data	25
Describe these so it's clear you understand LiDAR	25
What software?	25
Again, not described anywhere else	25
No. Point clouds are discrete, DTM/CHM are continuous	25
Why?	26
Why?	26
You don't have a research question, and your problem statement doesn't mention canopy heights	26
You haven't introduced why you need these two datasets	26
How are you calculating canopy density/cover?	26
Why not ATL08 veg classifications? At least discuss	27
Sketch!	28
Rewrite to make it easier to understand. Reconsider the plot...	28
Remember to only show results. Do not discuss here	31
Describe how each filtering method performs, and relevant influences on results. Add transect plots showing the effect of each method	32
Describe results for each site. How are they varying? Mention the influences on-site if relevant. Add transects.	33
Re-check if beam is relevant	33
Write the entire discussion chapter...	40
Performance of algorithms? -> Difficulty to get validation data (field work planning (RE: Jevnaker))? -> ATL08 veg classification? Possibility to apply this at larger scale? Correlation evaluation also. Recommended next steps?	40
Mention max returns from M300 (3). Discuss canopy cover threshold (2m)	40
Is the research question answered?	44
How can these findings be used, and what needs to be improved or further explored/tested	45

Simen Dalseid Aune

Snow depth estimation in forested terrain using the ICESat-2 space laser

Version:

Thursday 2nd May, 2024

22:54

Supervisors:
Désirée Treichler
Clare Webster

Abstract

NASA's Ice, Cloud, and land Elevation Satellite 2 (ICESat-2) provides a way to measure heights at a global scale, at a minimum of 90-day intervals, with high resolution. An existing project exists to assess snow depths using these measurements. This thesis builds on the work in that project, and attempts to validate these measurements in areas with vegetation. UAV-borne LiDAR datasets have been collected in the chosen study sites, to generate high-resolution terrain models and canopy height models. These models are used to verify the photon heights from the ATLAS instrument, which have been filtered with one of three different methods:

1. Percentile-based threshold distance from a ground-surface DTM
2. A simple Nearest-neighbour averaging algorithm
3. Yet Another Photon Classifier (YAPC) algorithm

The output from the three methods are then compared to each other, to evaluate their efficiency in filtering out noise and stray photons.

Don't touch the abstract until the end

Acknowledgement

Write
acknowledgements

Acknowledgement

Contents

1	Introduction	1
1.1	Background	2
1.2	Research question	3
1.3	Theory	4
1.3.1	Geodesy?	4
1.3.2	LiDAR.	4
1.3.3	Forest mapping	4
1.3.4	Snow mapping	4
2	Data acquisition	5
2.1	Instrumentation	6
2.1.1	ICESat-2.	6
2.1.2	DJI Zenmuse L1	8
2.2	UAV data acquisition	9
2.2.1	The actual site selection process	10
2.3	Field sites	11
2.3.1	Drammen	12
2.3.2	Hof	14
2.3.3	Jevnaker.	15
2.3.4	Vikerfjell	16
2.4	Data	20
2.4.1	ICESat-2 products	20
2.4.2	UAV-borne LiDAR point clouds	20
2.4.3	Norwegian elevation data	20
3	Data processing	21
3.1	Software	22
3.1.1	DJI Terra.	22
3.1.2	Jupyter	22
3.1.3	R	22
3.1.4	Python	23
3.2	Processing	25
3.2.1	UAV data	25
3.2.2	Data structuring and metrics extraction	26
3.2.3	Canopy height model	26
3.2.4	Photon filtering methods	27
3.2.5	Snow depth calculations	30
3.2.6	Bias estimation	30
3.2.7	Canopy vs. bias correlation.	30

Contents

4	Results	31
4.1	Performance of filtering methods	32
4.1.1	Threshold validation	32
4.1.2	Point grouping.	32
4.1.3	Yet Another Photon Classifier	32
4.2	Per field site.	34
4.2.1	Drammen	34
4.2.2	Hof.	35
4.2.3	Jevnaker.	36
4.2.4	Vikerfjell East	37
4.2.5	Vikerfjell West.	38
5	Discussion	39
5.1	Application	40
5.2	Workflow Review.	41
5.3	Software Review	42
6	Conclusion	43
6.1	Main findings	44
6.2	Applications and future work	45

List of Figures

2.1	Global map showing where snow is expected to be present during winter, and the minimum and maximum latitudes of the GEDI mission for reference. (Snow data from MODIS, August 2023 and January 2024.)	6
2.2	Spatial pattern of the ATLAS beams and footprints. From T. A. Neumann et al. (2023)	7
2.3	Artist rendition of the ICESat-2 satellite platform. ©NASA	7
2.4	DJI Matrice 300 RTK UAV with DJI Zenmuse L1 on a gimbal mount. (Image from DJI.com, accessed on 2024-04-09)	8
2.5	Overview map of Southern Norway, with each field site represented by red dots. The red lines show the ICESat-2 ground tracks used in the thesis.	11
2.6	The site in Drammen municipality..	12
2.7	Sample image from the UAV flight, showing the canopy structure, covered in snow.	13
2.8	The site in Hof municipality.	14
2.9	Sample image from the UAV flight, showing the canopy structure, covered in snow.	15
2.10	The study site in Jevnaker municipality.	16
2.11	The eastern part of the site in Vikerfjell.	17
2.12	The western part of the site in Vikerfjell.	18
2.13	Sample image from the UAV flight in Vikerfjell, showing no snow cover on the canopy structure.	19
3.1	A visualisation of the change in the maximum acceptable snow depth value, as a function of the percentile value, as performed in <i>eval_ph</i>	29
3.2	A sample set of photons from the Hof site, before and after the threshold validation is applied as described in <i>eval_ph</i>	29
4.1	The distribution of photon bias after the filtering methods are applied, per area. Median bias is indicated by lines.	32

List of Figures

List of Tables

2.1	Orbital parameters of ICESat-2	8
2.2	DJI Zenmuse L1 sensor properties (From dji.com)	8
2.3	The main criteria affecting the field site selection for ICESat-2 and UAV data acquisition.	9
2.4	The field sites with their geodetic coordinates and times of UAV and ICESat-2 data acquisition. (All times are in the local timezone.)	10
4.1	Statistics per method, Drammen (Both beams)	34
4.2	Statistics per method, Hof (Both beams)	35
4.3	Statistics per method, Jevnaker (Both beams)	36
4.4	Statistics per method, Vikerfjell E (Both beams)	37
4.5	Statistics per method, Vikerfjell W (Both beams)	38

List of Tables

Chapter 1

Introduction

Finish the background section. Describe what has been done, and why the research question is relevant. Why are forests challenging? Annual snow cover vs. forest. NMA LiDAR in fall.

1.1 Background

Repeated measurements of snow depths – at large scale – can serve a multitude of purposes in Geosciences. They can provide estimates of available hydropower and drinking water resources; As time series spanning several years they can serve as an indicator of climate change; Given high spatio-temporal resolution they can even help predict the risk of avalanches. In turn, decision makers are able to make better decisions in the future. However, gathering the data from the field is resource demanding, and practically only a viable option in wealthy, developed countries. Developing countries with increasing energy demand, sometimes combined with challenges brought on by climate change, generate the need for low-cost methods of supervising the amounts of snow in large catchments.

Satellite-born sensors may provide one solution to this problem. The Ice, Cloud and land Elevation Satellite (ICESat-2) mission carries an instrument called the Advanced Topographic Laser Altimeter System (ATLAS), with a near-global coverage and 91-day repeat period (Thomas A. Neumann et al. 2019). The instrument has six beams of a green laser (532 nm), using the time of flight for each photon to calculate the distance to the surface.

1.2 Research question

Several methods have been developed to measure heights using the ICESat-2 ATLAS instrument in forests [CITATION NEEDED], and some studies have looked into its ability to accurately measure snow surfaces [CITATION NEEDED]. Few — if any — attempts have been made to combine the two, and evaluate the effect forest canopies can have on the instruments ability to measure the surface in snow-covered terrain. Identifying an effective method of filtering out non-surface photon returns could improve the precision of the measurements.

This thesis will assess the ability of the ATLAS instrument to measure surface heights in dense vegetation and forest canopies, when the surface is covered in snow. It will also explore methods for filtering out noise from the data, with the purpose of increasing accuracy. The research question for the thesis is therefore:

How accurately does the ATLAS instrument on ICESat-2 measure snow surface heights in forested areas, and what methods are effective for filtering out non-surface photon returns?

1.3 Theory

This section will provide the theoretical background for the thesis. Relevant fields of research include geodesy, forest mapping, snow surface mapping and Light Detection And Ranging (LiDAR).

1.3.1 Geodesy?

1.3.2 LiDAR

Photon-counting laser altimetry

Full-waveform LiDAR

1.3.3 Forest mapping

Space-borne laser altimeters are a relatively

1.3.4 Snow mapping

Put literature study here.
Describe point classification vs. photon classification?
Geoid vs. ellipsoid? Other geodetic problems?
Green laser -> Atmospheric window

DES

Check some papers? -> Snow general -> Forest lidar. Literature review in a separate section? Needs to be several pages long for a thesis.
Good intro on difficulty of measuring/modelling snow in forests.

