B087573

Object Orientated Software Engineering: Spatial Algorithms Assignment

Mapping Pine Island Glacier

1. Introduction

1.1. Pine Island Glacier

The area of interest in this study is the Pine Island Glacier located on the West Antarctic Ice Sheet in Antarctica. The glacier has been recognised as the fastest melting glacier in Antarctica.

1.2. Land, Vegetation and Ice Sensor

LVIS is a full-waveform imaging radar, developed by NASA. The instrument measures the reflected laser intensity as a function of range *(Baltsavias, 1999)*, thus it's entire history of the outgoing and return pulses in digitized.

2. Methods and Results

2.1. Task 1: DEM from a single flight line

This task aims to process a single LVIS flight line to produce a Digital Elevation Model (DEM) in a GeoTIFF format of any chosen resolution. In order to derive a DEM, *flight_2_dem.py* first denoises the waveforms and then identifies the center of mass of each waveform and saves that as a ground elevation value (see *figure 1*).

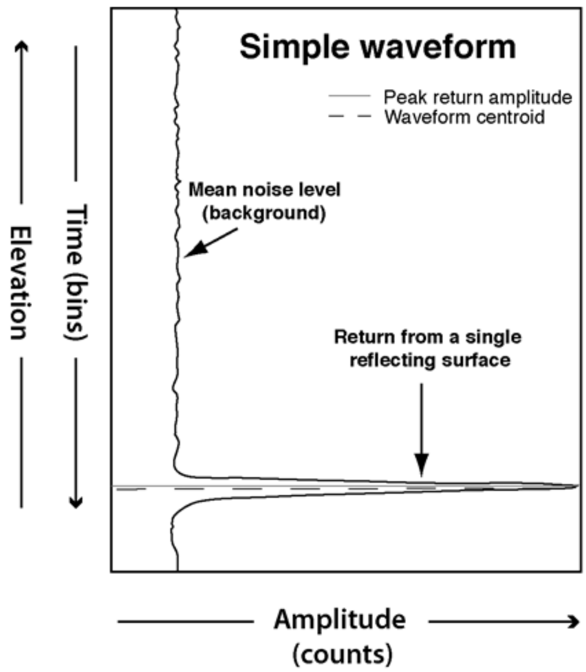

Figure 1. The ground elevation can be easily extracted by computing the center of mass of each waveform. Source: NASA.

The script then uses the identified elevations to convert the point data to a raster. The value of each pixel is taken as a mean of all points that fall within the pixel. The resulting file is then saved as a GeoTIFF.

The figure 2 illustrates the output of this script. It shows that rasterised points are not evenly spaced and data is somewhat scarce.

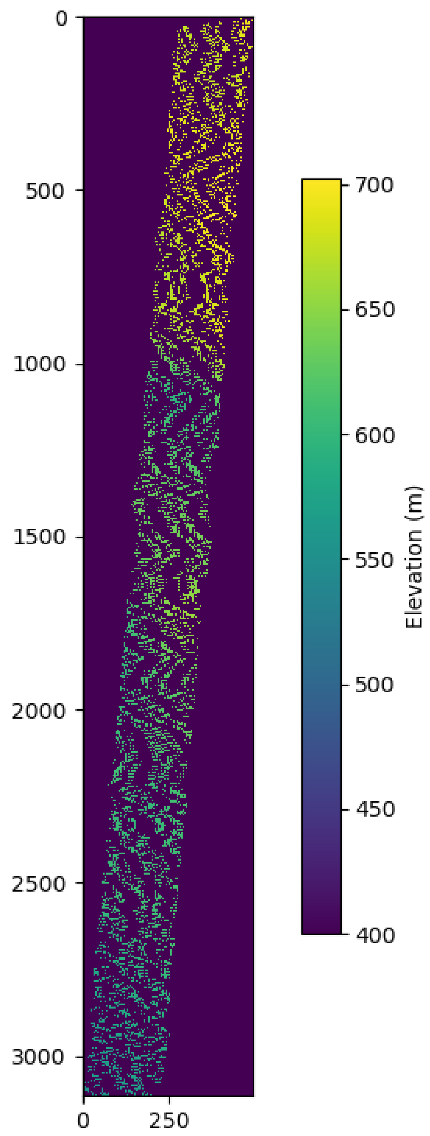

Figure 2. 10 m DEM derived from *ILVIS1B_AQ2015_1017_R1605_056419.h5 flight line from 2015.*

2.2. Task 2: gap-filled DEM from all the 2015 data

The *full_dem.py* script processes all the flight lines that fall within the area of interest for the specified year into a single DEM. In order to minimize RAM usage, the script saves the intermediate GeoTIFFs to disc and executes a garbage collection process after each processed flight line.

