Digital Surface Model Production

Eduard Heijkoop

University of Colorado Boulder

July 17, 2023

Contents

1 General

1.1 Introduction

This document will describe the process with which to create mosaics of high resolution digital surface models (DSMs), with scenes produced by SETSM [Noh and Howat, 2015] and subsequent strips produced by the Polar Geospatial Center's batch_scenes2strips.py code. These mosaics can be used for InSAR processing or inundation projections.

In general, each section in this document will describe a set of flags that can be activated when one runs the code, and a configuration file that will define constants and paths to specific files needed to run the software. It is recommended to edit the relevant configuration files once, and store it separately, so when a `git pull` is run, it is easy to quickly replace the configuration file. This file will change much less frequently than new updates to the code, but upon a new pull it will be replaced. It is most likely that only the paths and usernames need to be edited; constant values should work, but may be edited by users to suit their needs. Note that any command or variable (such as `git pull`) shown in this document is case-sensitive.

Most scripts will also have the capability of reading input files, allowing for many sequential jobs to be run without any user input. This was done primarily to scale up processing for the NASA Sea Level Team (N-SLCT), which is also the source of funding for this work.

Please note that this software has only been tested in an Ubuntu Linux environment. No guarantee is given that this will work on Mac OS or Windows (or even on different Linux distributions).

1.2 Repositories

The code described here is available on GitHub, specifically in the DEM, ICESat-2 and Utilities repositories. For first time downloading, use the `git clone` command; for keeping things up-to-date afterwards, use the `git pull` command.

1.3 Python Packages & Environments

The software requires a number of python packages to work. One way to ensure compatibility is to create a new anaconda environment (which requires anaconda/miniconda to be installed), and install the packages in this new environment. For example:

1. `conda create -n mosaic python=3.11`
2. `conda activate mosaic`
3. `conda install -c conda-forge numpy pandas geopandas gdal scipy netCDF4 matplotlib h5py requests`

1.4 Different Machines

The code has been written to run on a number of different machines at CU Boulder, each with their own pathing to the same directory, with `t` being the default. Only when `b` or `local` are selected, will the paths in the configuration file be edited after loading them up. After editing the paths in the configuration file to suit one's needs, it is recommended that the machine flag is omitted, so the paths will not be edited.

1.5 Other Software & Requirements

- A NASA EarthData user account. This can be created for at https://urs.earthdata.nasa.gov/. It is recommended to never store your password in plaintext, and anytime the software asks for your password, it will not be stored anywhere.
- NASA Ames Stereo Pipeline (ASP), specifically the `dem_mosaic` command. ASP can be installed by downloading the latest (stable) release from their GitHub page, unzipping the `.tar` file and adding the path to your `bashrc`.
- AWS Command Line Interface, used to download the Copernicus 30 m DEM from its AWS S3 bucket. No costs are associated with the user for downloading this data off of AWS, but an account may be required. This can be avoided if either SRTM or ASTER is used instead, which are hosted by NASA/USGS and require `wget` to access instead.

- Optional: a Google Earth Engine (GEE) account. The spatial filtering of ICESat-2 uses Sentinel-2 obtained through and processed on GEE. Note: non-academic users may incur (significant) costs when using this cloud-based computing platform! Please thoroughly assess the costs before scaling up this tool.
- General Unix utilities such as `wget` and `unzip`.

1.6 Other Files

A number of files are required for the code to work:

- The OpenStreetMap coastline [OpenStreetMap contributors, 2017], available here. Make sure to download the file that's in the WGS84 projection and is not split.
- Geoids: All three *a priori* DEMs (SRTM, ASTER & Copernicus) come in orthometric heights and our work is in the ellipsoidal space. Thus, to convert these coarser (30 m) products to ellipsoidal heights, a geoid is needed, specifically EGM96 for SRTM & ASTER, and EGM2008 for Copernicus. EGM96 can be found here and EGM2008 can be found here.
- IPCC AR6 regional projections of sea level change. These were hosted on PO.DAAC in NetCDF format, but have since moved to the PO.DAAC cloud and are temporarily unavailable.
- Global Surface Water [Pekel et al., 2016] (GSW) data, specifically the extent tiles. $10° \times 10°$ tiles can be obtained here.