Chapter 2

Data acquisition

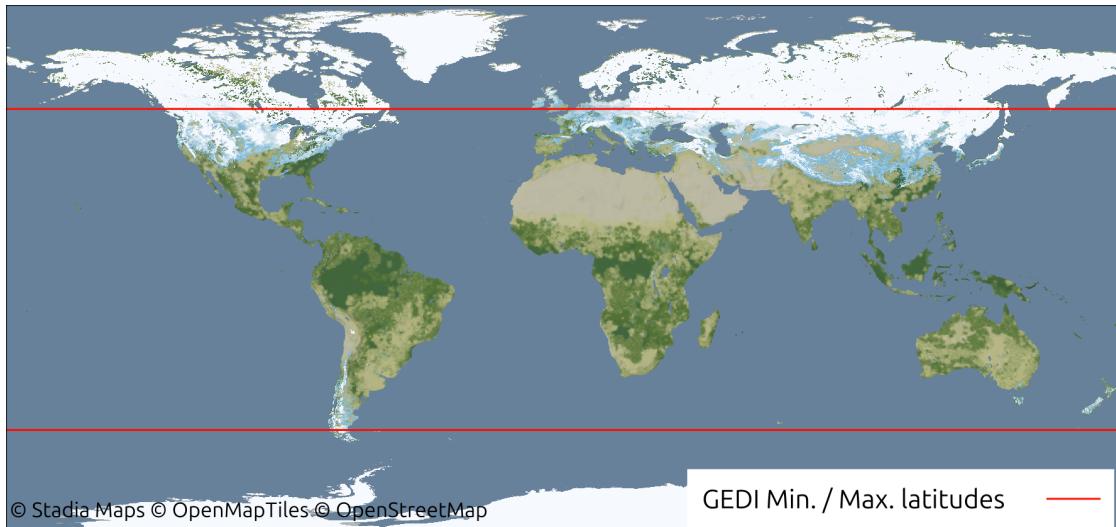


Figure 2.1: Global map showing where snow is expected to be present during winter, and the minimum and maximum latitudes of the GEDI mission for reference. (Snow data from MODIS, August 2023 and January 2024.)

Describe the instruments. Capabilities and limitations. Remember that examiners rarely know both data sources, so be thorough.

2.1 Instrumentation

The following section will describe the instruments used in the acquisition of data for this thesis — The ICESat-2 system and the DJI Zenmuse L1 LiDAR device. A short description of the platforms are included, but the main focus will be on the sensors themselves. Data from other providers are described in section 2.4.

2.1.1 ICESat-2

All platform and sensor information in this subsection is referenced from the ICESat-2 system description (Thomas A. Neumann et al. 2019), unless otherwise specified. The *Ice Cloud and Land Elevation Satellite - 2* (ICESat-2) mission is a part of *NASA's Earth Observing System* (2024). As the number suggests, it is the second iteration of a mission for this purpose, with the first mission (ICESat) lasting from 2003 to 2010. ICESat-2 consists of a single satellite in a near-polar orbit, carrying one instrument — the Advanced Topographic Laser Altimeter System (ATLAS), which is a photon-counting laser altimeter. Laser altimetry from space-born sensors is not widely used, with the Global Ecosystem Dynamics Investigation (GEDI) mission being the only other option besides the two already mentioned. GEDI is mounted aboard the International Space Station (ISS), and its current applications are solely within biosphere and hydrosphere monitoring (Magruder et al. 2024). The ISS orbit is also limited to 51.6°N and 51.6°S, which does not cover the polar regions (See fig. 2.1).

Sensor properties

In the ATLAS instrument, a single pulse of green (532 nm) laser is split into a total of six beams — in three pairs — which are directed toward the Earth surface. In each beam pair, one beam is stronger than the other, meaning that it has more energy. The ratio of energy between then two beams is 1:4. The lower-energy beam is thus referred to as the *weak beam*, while the other is the *strong beam*. The different energy output in turn means that the strong beam nominally provides four times as many photon returns

as the weak beam. The entire sensor is slightly rotated, and the beams hit the surface with a time delta of approximately CITEMATION NEEDED: Find the time delta

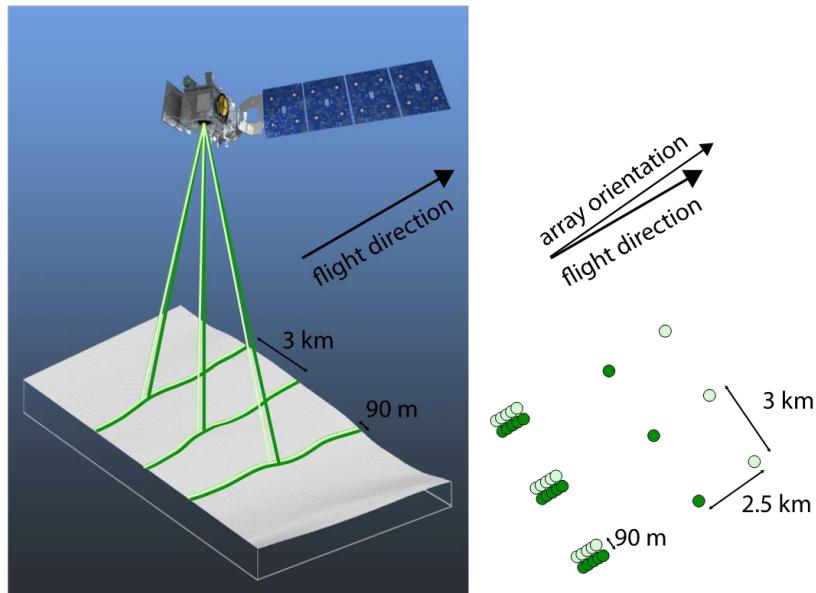


Figure 2.2: Spatial pattern of the ATLAS beams and footprints. From T. A. Neumann et al. (2023)

Platform properties

The ICESat-2 platform is a LEOStar-3 satellite bus, carrying the ATLAS as its sole instrument. It was launched on 15 September 2018 from Vandenberg Space Force Base in California, USA. The planned mission duration was for 3 years, but it is still fully operational.



Figure 2.3: Artist rendition of the ICESat-2 satellite platform. ©NASA

Table 2.1: Orbital parameters of ICESat-2

Reference system:	Geocentric
Regime:	Low Earth Orbit (LEO)
Nominal altitude:	500 km
Inclination:	92°
Period:	94.22 min
Velocity:	6.9 km/s
Repeat period:	91 days

2.1.2 DJI Zenmuse L1



Figure 2.4: DJI Matrice 300 RTK UAV with DJI Zenmuse L1 on a gimbal mount. (Image from [DJI.com](https://www.dji.com), accessed on 2024-04-09)

Sensor properties

The DJI Zenmuse L1 is a gimbal-mounted LiDAR sensor capable of high accuracy

Table 2.2: DJI Zenmuse L1 sensor properties (From [dji.com](https://www.dji.com))

Operating temperature:	-20° C to 50° C
Maximum Returns Supported:	3
Ranging Accuracy (RMS 1 σ) ² :	3 cm @ 100 m
Flight time:	\approx 25 minutes (per battery pair)
Point rate:	Single return: max. 240,000 pts/s Multiple return: max. 480,000 pts/s

Platform properties

The Zenmuse L1 requires a specific UAV platform to function: The DJI Matrice 300 RTK. As the name suggests, this UAV is capable of Real-Time Kinematic (RTK) positioning. RTK enables up to a centimetre-level accuracy in real time by using either virtual or physical base stations. Because the expected bias was in the sub-decimetre level, lower-precision positions would reduce the output quality of the further analyses.

2.2 UAV data acquisition

Near-simultaneous acquisition of data from both UAV and the ICESat-2 satellite would yield optimal results for comparative analyses, but it also requires meticulous planning and constant monitoring of the weather. This section will describe the different factors affecting the acquisition, and how the actual acquisition was eventually conducted. The site selection was essentially a multi-criteria analysis, with the most important criteria shown in table table 2.3.

Table 2.3: The main criteria affecting the field site selection for ICESat-2 and UAV data acquisition.

Criterion	Type	Limit	Affects
Wind speed	Hard	< 15 m/s	DJI M300
Temperature	Hard	> -20°C	DJI M300
Cloud cover	Hard	< 1 okta	ICESat-2
Snow depth	Hard	> 0.5 m	ICESat-2
Driving distance	Soft	< 3 hrs	Personnel
Daylight	Soft	Present	Personnel

The main challenge in selecting the field sites was timing. The ICESat-2 orbit yields a 91-day exact repeat interval, meaning that each potential site could only be used for one satellite overpass — or two if the ascending and descending tracks intersected there — per season. The weather affects both instruments in different ways, and if the conditions are not right when the satellite passes the site, the data collection would not yield useful results. The beams of the ATLAS instrument could not penetrate clouds, but clouds are prevalent during the winter season. Meanwhile, the UAV was unable to operate if the wind speed surpasses 15 m/s. Even if it is capable of flying in wind speeds approaching that, it requires more energy to operate, which in turn reduces the flight time per battery.

The weather could potentially also alter the snow properties in just a few hours if the temperature approaches or exceeds the snow melting point at 0 °C. Ensuring similar snow properties in the two resulting datasets therefore depended on either simultaneous acquisition or stable, cold weather in the period in-between. In the latter case, wind could be accepted at the ICESat-2 acquisition time and cloud cover at the UAV acquisition time. Low temperatures would in turn reduce battery life on the UAV. Weather predictions with enough precision to sufficiently guarantee acquisition are only available a few days in advance.

Another main factor was the logistics of the acquisition. The UAV and its pilots were located in Oslo, and manually moving the equipment over distances was cumbersome due to large size and heavy weight. Vehicular access to the field site and relatively short driving distance was therefore considered a requirement. At the same time, vehicular access is mostly available near people and infrastructure, where UAV flying would not be possible due to the drone operating rules of Norway, without special permission. The default safety distance to populated areas is defined as 150 m. Additional no-fly zones exist around airports and objects relevant to national security.

A high readiness and flexibility in deciding on which field sites to use was therefore paramount. For practical and logistical reasons, this also meant that the study area would need to be close to where the equipment and people were, i.e. within a few hours drive from Oslo.

DES
How sites were chosen.
Delay between collections.
Looking at cloud coverage (EUMETSAT...)

Add more criteria to the table

Describe somewhere how long winter lasts in Norway?

In addition to all the mentioned factors, the chosen field sites would need to have a snow depth high enough to actually be measurable by the ATLAS instrument and a significant forest canopy structure. Previous research on the vertical accuracy of ICESat-2 measurements indicated that at least 0.5 m, but preferably more than 1.0 m of snow depth would be enough to properly answer the research question.