Once all the flight lines are processed, it merges all the DEMs into one by building a virtual raster using GDAL and applying a pixel function which averages the overlapping pixels. No data values are then filled by interpolating (max distance of 50 pixels was specified). A Gaussian filter with a 5x5 kernel is then applied to smooth out the artefacts in the gap-filled DEM. The resulting DEM is then written to a *YEAR.tif* file.

Since LVIS data processing is a computationally intensive procedure, a resolution of 100 m was chosen as it was found to minimize the trade-off between the processing time and the preservation of detail in the imagery.

Multiple tests were performed before running the code over the full dataset. To track RAM usage, *psutil* was used to place memory management checkpoints after each file read-in. The runs were also timed using *timeit* module to get the sense of processing timescales and see if any pieces of code could be optimised. It took around 3 hours to process all of the 2015 data and about 25 min for 2009 since there were significantly less data.

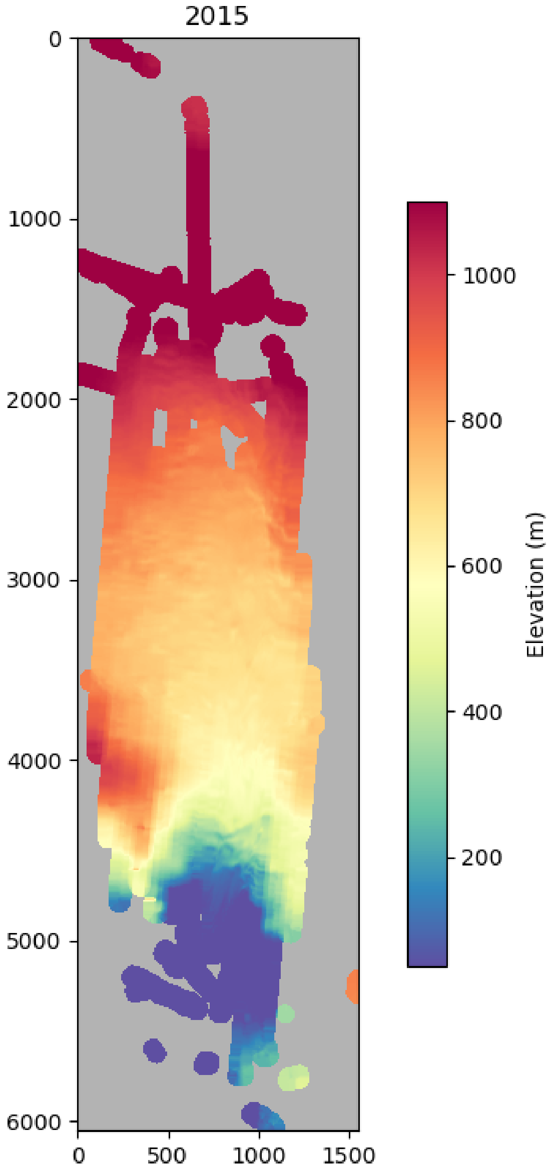

Figure 3. Gap-filled and smoothed out 100 m Pine Island Glacier DEM derived from 2015 LVIS data.

2.3. Task 3: change in elevation between 2009 and 2015

The code from task 2 has been used to process all of the 2009 data and produce a gap-filled DEM. A new *DEM_merge* class has been written to read in both DEMs into RAM. The *elevation_change.py* then preprocesses both input datasets to a common projection, extent and resolution. The resulting rasters are then differenced to create an elevation change map, which is then saved as a GeoTIFF. The resulting elevation difference map between 2009 and 2015 *(figure 4)* shows that the majority of the glacier has negative elevation change, with most values falling in the 10-30 m loss range. It should be noted that some edge effects are present in the derived DEM due to interpolation from sparse data.