1.7 Procedure

The whole procedure can be summarized by the following steps, each of which is briefly explained below and will have its own section in this document.

1. Download ICESat-2
2. Optional steps:
   - Detect DSM strips that are too cloudy or too much over water
   - Filter ICESat-2 for vegetation and inland water
   - Correct DSM strips with (filtered) ICESat-2
3. Turn DSM strips into a single coherent mosaic
4. Co-register DSM mosaic to ICESat-2
5. Compute inundation projections

1.7.1 Download ICESat-2

The individual DSM strips and the resulting mosaic lack an absolute vertical accuracy. As such, we use the ICESat-2 ATL03 geolocated photon product as an absolute vertical truth. This code will download the product from the NSIDC and turn it into a format that can be used to (optionally) first correct the individual strips and then co-register the final mosaic.

1.7.2 Optional Steps

The following steps are optional, but were all developed to increase the vertical precision and accuracy of the final product.

Detect Cloudy DSM Strips

Some strips are produced with imagery acquired during cloudy conditions. These clouds manifest themselves as anomalously high (off by > 100 m) regions in the DSM, which is undesirable when using the product for certain applications, like inundation projections. For some regions, particularly in the tropics, it is inevitable that cloudy images are used, but it is still important to quantify this.

This code will download an *a priori* DEM (either SRTM, ASTER or Copernicus) and difference it with the strip(s). As SRTM and Copernicus are derived from radar data (the Shuttle radar and TerraSAR-X, respectively), they are not susceptible to cloudy conditions and can be used as a baseline. ASTER is generally considered an improvement over SRTM, but some issues remain with it, as it is derived from (near) visible light bands.

Filter ICESat-2

The ICESat-2 ATL03 product's classification only includes confidence levels for land, ocean, sea ice, land ice and inland water. For coastal application only the first two are of interest, and precisely in this region the land classification is buffered offshore by 25 km and the ocean classification is buffered inland by 10 km [Neumann et al., 2021]. Additionally, the inland water classification is only at a 0.01° resolution, corresponding to a distance of approximately 1 km. Finally, there is no vegetation classification in the ATL03 product.

As such, we wish to detect inland water and vegetated areas with a much higher accuracy than is available with any built-in data, in order to remove photons over water or vegetated areas that would bias the co-registration process. To do this, we co-locate Sentinel-2 multispectral imagery in space and time with ICESat-2 acquisitions and derive vegetation and water masks from these images with which we filter the ICESat-2 ATL03 geolocated photons. Co-registering the DSM (either a strip or a mosaic) to this filtered ICESat-2 dataset improves vertical precision w.r.t. the original ICESat-2 dataset.

Correct DSM Strips

Individual strips, particularly those derived from WorldView-1 imagery may be affected by jitter of the spacecraft sensor. This jitter manifests itself as an along-track (which is approximately North-South, but angled by $|90^\circ - i|$, where i is the spacecraft inclination) oscillation, with a period of roughly 15 km and an amplitude of 50 cm. Again using ICESat-2 as truth, we can align a strip to ICESat-2 and compute residuals. If these residuals follow the aforementioned jitter signal, we can compute a smooth surface with which we can correct the DSM strip, thereby improving vertical precision.

1.7.3 Create Mosaic

The individual strips that are acquired over a particular area are unlikely to cover the entire region of interest with just one strip. Therefore multiple strips, each with their own acquisition time and non-uniform spatial extent, must be stitched together to form a single cohesive "mosaic" which spans the entire area of interest. As each strip suffers from its own absolute vertical error, a simple mosaic would introduce large boundary effects, introducing relative vertical errors into the mosaic on the order of the absolute error of individual strips. This step sequentially vertically aligns each strip to a single, primary strip after which the mosaic process is performed, significantly reducing (but not entirely removing) edge effects.