2.2.1 The actual site selection process

During the winter season, the process to assess field site suitability ended up as a constant monitoring workflow:

1. Check if ICESat-2 ground tracks were planned within reasonable driving distance in the next 14 days
2. Identify potential sites along the track, with a suitable snow depth and canopy structure, while also being accessible with a vehicle
3. Check that the predicted weather and expected cloud cover for the sites would allow data acquisition
4. Make practical preparations for field work (charge batteries, reserve UAV and vehicle, etc.)
5. Monitor the weather forecast until the ICESat-2 overpass
6. Depending on the weather forecast
 - If no cloud cover was forecast at time of overpass: Acquire data with UAV as close to the overpass as practically possible
 - If low cloud cover was forecast, remain prepared to go immediately after the overpass in case the clouds did not actually appear.

At the time of overpass, the Cloud Mask data product from EUMETSAT was used to confirm whether the ICESat-2 could have acquired data.

Over a period of three months, between January and April of 2023, a total of four field sites were eventually used, as seen in table 2.4. The sites are described further in section 2.3. The table also shows that the largest time difference between UAV and ICESat-2 acquisition was just below 48 hours, at the site in Drammen.

Table 2.4: The field sites with their geodetic coordinates and times of UAV and ICESat-2 data acquisition. (All times are in the local timezone.)

Site	Latitude	Longitude	ICESat-2	UAV
Drammen	9.9843°	59.7882°	2023-01-21 13:12	2023-01-23 16:05
Hof	10.0352°	59.5191°	2023-01-21 13:12	2023-01-23 12:08
Jevnaker	10.4914°	60.2416°	2023-04-19 22:24	2023-04-19 19:58
Vikerfjell East	9.9773°	60.5021°	2023-02-19 11:48	2023-02-19 12:30
Vikerfjell West	9.9726°	60.5021°	2023-02-21 00:12	2023-02-19 12:30

2.3 Field sites

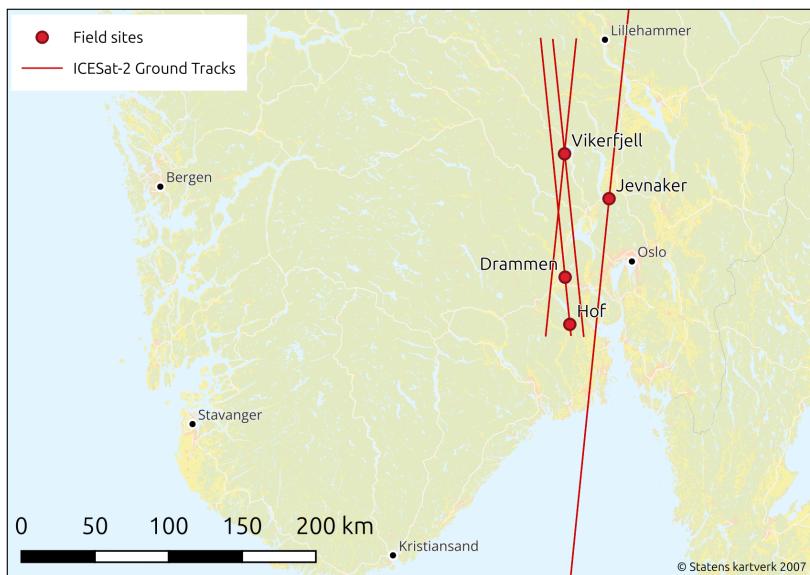


Figure 2.5: Overview map of Southern Norway, with each field site represented by red dots. The red lines show the ICESat-2 ground tracks used in the thesis.

The sample sites used in this thesis all lie to the west of Oslo (fig. 2.5). A total of four sites were chosen, all containing varying slope and canopy cover. One of the chosen sites, in Jevnaker, ended up with a low spatial overlap between the UAV and ICESat-2 data, essentially only covering the weak beam. The canopy structure at all the sites mostly consisted of large (>10 m) coniferous trees like spruce and pine, with a insignificant presence of deciduous trees.

Describe snow cover

2.3.1 Drammen

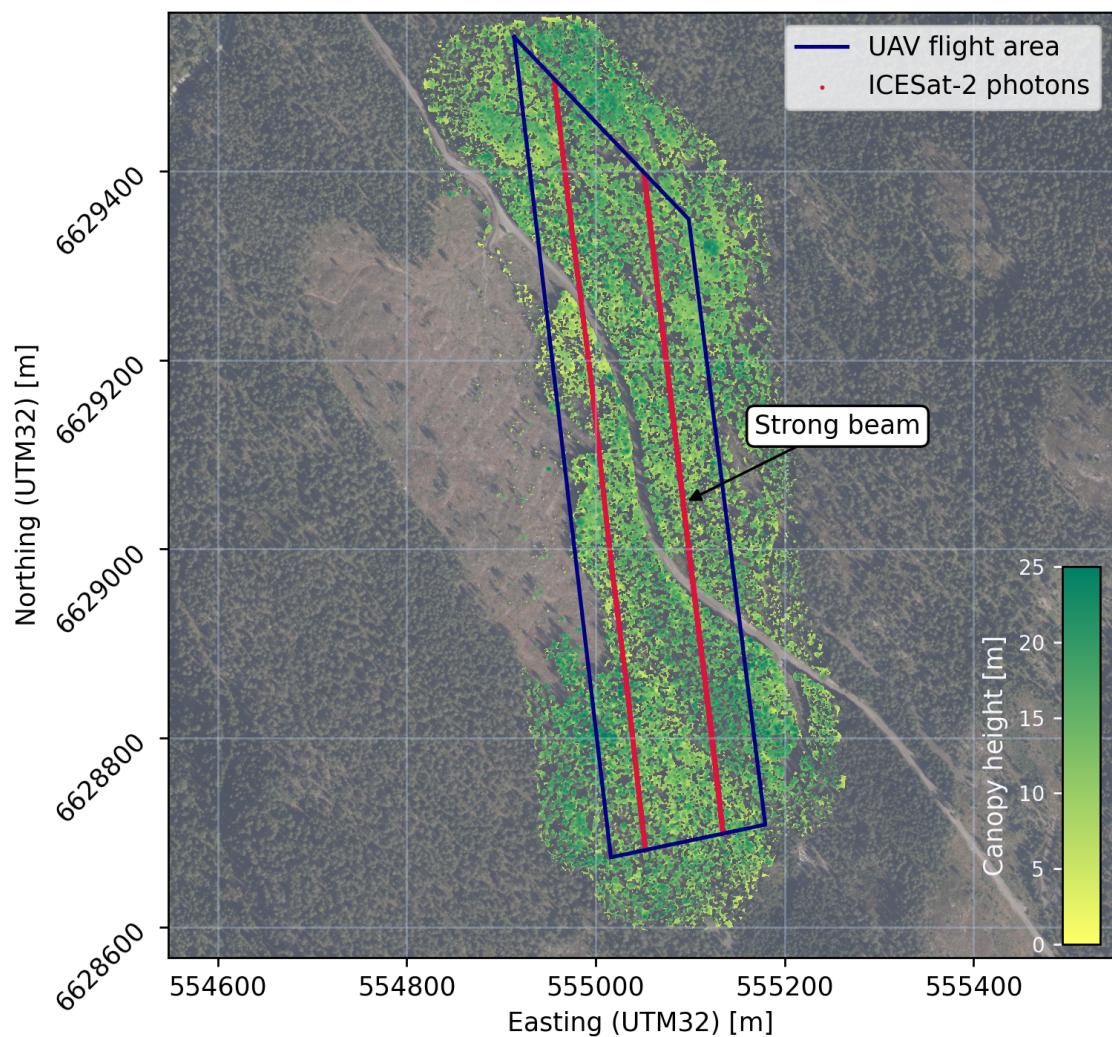


Figure 2.6: The site in Drammen municipality.

The first site lies in the western part of Drammen municipality, just north of Krokstadelva.

Add scale bar to
images.

2.3. Field sites



Figure 2.7: Sample image from the UAV flight, showing the canopy structure, covered in snow.

2.3.2 Hof

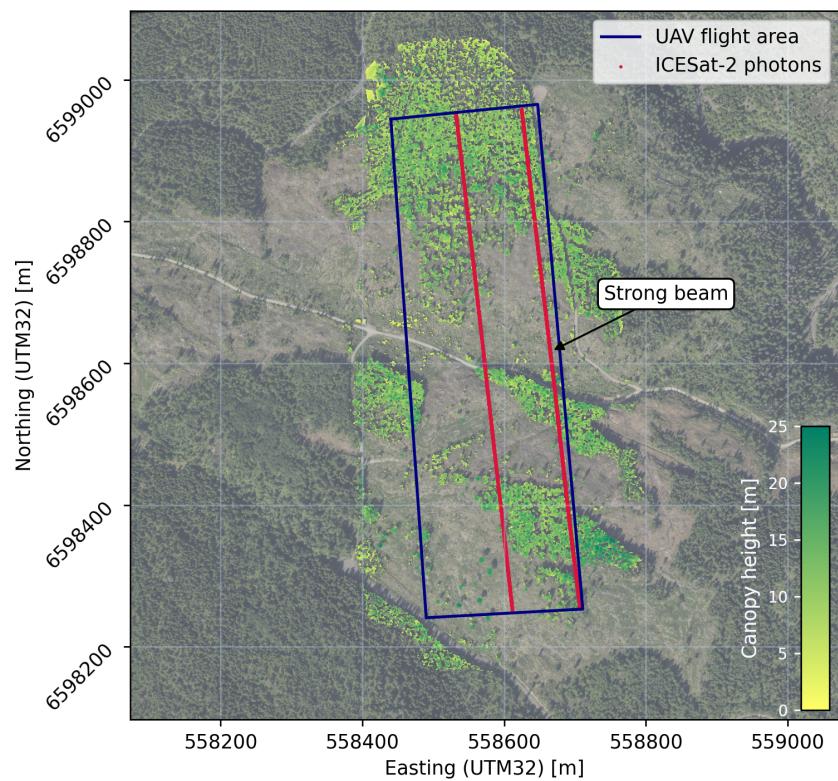


Figure 2.8: The site in Hof municipality.