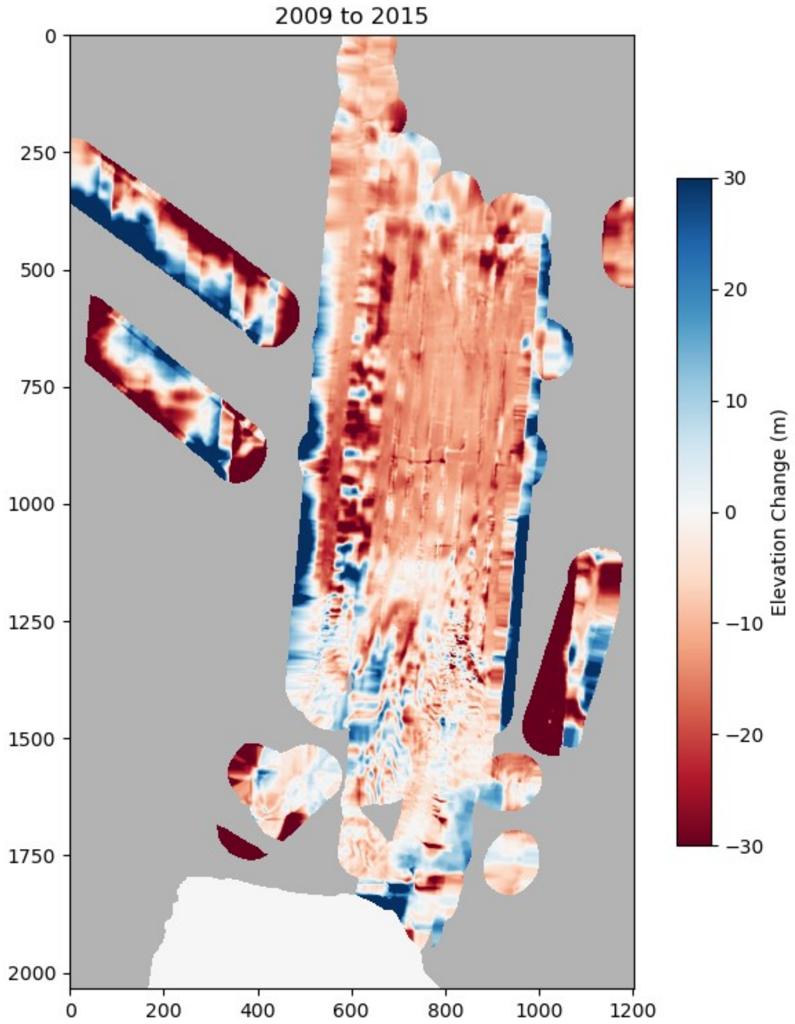

Figure 4. Elevation change map for Pine Island Glacier from 2009 to 2015 (100 m resolution).

The *DEM_merge* class also has *volume_change()* method which computes multiple glacier volume and mass change metrics, such as mean elevation change rate, total area, volume and mass change.

The change metrics computed for Pine Island Glacier revealed that in the area of interest the glacier lost on average 1.15 m of ice per year, which is equivalent to 68.23 km^3 loss in volume and approximately 58.00 Gt loss in mass from 2009 to 2015.

2.4. Task 4: calculate contour lines

The challenge in this task is to add method for computing contour lines at a user specified interval. After doing some research about existing algorithms for computing isocontours, I came across the **Marching Squares algorithm**, which is commonly used in computer graphics.

In the *contours.py* script, I have first investigated how the Marching Squares algorithm is implemented in the skimage package *(see contour()).* Then I have attempted to implement the algorithm myself in the *contour_trial()*. The method does compute the vertices of the contours but it proved to be incredibly inefficient when compared to the skimage's one. Moreover, my implementation computed all of the vertices, thus contours should be simplified using the Douglas-Peucker algorithm. Given more time for the assignment, algorithm optimization and contour refinement would be a priority. The outputs of *contours.py* are illustrated below.