1.7.4 Co-register Mosaic to ICESat-2

The aforementioned process reduces the relative vertical errors of the mosaic, but aligning all strips to a single, primary strip will give the entire mosaic an approximate absolute vertical error equal to that primary strip. In order to reduce this, the full mosaic is then co-registered to ICESat-2, properly aligning the entire mosaic in the vertical.

1.7.5 Compute Inundation Projections

One application where high vertical accuracy is crucial is computing projections of inundation risk due to future sea level rise. With this code we combine mean sea surface heights, high tide values, and projections of future regional sea level change, optionally with vertical land motion included, to compute inundation risk at the same resolution as the DSM.

2 ICESat-2 Download

Script: `GCP_Read_Mask_Write_ICESat2_ATL03.py` in ICESat-2 repository.

This section covers the code in the ICESat-2 repository to download ICESat-2 from the National Snow and Ice Data Center (NSIDC). We use ICESat-2 as a vertical benchmark so we can co-register the DSMs to an absolute vertical truth. Given a set of coordinates to denote a bounding box, the code will download ATL03 geolocated photons for that particular region.

Running the `GCP_Read_Mask_Write_ICESat2_ATL03.py` script will download the data.

2.1 Input File

Figure 2.1 shows an example input for the ICESat-2 download, called `GCP_Input.txt`. The input file requires one header line and needs longitude and latitude extents, describing the area of interest. Temporal filters are possible, but optional, and must be in the YYYY-MM-DD format. If more complex shapes are desired, it is recommended to download the full bounding box extents first and subset later on, as subsetting on the NSIDC's end can be rather slow. Section 7 covers how to do this. Multiple input locations (each on a separate line) in this input file will be processed sequentially.

```
eheijkoop@ubuntu:~$ head ~/INPUTS/GCP_Input.txt
INPUT: NAME, LON MIN, LON MAX, LAT MIN, LAT MAX, TIME START, TIME END
Mexico_Veracruz,-96.35,-95.92,18.98,19.58,2022-01-01,2023-12-31
```

Figure 2.1: Example input for the ICESat-2 download.

2.2 Flags

A number of configuration flags are available and they are described here.

- `-h,--help` : Display all the options.
- `--machine` : Select which machine you are running this script on. The machine name only changes path names, as paths to a particular directory are slightly different from one machine to the other. Defaults to `t`.
- `--landmask` : Mask returned photons and return those over land only. It will subset the OpenStreetMap coastline file to the extents from the input file, and use that to define water and land. It then uses the point-in-polygon algorithm to classify individual pixels. Default is `False`.
- `--time` : Turn on to add a column of timestamps. Default is `False`.
- `--beams` : Turn on to add a column of beams (e.g. gt2r). Combined with the timestamps this is a useful column to isolate individual beams of ICESat-2. Default is `False`.
- `--sigma` : Turn on to add a column of standard deviation values for each photon. Default is `False`.
- `--weak` : Do you want to download weak photons instead? Default is `False`.
- `--N_cpus` : How many CPU cores do you want to use to run the landmask. If the landmask toggle is turned off, this flag is useless. Default is 1.
- `--version` : Which version of ICESat-2 ATL03 do you want to download? Default is 5, because 6 had some rollout issues, but those appear to have been resolved. Version 5 runs until approximately October 2022, whereas version 6 includes data up until approximately March 2023.
- `--copernicus` : Do you want to apply a vertical filter to the photon heights? This flag will download the Copernicus 30 m DEM over the extent of the ICESat-2 input and sample the DEM at the photons' locations. Any difference larger than a certain amount (default is 10 m) will cause the photon to be discarded. This flag will create a new file, the unfiltered output file will always be produced. In newer versions (4 and above) a similar process has already been applied, so the added benefit is marginal now. Default is `False`.