Describe Hof



Figure 2.9: Sample image from the UAV flight, showing the canopy structure, covered in snow

2.3.3 Jevnaker

Describe Jevnaker

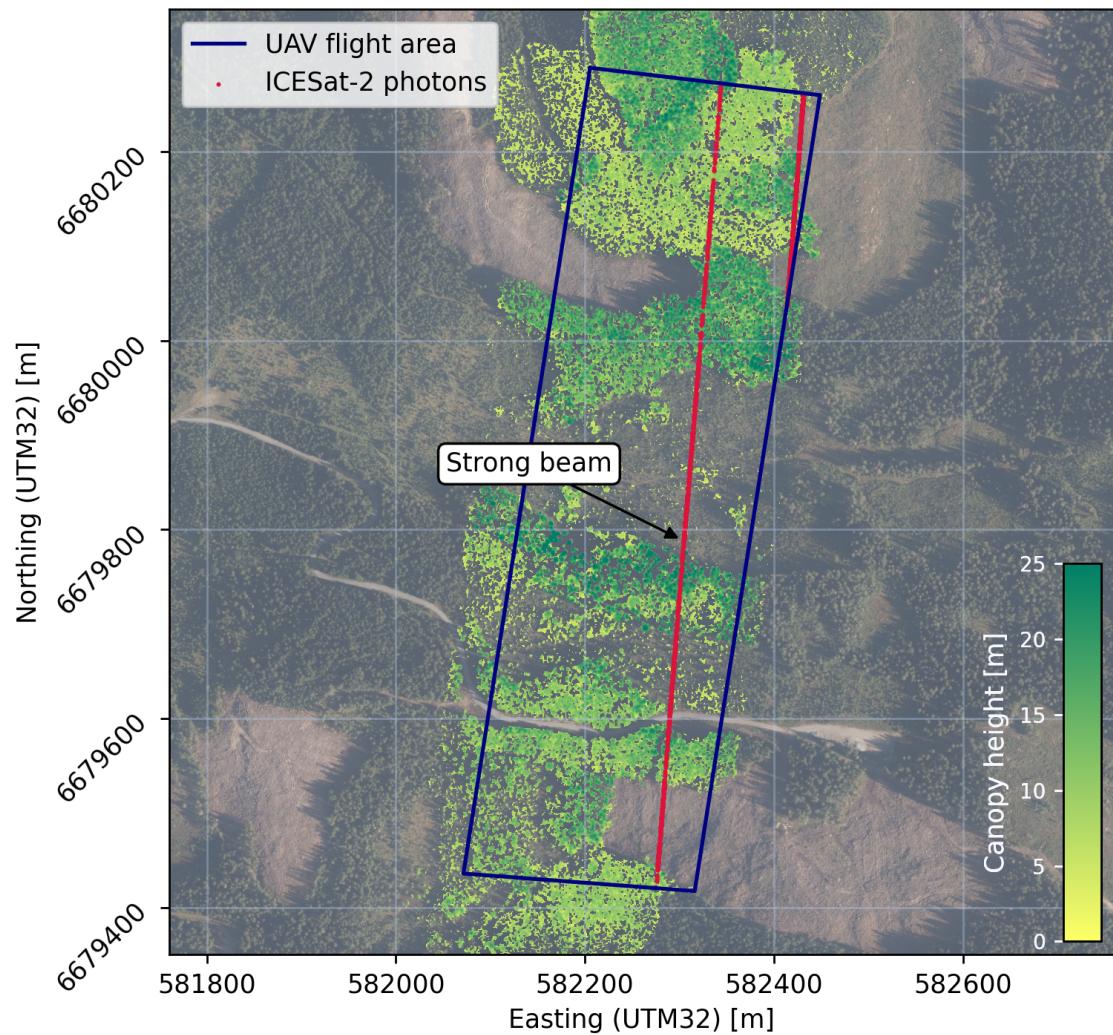


Figure 2.10: The study site in Jevnaker municipality.

2.3.4 Vikerfjell

Describe Vikerfjell

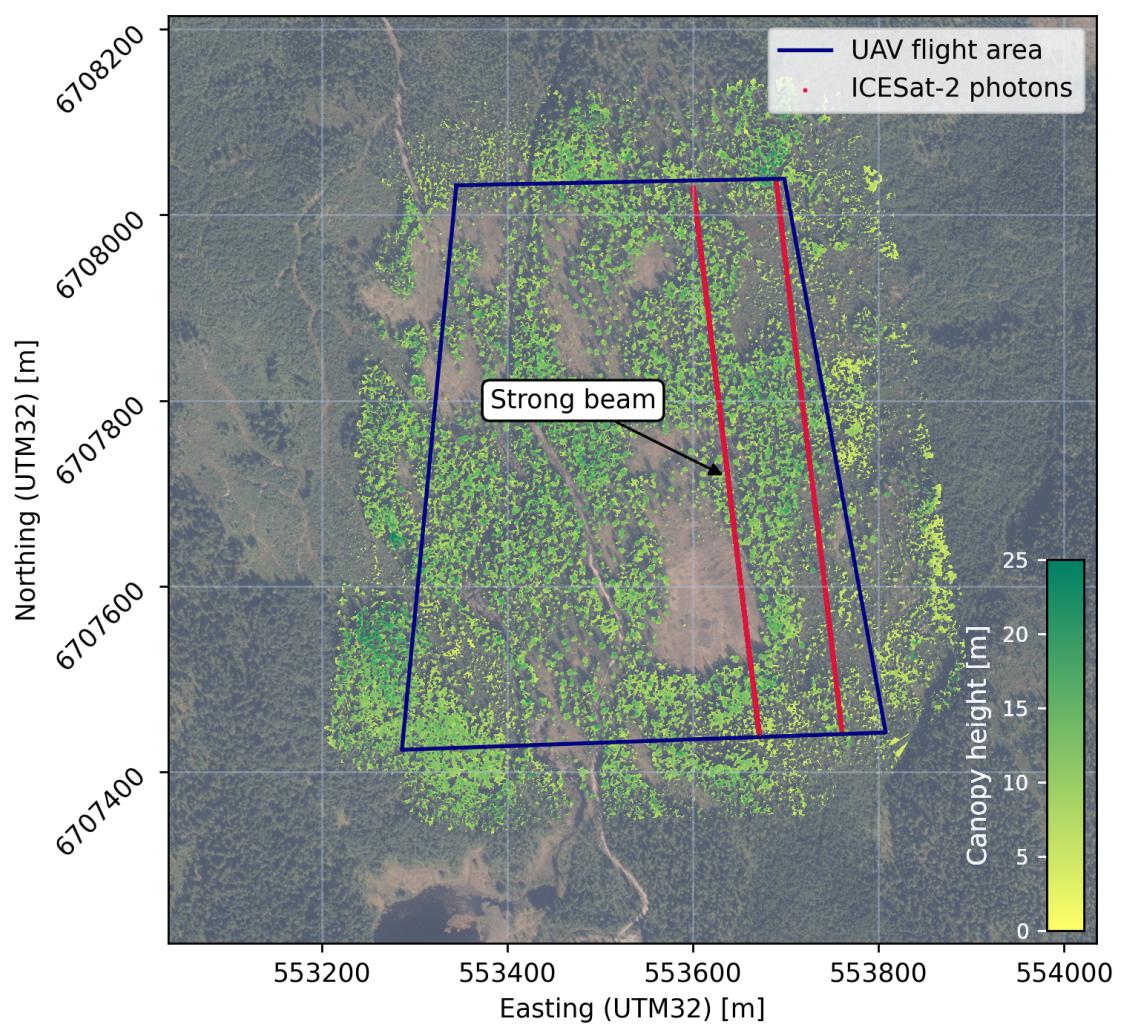


Figure 2.11: The eastern part of the site in Vikerfjell.