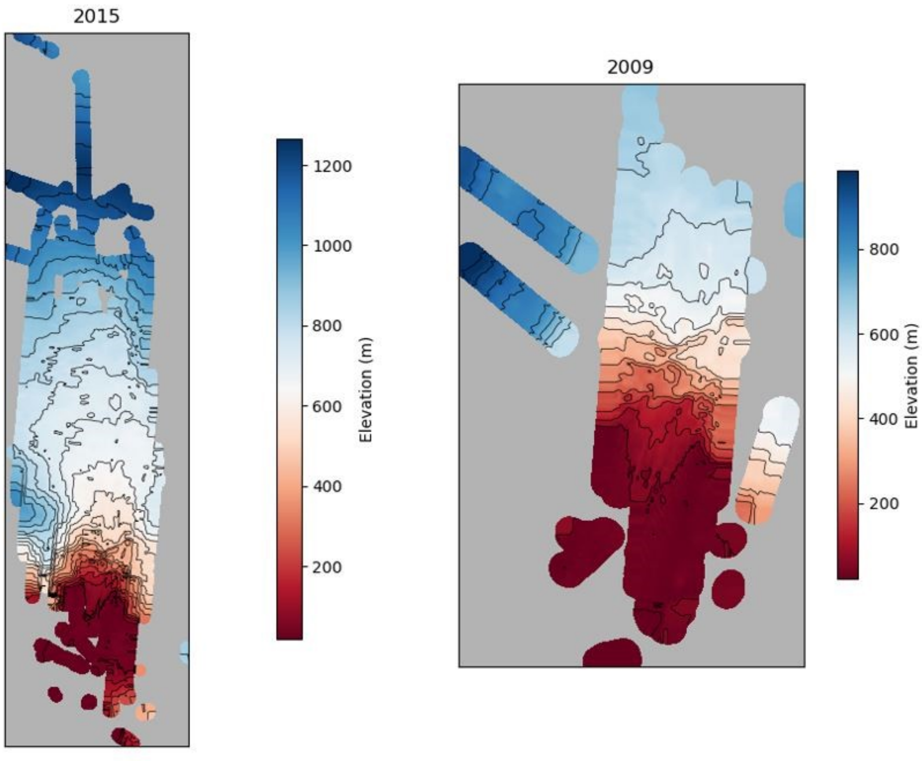

Figure 5. Contours overlaid every 50 m on 2009 and 2015 Pine Island Glacier DEMs.

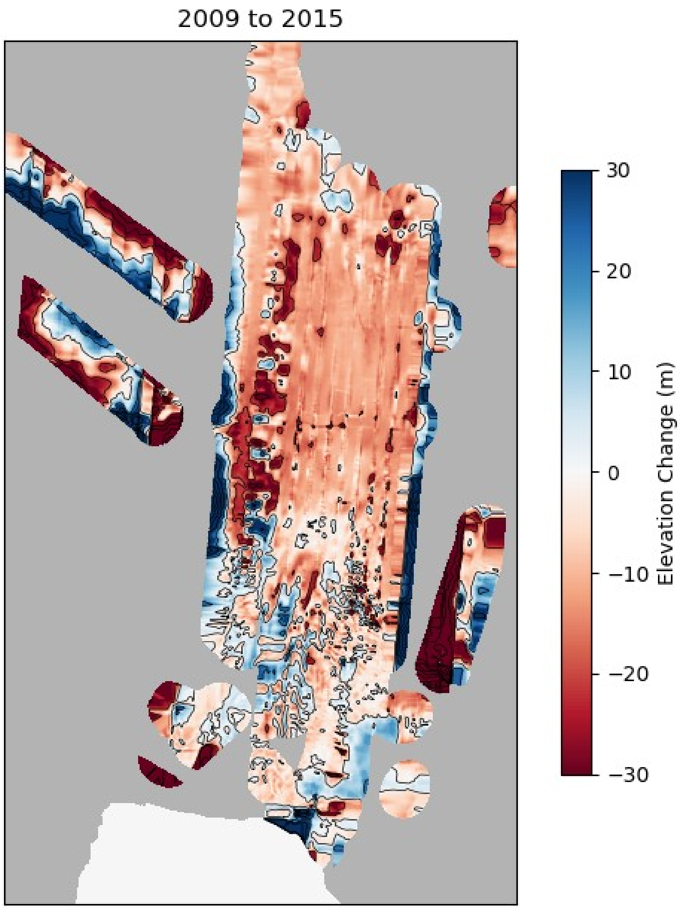

Figure 6. Contour overlaid every 20 m on the Pine Island Glacier elevation change map.

4. Discussion

A good starting toolset was developed for processing raw LVIS data in Python. However, some of the tools are not yet fully complete. Given more time, I would prioritize optimizing the code and reducing the number of intermediate files produced. I would also focus on making the code fully flexible, meaning the user would have to think less about directory structure and file names. More work needs to be done on the contours class as the performance of Marching Squares algorithm is not great. It might be a good idea just to stick with skimage's implementation of the algorithm as it is already optimised and uses Cython for the most important parts.

The DEM products, derived from LVIS waveforms, have been shows to be useful in visualising and quantifying ice loss in Antarctica. There is a lot of potential in using LVIS data for scientific glaciology research, specifically for identifying areas of high/low ice loss or possibly even glacier balance modelling. However, such analyses are limited by the availability of the data. In this study, we have only examined the northern part of the Pine Island Glacier, where there is sufficient amount of data. Even in these areas, where the distance between flight lines is rather big, edge effects and interpolation errors are present in the imagery.

To sum up, this has been as a rather challenging project to implement as not only the code must have been object-oriented, but also things like RAM usage, processing time and algorithm performance must have also been taken into account. I think the current code is satisfactory but it definitely could be improved.

6. References

Baltsavias, E. (1999). "Airborne laser scanning: basic relations and formulas." Isprs Journal Of Photogrammetry And Remote Sensing **54**(2-3): 199-214.