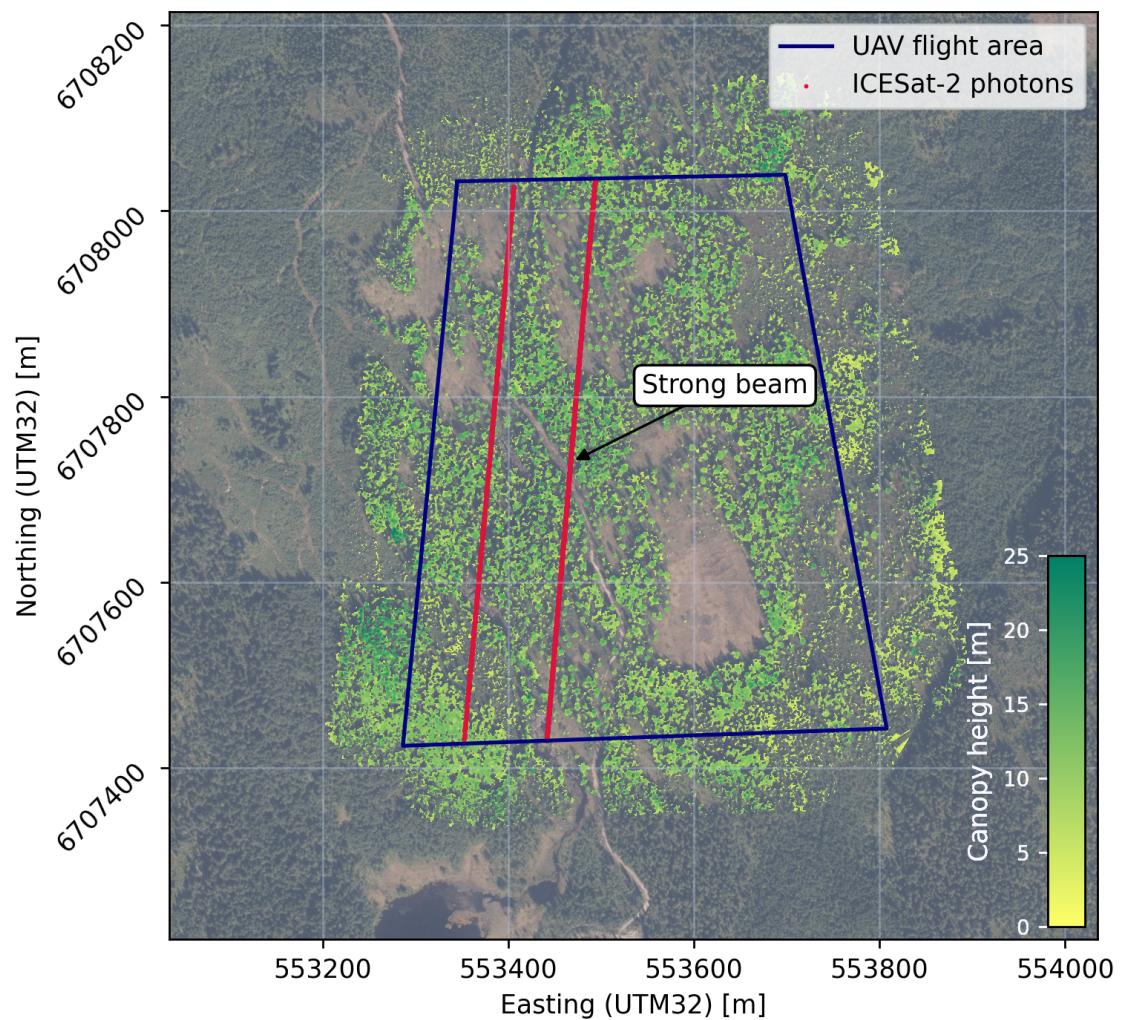


Figure 2.12: The western part of the site in Vikerfjell.

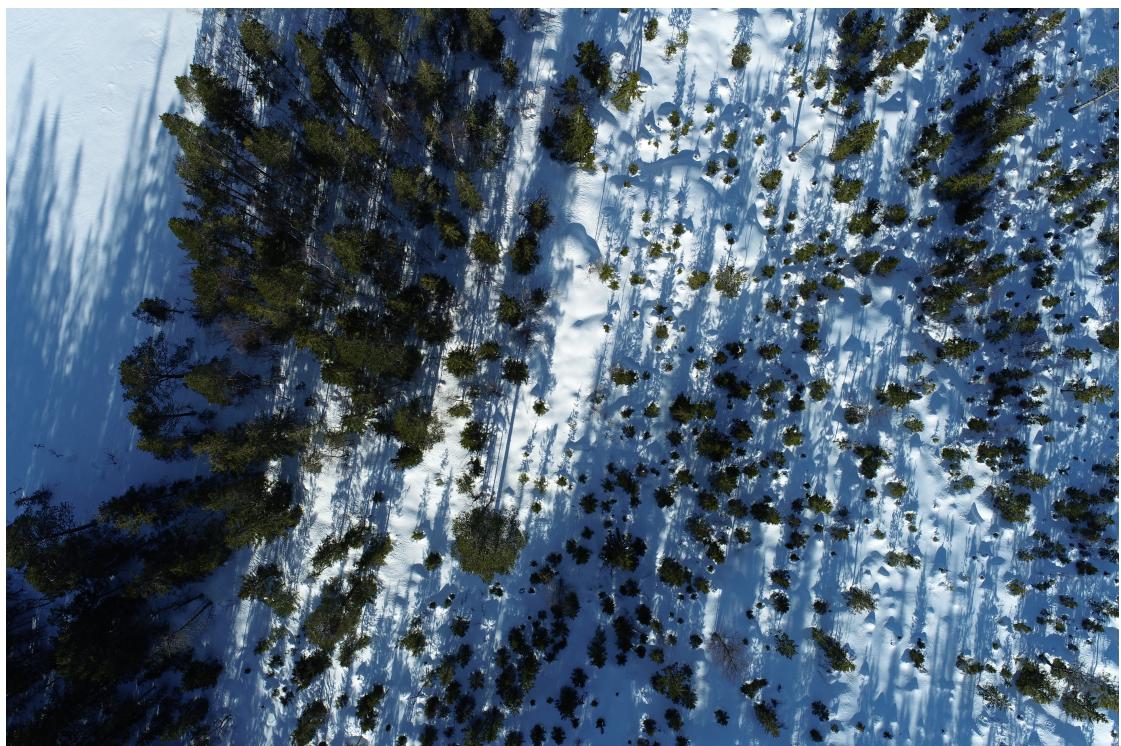


Figure 2.13: Sample image from the UAV flight in Vikerfjell, showing no snow cover on the canopy structure.

2.4 Data

The primary data used in this thesis was the data acquired by the ATLAS instrument and the UAV-borne LiDAR sensor. Although they both originated from LiDAR sensors, they contain different levels of information. The ATLAS instrument is a photon counting sensor, which essentially means that the data output is just individual returned photons and their two-way travel time. The Zenmuse L1 is a full-waveform LiDAR sensor, and each point therefore contains information about the illuminated surface as well.

In addition to the ICESat-2 and UAV data, official elevation models from the Norwegian Mapping Authority (NMA) was used as additional control data. This data included Digital Terrain Models (DTM) and a geoid model. The purpose of using this data was to confirm whether the UAV acquired "ground truth" provided an reasonably accurate representation of the ground, and serve as an indicator of whether the results could be scalable. The NMA DTMs were also necessary to enable use of the UAV terrain following function, which ensures an even distribution of LiDAR points throughout the field site.

Photons vs. points.
Main differences.
Why so many
DEMs? Max. min.
returns - Difference
and influence?

2.4.1 ICESat-2 products

The available ICESat-2 mission generates a wide range of products from the photon returns, although the main purpose of the mission is to measure polar sea ice and land ice. A total of 23 different product indicators are listed at the *Data Products* section of the ICESat-2 mission site. This thesis is based on the *Global Geolocated Photon Data* (ATL03) and *Land Water Vegetation Elevation* (ATL08) products. This applies both to the data downloaded directly from the NSIDC and from Sliderule. Both products are available in the *Hierarchical Data Format* (HDF).

DES

Show an example!
Photon plot?

Global Geolocated Photon Data — ATL03

As the name suggests, this product contains the spatio-temporal location of each photon returned to the ATLAS instrument. The photons are "classified by signal vs. background, as well as by surface type (land ice, sea ice, land, ocean), including all geophysical corrections (e.g. Earth tides, atmospheric delay, etc...)" (NASA 2024). The photon products are provided in granules, each spanning several minutes.

Land Water Vegetation Elevation — ATL08

The ATL08 product is more specialised, and contains ground height and canopy surface. If the data permits it, this product also includes canopy height, canopy cover percentage, surface slope and roughness, and apparent reflectance (NASA 2024).

2.4.2 UAV-borne LiDAR point clouds

2.4.3 Norwegian elevation data

Digital Terrain Model

Geoid model

Chapter 3

Data processing

3.1 Software

CLA

Cut the bulk of this section in favour of background/methods/results. Talk about tools _relevant_ to your research questions.

The software used in this thesis is described in this section. All processing, except the initial post-processing of the UAV point clouds, was handled with open source programming libraries in either Python or R languages. The only exception is the initial post-processing of the UAV point clouds, which had to be done in the DJI Terra software due to the sensor outputs proprietary format. They were then saved to the LAS file format, which is open source and easy to work with (ASPRS 2019).

The actual processing steps, and parameters used, is further described in section 3.2.

CLA

Why was it used? See it as accompanying L1, not a separate thing.

3.1.1 DJI Terra

DJI Terra is a licensed software provided by the same manufacturer as the drone and Zenmuse L1 LiDAR instrument. Its simple user interface makes it easy to learn and use, but as it is not open-source software, the user has less control over — and insight into — what happens with the data. The software can generate secondary products like DEMs, but using them would make the results of this thesis harder to reproduce.

CLA

Really?

3.1.2 Jupyter

Project Jupyter (Kluyver et al. 2016) is responsible for the development of a suite of applications, like *JupyterLab*. *JupyterLab* is a coding interface which includes support of several file formats and also has a file explorer. It integrates well with *GithHub*, which makes version control simple. *JupyterLab* is especially suitable for working with *Jupyter* notebook files. These are interactive documents which can use both *R* and *Python* languages in dedicated cells, in combination with pure markdown content for annotation and longer texts. Each cell can show its output directly, while variables and functions are remembered across the entire document. All of the code in this thesis, be it *Python*, *R* or *bash* scripts (GNU 2007), is written using *JupyterLab*.

3.1.3 R

R is a programming language which is particularly suitable for statistical computing and data visualisation, and is available on many platforms (R Core Team 2023). *R* was also available, along with necessary libraries, on the university’s high performance server, which significantly reduced processing time of the relatively heavy LiDAR point clouds.

lidR

The R library *lidR* is a powerful alternative for working with point clouds in the LAS/LAZ format, which makes it possible to use parallel processing to create canopy height models (CHM) and digital elevation models (DEM) (Roussel et al. 2020).

terra

The *terra* library (Hijmans 2024) is useful for working with spatial raster files, like DEMs. Saving the DEM and CHM products to file was done with *terra*.

3.1.4 Python

Most of the processing in this thesis is written in Python. The language is easy to learn, and uses a vast amount of packages, for almost any use case. With a simple, English-like syntax and cross-platform basis it provides a great starting point for working with data science and visualisation (Van Rossum and Drake 2009). Several libraries are used in this thesis:

pandas

pandas (The pandas development team 2020) is the most widely-used library for data analysis in Python. It handles a wide range of tabular data formats, as well as time series and other structured data sets. Data is structured in DataFrames, which in turn consist of Series. *pandas* contains a multitude of classes and functions to analyse big datasets in a fast and flexible way. Pandas also makes it easy to export tables in LaTeX format, which are used in the writing of this thesis.

DES

Shorten most of the libraries into one. Keep xDEM and SlideRule. Generic libraries can be mentioned in processing?

GeoPandas

GeoPandas is based on Pandas, but adds support for geospatial data and operations (Bossche et al. 2023). Examples include adding buffers and selecting data based on location, which is done in this thesis. GeoPandas uses GeoDataFrames as its basic data structure, which essentially is a Pandas DataFrame with an additional GeoSeries containing geometries (points, lines, polygons, etc.).

CLA

Keep Pandas/GeoPandas, xDEM, SlideRule. Rest in context of postprocessing

xDEM

Although in its early stages of development, xDEM is helpful when working with digital elevation models (DEM). In this thesis it is mainly used to extract elevation data from DEMs at the photon centre points, but it also provides functions for co-registration of DEMs with a method developed by Nuth and Kääb (2011).

SlideRule

Photons with the *Yet Another Photon Classifier* (YAPC) algorithm applied are downloaded from a web service API, through this *SlideRule* (Swinski et al. 2024) python client. YAPC (Sutterley 2022) is the basis of one of the filtering methods evaluated in this thesis.

NumPy

The *NumPy* library forms the basis of a lot of the computing operations in *Pandas*, but some operations in the processing in this thesis is done using NumPy functions directly. The library is one of the most widely used in the Python universe, built upon by a multitude of other libraries (Van Rossum and Drake 2009).

matplotlib

The visual presentations of the findings in this thesis is made with the library *matplotlib* (The Matplotlib Development Team 2023). *matplotlib* makes it easy to create simple data visualisation plots in Python, while also allowing users to modify their plots in more complex ways (Hunter 2007).

rasterstats

The rasterstats library (Perry 2024) makes it possible to extract raster information in several ways, including inside a polygon (zonal statistics). The ICESat-2 photons are defined with centre point geometry — while the photons have a surface footprint diameter of approximately 13 meters — and this library was helpful in getting DEM and CHM statistics inside the entire footprint.

3.2 Processing

3.2.1 UAV data

In order to create the ground truth elevation and canopy height models, the drone data had to be reworked from point clouds into suitable raster formats for later use with the ICESat-2 and Norwegian Mapping Authority (NMA) data.

Flight line alignment

The data collected from the UAV for this thesis was initially in a proprietary raw data format. To process it, the first step involved using the DJI Terra software for post-processing flight data. This software takes the point cloud data and other relevant mission files as inputs and converts them into a LAS point cloud file. In addition, the software can also perform ground point classification, which was applied during this process. Ideally, open-source software would have been used for this classification, but attempts with such software yielded difficult-to-manage results. The final output was a single LAS file, which was further converted to the LAZ format for a higher compression level of approximately 90 %. The conversion process utilised LASzip, which is an open-source tool developed specifically for this purpose (Isenburg 2017).

DTM and CHM generation

Point clouds are useful for discrete data, with information about each LiDAR return, but further processing relied on continuous data in the form of raster files. To generate the rasters, interpolation methods were applied to the point clouds. The point clouds for all locations were then used to generate Digital Terrain Models (DTM) and Canopy Height Models (CHM). Point clouds are very useful for storing information about the LiDAR laser returns, but they can be large and cumbersome to work with. As the comparable datasets from NMA — like the geoid model and DTMs — are in raster format, it is useful to create data models in the same format.

The *R* library *lidR* was very useful to achieve this. It enables parallel-processing on several of its built-in calculation algorithms, and could therefore work through each file in relatively short time. Still, even when using one of the university's high-performance servers the processing of each file took approximately one hour. The lidR script first takes an input LAS/LAZ file, and each point's x, y and z values — in addition to classification, number of returns, and return number — was selected for further processing. The script then returns the LAS file information. Example output is shown below, which describes a point cloud from the Drammen site:

```
class      : LAS (v1.2 format 3)
memory    : 10.7 Gb
extent     : 554808.1, 555267.5, 6628592, 6629570 (xmin, xmax, ymin, ymax)
coord. ref. : WGS 84 / UTM zone 32N
area       : 299760 m2
points     : 318.3 million points
density    : 1061.85 points/m2
density    : 765.59 pulses/m2
```

Finish describing each processing step. Describe more what I actually did, rather than what is possible. Less abstract. Show results along the way. Flow chart?

CLA

You haven't yet said how you will be using this data

CLA

Describe these so it's clear you understand LiDAR

CLA

What software?

CLA

Again, not described anywhere else

CLA

No. Point clouds are discrete, DTM/CHM are continuous

The next step involves applying a rasterisation function on the data, which generates the DTM. This function allows the use of different algorithms to create DTMs, but the default triangulated irregular network (TIN) method was applied here. As the name suggests, this method creates a triangulated network of the points classified as ground points, which makes it possible to get the height of any arbitrary point inside the network, using a bivariate function. The spatial resolution of the DTMs was set to 0.5 m.

CLA

Why?

CLA

Why?

CLA

You don't have a research question, and your problem statement doesn't mention canopy heights

CLA

You haven't introduced why you need these two datasets

The point clouds were also used to create the Canopy Height Models (CHM), which were integral for answering the research question in this thesis. The process is similar, but with a few important differences. Firstly, because some trees had been cut between the two data acquisitions (with and without snow), only the point clouds with snow cover could be used to create useful CHMs. Otherwise the models would not contain the same canopy structure as the ICESat-2 data. Second, three steps had to be completed before creating the output model:

1. The heights need to be normalised. In short, this process subtracts the ground height from each point, meaning that height values are now above-ground, instead of the elevation.
2. The next step was to rasterise the canopy. This is similar to when rasterising the terrain, but a higher resolution of 0.2 m was used, in addition to another algorithm, more suited for the large height variability of a canopy model compared to a ground model.
3. Lastly, before exporting the canopy model, a lower height threshold of 2 m was applied. Any structure above that height should be counted as forest canopy, which the thesis is focusing on. (No man-made structures were present in the study sites.)

The CHM was finally exported. All files, both DTM and CHM, were stored in the *GeoTIFF* format, which is a widely-used Open Geospatial Consortium standard format (OGC 2019).

The R processing script was applied to each point cloud file using a simple GNU bash script (see Appendix B), which also creates a log file with all console output and timings.

3.2.2 Data structuring and metrics extraction

The data is now ready for the main processing part. The data sources are in several different formats, making it beneficial to structure all relevant information in one dataset.

Pandas DataFrames are easy to work with,

3.2.3 Canopy height model

A function made to apply information from the canopy height model to each photon footprint. Implements the `zonal_stats` function from `rasterstats`, but also adds calculation of canopy density.

```

1 \todo[author=CLA]{Put this in an appendix?}
2 def chmStats(gdf, raster, diameter=12, stats=['mean'], density=False):
3     ...
4
5     Usage: chmStats(gdf, raster, diameter, stats, density)
6     Use a canopy height model (CHM) raster to calculate a set of stats
7     for the area within a photon footprint diameter. Defaults to
8     mean canopy height. With density=True, the canopy density is
9     calculated

```

Generate timestamp, merge info from ATL08, classify beams, interpolate data in gaps, apply pre-filtering (low signal conf, outside AOI, no heights)

CLA

How are you calculating canopy density/cover?

```

8     by also adding a 'count' and 'nan' column, which counts cells with
9     values
10    and NaNs respectively.
11
12    gdf:      geopandas.GeoDataFrame with the photons. Must contain a
13    geometry field
14    raster:    The CHM raster to use for calculations
15    diameter:  Photon footprint diameter [m]
16    stats:     Which statistics to calculate. Alternatives derive from
17    rasterstats.zonal_stats
18    density:   Whether to calculate canopy density. Will also add 'count'
19    ' and 'nan' to stats
20
21    Output:    The input gdf with new columns for stats
22    '',
23    from rasterstats import zonal_stats
24
25    if density:
26        stats += ['count', 'nan']
27    data = gdf.geometry.apply(
28        lambda pt: zonal_stats(pt.buffer(diameter/2), raster, stats=stats
29        , all_touched=True)[0]
30    )
31    chdf = pd.DataFrame(list(data), index=data.index.tolist())
32    if density:
33        chdf['density'] = chdf['count'] / (chdf['count'] + chdf['nan'])
34    for col in chdf.columns:
35        chdf.rename({col: f'chm_{col}'}, axis=1, inplace=True)
36    gdf = gdf.merge(chdf, how='outer', left_index=True, right_index=True)
37    return gdf

```

3.2.4 Photon filtering methods

The following subsection will describe the three photon filtering methods used to remove the ICESat-2 photons that likely do not represent the surface. The *threshold validation* and *Point grouping* methods are developed for this thesis, while the YAPC method uses the algorithm with the same name. The point grouping and YAPC methods are similar in principle, but YAPC is a far more complex algorithm.

DES

Why not ATL08

veg classifications?

At least discuss

Point grouping

The principle of the point grouping method is that if there are enough photons in the immediate vicinity of each other, they are more likely to represent a surface than not. The function lets the user define a lower threshold for the number of photon returns needed within a buffer, and the spatial extent of that buffer. This filtering is applied to the input photons with an inner function, before a rolling median is applied to the remaining photons. As with the threshold validation method, the values used in this function was found by trial and error until the results were optimal on one site, before applying the same values to all sites. The code for this function can be seen below:

```

1 def grpheights(gdf, hcol='h_ph', xybuff=1, zbuff=1, threshold=5, window
2 =20):
3     ''
4     Usage: grpheights(gdf, hcol, xybuff, zbuff, threshold, window)
5     For each photon in the data column hcol, use a 3D buffer of
6     horizontal size xybuff and vertical size zbuff, to scan for other
7     photons. If number of photons surpass threshold, the original photon

```

```

is assigned the mean height of the all photons in buffer. Otherwise
np.nan is assigned. Finally, a a rolling median function is assigned
to all photons.

5      gdf:           geopandas.GeoDataFrame containing photon heights
6      hcol:          The name of the photon height column
7      xybuff:        Horizontal buffer size
8      zbuff:         Vertical buffer size
9      threshold:    Photon count threshold
10     window:        Window to use for rolling median
11
12     Output:        geopandas.Series with new photon height values
13     '',
14
15
16     def group(row):
17         grp = gdf.loc[
18             ( gdf.within(row.geometry.buffer(xybuff)) ) &
19             ( abs(row.h_ph - gdf.h_ph) < zbuff )
20         ].h_ph
21         if len(grp) >= threshold:
22             return grp.mean()
23         else:
24             return np.nan
25
26     h_grp = gdf.apply(lambda row: group(row), axis=1)
27     h_grp = h_grp[h_grp.notnull()]      # Drop NA rows
28     h_grp = h_grp.rolling(window, closed='both', center=True).median()
29
30     return h_grp

```

DES

Sketch!

Rewrite to
make it easier
to understand.
Reconsider the
plot...

Threshold validation

The threshold validation method is based on a function developed while working on this thesis. In short, the function uses incremental percentiles of the height difference values, to define a threshold for reasonable snow depths. The input is a *Pandas Series* containing snow depth values. The function then finds the corresponding value of incrementally higher percentiles — in steps of 5 % between 30 % and 100 %. A visualisation of this can be seen in fig. 3.1. When these values increase by more than a pre-defined margin per increment, this value is then set as the threshold. The function returns a new *Pandas Series*, containing boolean value *True* where the input value was below the threshold, and *False* if they are above it. The original Series' indices are kept, making it easy to concatenate the returned Series to the original DataFrame.

The margin is set to a default value of 0.1 m in the function, which was found by simple trial and error on the ICESat-2 snow depths from one of the study sites. The margin was changed until the mean snow depth corresponded closely to that of the UAV snow depths. After testing it on the other sites, this margin turned out to work equally well there, regardless of the differences in both terrain slope and canopy structure/density. A downside to this method is that it requires the input values to be normalised, meaning that some form of "truth" dataset must be available. In this thesis, the snow-off DTM from the UAV flights were used. The principle behind this function is that when the snow depth values increase by more than a certain margin per increment of percentile value, these snow depths are actually, more likely than not, something other than the surface. This works best when the snow cover is expected to be relatively uniform throughout the area of interest, e.g. flat terrain or areas with equal temperature and precipitation. The function code is seen below:

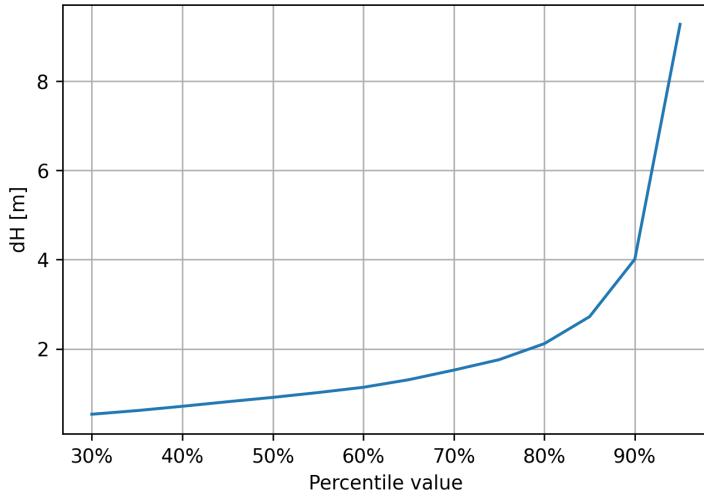


Figure 3.1: A visualisation of the change in the maximum acceptable snow depth value, as a function of the percentile value, as performed in *eval_ph*.

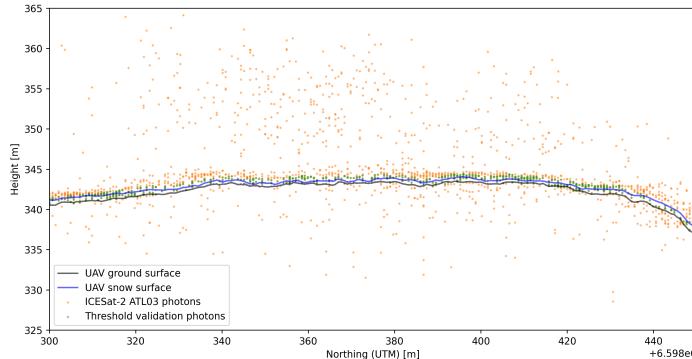


Figure 3.2: A sample set of photons from the Hof site, before and after the threshold validation is applied as described in *eval_ph*.

```

1 def eval_ph(values, margin=.1):
2     """
3         Usage: eval_ph(values, margin)
4         Takes a set of dH (snow depth) values as input. Calculates a suitable
5             percentile threshold to use for photon selection, and returns an
6             equal-shape series with boolean True where photons are within
7             threshold. (Preserves input indices.)
8
9     values:    pandas.Series with dH values
10    margin:    Max acceptable dH increase per 5 pct percentile increase
11
12    Output:   pandas.Series with boolean True/False for valid/invalid
13        snow depths
14    """
15
16    valid = values
17    x = np.arange(0.3, 1, .05)
18    y = abs(values).quantile(x)

```

```

15     y0 = y.iloc[0]
16     for threshold in y:
17         if threshold - y0 >= margin:
18             valid[(values > threshold) | (values < 0)] = False
19             return valid.astype('bool')
20         y0 = threshold
21     return values.astype('bool')

```

YAPC

This method uses the Yet Another Photon Classifier (YAPC) algorithm, which is developed by Jeff Lee at NASA's Goddard Space Flight Center (Sutterley 2023). It is a pure implementation of already-existing functionality, provided by the Python *SlideRule* library, described in section 3.1.4. The algorithm calculates a score for each photon, which is based on the number of nearest neighbours and a three-dimensional buffer. This is similar to what is done in the point grouping method above, but the calculation is more complex. It also allows the user to set a new threshold without recalculating anything, just by allowing a lower or higher score in further processing.

3.2.5 Snow depth calculations

3.2.6 Bias estimation

The accuracy of the point classification and ICESat-2 snow depth estimates are examined by comparing the ICESat-2 heights to the UAV heights.

3.2.7 Canopy vs. bias correlation

To assess whether the canopy density in the study sites affects the snow depth measurement errors (bias), a Pearson correlation coefficient r is calculated between these two variables. This coefficient can be used to indicate whether there is a linear correlation between two quantitative variables (Moore, McCabe and Craig 2012).

The Pearson correlation coefficient can be expressed as the mean of the products of the standard scores of the two variables x and y , with n observations:

$$r_{xy} = \frac{1}{(n-1)} \sum_{i=1}^n \left(\frac{x_i - \bar{x}}{s_x} \right) \left(\frac{y_i - \bar{y}}{s_y} \right)$$

The returned values are

Chapter 4

Results

DES

Remember to only
show results. Do
not discuss here

4.1 Performance of filtering methods

Describe how each filtering method performs, and relevant influences on results. Add transect plots showing the effect of each method

This section will contain plots and more detailed descriptions of how each filtering method performs.

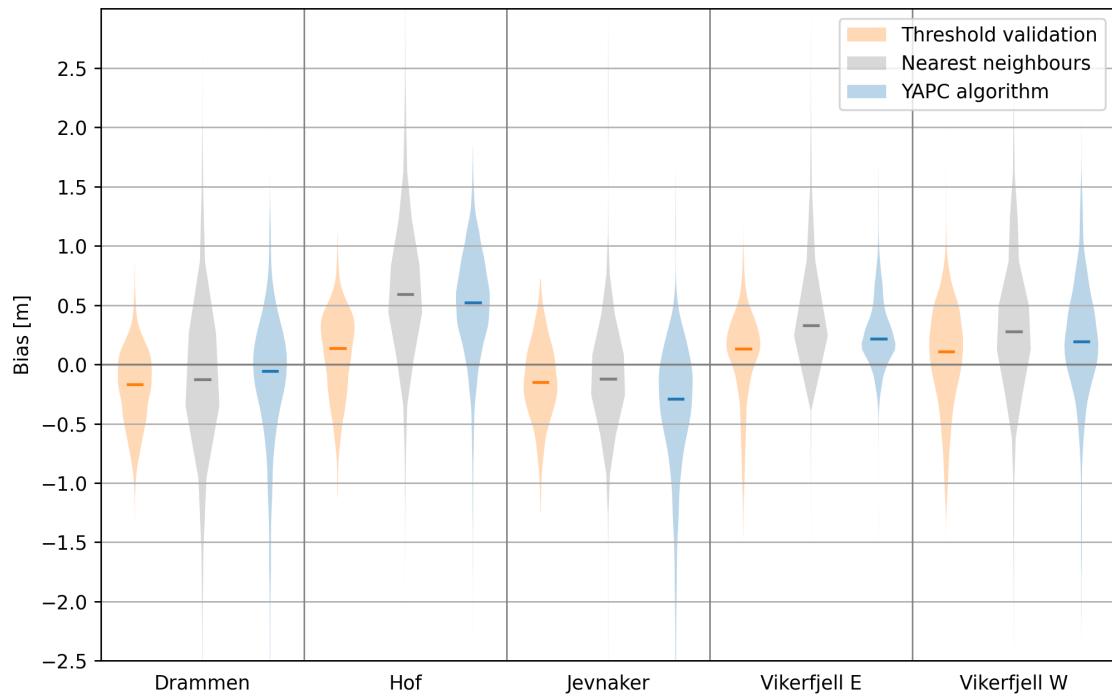


Figure 4.1: The distribution of photon bias after the filtering methods are applied, per area.
Median bias is indicated by lines.

4.1.1 Threshold validation

4.1.2 Point grouping

4.1.3 Yet Another Photon Classifier

4.1. Performance of filtering methods

Describe results for each site. How are they varying? Mention the influences on-site if relevant. Add transects.

Re-check if beam is relevant

4.2 Per field site

4.2.1 Drammen

Table 4.1: Statistics per method, Drammen (Both beams)

Method	Bias	MAE	RMSE	Rel. Bias	Rel. RMSE	Corr. veg.
Unfiltered	-1.773 m	3.914 m	6.379 m	-300%	1,081%	-0.13
Threshold	0.198 m	0.292 m	0.373 m	33%	63%	0.05
Grouped	0.123 m	0.227 m	0.677 m	21%	115%	0.01
YAPC	0.152 m	0.411 m	0.639 m	26%	108%	0.14

4.2.2 Hof

Table 4.2: Statistics per method, Hof (Both beams)

Method	Bias	MAE	RMSE	Rel. Bias	Rel. RMSE	Corr. veg.
Unfiltered	-1.226 m	2.343 m	4.087 m	-246%	820%	-0.21
Threshold	-0.088 m	0.306 m	0.365 m	-18%	73%	-0.11
Grouped	-0.622 m	0.465 m	0.802 m	-125%	161%	-0.05
YAPC	-0.483 m	0.579 m	0.682 m	-100%	141%	0.12

4.2.3 Jevnaker

Table 4.3: Statistics per method, Jevnaker (Both beams)

Method	Bias	MAE	RMSE	Rel. Bias	Rel. RMSE	Corr. veg.
Unfiltered	-0.887 m	1.786 m	3.878 m	-200%	872%	-0.31
Threshold	0.146 m	0.282 m	0.355 m	33%	80%	-0.41
Grouped	0.118 m	0.134 m	0.652 m	27%	147%	-0.06
YAPC	0.385 m	0.495 m	0.705 m	23%	43%	0.01

4.2.4 Vikerfjell East

Table 4.4: Statistics per method, Vikerfjell E (Both beams)

Method	Bias	MAE	RMSE	Rel. Bias	Rel. RMSE	Corr. veg.
Unfiltered	-0.638 m	1.189 m	2.311 m	-65%	235%	-0.16
Threshold	-0.063 m	0.302 m	0.386 m	-6%	39%	0.02
Grouped	-0.429 m	0.344 m	0.820 m	-44%	84%	-0.16
YAPC	-0.255 m	0.299 m	0.390 m	-26%	39%	-0.25

4.2.5 Vikerfjell West

Table 4.5: Statistics per method, Vikerfjell W (Both beams)

Method	Bias	MAE	RMSE	Rel. Bias	Rel. RMSE	Corr. veg.
Unfiltered	-0.316 m	1.244 m	2.874 m	-33%	299%	-0.10
Threshold	-0.063 m	0.347 m	0.434 m	-7%	45%	-0.03
Grouped	-0.297 m	0.353 m	0.659 m	-31%	69%	-0.10
YAPC	-0.186 m	0.409 m	0.555 m	-19%	58%	-0.03

Chapter 5

Discussion

5.1 Application

Do the results provide answers to the research question? Would other areas yield comparable results? Discuss every lesson learned along the way!

DES

Performance of algorithms? -
-> Difficulty to get validation data (field work planning (RE: Jevnaker))? -
-> ATL08 veg classification?
Possibility to apply this at larger scale? Correlation evaluation also.
Recommended next steps?

CLA

Mention max returns from M300 (3). Discuss canopy cover threshold (2m)

5.2 Workflow Review

Review the processing workflow and software used. What works, and what could have been better/more efficient?

5.3 Software Review

Chapter 6

Conclusion

6.1 Main findings

Is the research
question answered?

6.2 Applications and future work

How can these findings be used, and what needs to be improved or further explored/tested

Chapter 6. Conclusion

Bibliography

- ASPRS (9th July 2019). *LAS Specification 1.4 - R15*. The American Society for Photogrammetry & Remote Sensing. URL: https://www.asprs.org/wp-content/uploads/2019/07/LAS_1_4_r15.pdf (visited on 16/03/2024).
- Bossche, Joris Van den et al. (June 2023). *geopandas/geopandas: v0.13.2*. Version v0.13.2. DOI: 10.5281/zenodo.8009629. URL: <https://doi.org/10.5281/zenodo.8009629>.
- GNU, P (2007). *Free Software Foundation. Bash (3.2. 48)[Unix shell program]*.
- Hijmans, Robert J. (2024). *terra: Spatial Data Analysis*. URL: <https://CRAN.R-project.org/package=terra>.
- Hunter, J. D. (2007). ‘Matplotlib: A 2D graphics environment’. In: *Computing in Science & Engineering* 9.3. Publisher: IEEE COMPUTER SOC, pp. 90–95. DOI: 10.1109/MCSE.2007.55.
- Isenburg, Martin (30th Oct. 2017). *LAStools - efficient LiDAR processing software*. Version 171030, unlicensed. URL: <https://rapidlasso.com/LAStools>.
- Kluyver, Thomas et al. (2016). ‘Jupyter Notebooks – a publishing format for reproducible computational workflows’. In: *Positioning and Power in Academic Publishing: Players, Agents and Agendas*. IOS Press, pp. 87–90. DOI: 10.3233/978-1-61499-649-1-87. URL: <https://ebooks.iospress.nl/doi/10.3233/978-1-61499-649-1-87> (visited on 09/04/2024).
- Magruder, Lori A. et al. (Feb. 2024). ‘Monitoring Earth’s climate variables with satellite laser altimetry’. In: *Nature Reviews Earth & Environment* 5.2, pp. 120–136. ISSN: 2662-138X. DOI: 10.1038/s43017-023-00508-8. URL: <https://www.nature.com/articles/s43017-023-00508-8> (visited on 23/04/2024).
- Moore, David S., George P. McCabe and Bruce A. Craig (2012). *Introduction to the practice of statistics*. 7. ed., internat. ed. New York: Freeman [u.a.] 101-105. ISBN: 978-1-4292-8664-0.
- NASA (2024). *Data Products / ICESat-2*. Data Products | ICESat-2. URL: <https://icesat-2.gsfc.nasa.gov/science/data-products> (visited on 11/04/2024).
- NASA’s Earth Observing System (2024). URL: <https://eospso.gsfc.nasa.gov/> (visited on 09/04/2024).
- Neumann, T. A. et al. (2023). *ATLAS/ICESat-2 L2A Global Geolocated Photon Data, Version 6*. DOI: 10.5067/ATLAS/ATL03.006. URL: <http://nsidc.org/data/atl03/versions/6> (visited on 09/04/2024).
- Neumann, Thomas A. et al. (1st Nov. 2019). ‘The Ice, Cloud, and Land Elevation Satellite – 2 mission: A global geolocated photon product derived from the Advanced Topographic Laser Altimeter System’. In: *Remote Sensing of Environment* 233, p. 111325. ISSN: 0034-4257. DOI: 10.1016/j.rse.2019.111325. URL: <https://www.sciencedirect.com/science/article/pii/S003442571930344X> (visited on 15/06/2022).
- Nuth, C. and A. Kääb (29th Mar. 2011). ‘Co-registration and bias corrections of satellite elevation data sets for quantifying glacier thickness change’. In: *The Cryosphere* 5.1.

Bibliography

- Publisher: Copernicus GmbH, pp. 271–290. ISSN: 1994-0416. DOI: 10.5194/tc-5-271-2011. URL: <https://tc.copernicus.org/articles/5/271/2011/> (visited on 31/01/2024).
- OGC (14th Sept. 2019). *GeoTIFF Standard*. URL: <https://docs.ogc.org/is/19-008r4/19-008r4.html> (visited on 14/04/2024).
- Perry, Matthew (26th Mar. 2024). *perrygeo/python-rasterstats*. original-date: 2013-09-18T06:44:29Z. URL: <https://github.com/perrygeo/python-rasterstats> (visited on 29/03/2024).
- R Core Team (2023). *R: A Language and Environment for Statistical Computing*. Vienna, Austria: R Foundation for Statistical Computing. URL: <https://www.R-project.org/>.
- Roussel, Jean-Romain et al. (15th Dec. 2020). ‘lidR: An R package for analysis of Airborne Laser Scanning (ALS) data’. In: *Remote Sensing of Environment* 251, p. 112061. ISSN: 0034-4257. DOI: 10.1016/j.rse.2020.112061. URL: <https://www.sciencedirect.com/science/article/pii/S0034425720304314> (visited on 23/01/2024).
- Sutterley, Tyler (June 2022). *tsutterley/yapc: v0.0.0.8*. Version 0.0.0.8. DOI: 10.5281/zenodo.6717591. URL: <https://doi.org/10.5281/zenodo.6717591>.
- (2023). *YAPC Overview — pyYAPC 0.0.0.8 documentation*. Rev. 87023d6e. URL: https://yapc.readthedocs.io/en/latest/getting_started/Overview.html#yapc-goals (visited on 17/04/2024).
- Swinski, J. P. et al. (Mar. 2024). *ICESat2-SlideRule/sliderule: v4.3.1*. Version v4.3.1. DOI: 10.5281/zenodo.10797886. URL: <https://doi.org/10.5281/zenodo.10797886>.
- The Matplotlib Development Team (Nov. 2023). *Matplotlib: Visualization with Python*. Version v3.8.2. DOI: 10.5281/zenodo.10150955. URL: <https://doi.org/10.5281/zenodo.10150955>.
- The pandas development team (Feb. 2020). *pandas-dev/pandas: Pandas*. Version latest. DOI: 10.5281/zenodo.3509134. URL: <https://doi.org/10.5281/zenodo.3509134>.
- Van Rossum, Guido and Fred L. Drake (2009). *Python 3 Reference Manual*. Scotts Valley, CA: CreateSpace. ISBN: 1-4414-1269-7.