- `--keep_files` : Keep files obtained when applying the `--copernicus` flag, rather than deleting them. Default is `False`.

Note: these flags will be applied to each location in the input file. *I.e.* it is not possible to apply the landmask for one location, but not for the next location; if this is desired you must split the job into multiple batches.

2.3 Configuration File

The `icesat2_config.ini` configuration file has a number of paths that should be changed:

- `GENERAL_CONSTANTS/earthdata_username` : Enter your NASA EarthData username here.
- `GENERAL_CONSTANTS/osm_shp_path` : Path to the full OpenStreetMap shapefile.
- `GENERAL_CONSTANTS/landmask_c_file` : Path to the C code to do the landmasking. The repository comes with this code, so it should be the path to <my_dir>/ICESat-2/C_Code/pnpoly_function.c when you clone the repository into <my_dir>.
- `GCP_PATHS/input_file` : Path to the input file from which the code will read extents to download over.
- `GCP_PATHS/icesat2_dir` : Path to the directory in which subdirectories will be created for each location from the input file.
- `GCP_PATHS/error_log_file` : Path to file that will be created where errors will be noted.
- `GCP_PATHS/EGM96_path` : Path to the EGM96 geoid file with which the SRTM/ASTER DEMs can be converted to WGS 84 heights. (Deprecated and has since been replaced by EGM2008 for the Copernicus DEM.)
- `GCP_PATHS/EGM2008_path` : Path to the EGM2008 geoid file with which the Copernicus DEM can be converted to WGS 84 heights.

Constants:

- `GCP_CONSTANTS/landmask_inside_flag` : Set to 1 (default) to only select polygons on the inside of your shapefile (*i.e.* over land). Set to 0 to only select those outside the shapefile (*i.e.* over water).
- `GCP_CONSTANTS/Copernicus_Threshold` : Vertical threshold (default is 10 m) within which ICESat-2 photons must remain w.r.t. the Copernicus DEM to be considered.

2.4 Example Output

Figure 2.2 shows the results of a successful run for the city of Veracruz, Mexico. Figure 2.3 then shows the first few lines of the resulting file, showing a header line with a description for each column (in this case longitude, latitude, height of the ICESat-2 photon bounce point, time (UTC), beam name and standard deviation of the height). A simple `pd.read_csv()` command will load up all data into a Pandas DataFrame. Note: for large files (several GB in size) this may take a while and memory may be an issue.

```
(geospatial) eheijkoop@ubuntu:~/Scripts/ICESat-2$ python GCP_Read_Mask_Write_ICESat2_ATL03.py
--machine t --landmask --time --beams --sigma --N_cpus 4 --version 6
NASA EarthData password:
Working on Mexico_Veracruz
There are 22 granules of ATL03 over Mexico_Veracruz.
Going for synchronous request.
Downloading for Mexico_Veracruz...
Downloading file 1/1...
Download complete.
Done with Mexico_Veracruz at 2023-06-27 14:58:11
```

Figure 2.2: Example run for the ICESat-2 download, using the input file from Figure 2.1.

Finally, Figure 2.4 shows the spatial extents of this download. Note the OpenStreetMap coastline in white that was used to mask the photons (red).

```
head Mexico_Veracruz_ATL03_high_conf_masked.txt
lon,lat,height_icesat2,time,beam,sigma
-96.297972,18.980012,7.194137,2022-01-20 12:55:57.593948,gt1l,0.129534
-96.297972,18.980019,11.818744,2022-01-20 12:55:57.594048,gt1l,0.129534
-96.297974,18.980038,8.406869,2022-01-20 12:55:57.594348,gt1l,0.129534
-96.297978,18.980070,7.709708,2022-01-20 12:55:57.594848,gt1l,0.129534
-96.297978,18.980070,6.192324,2022-01-20 12:55:57.594848,gt1l,0.129534
-96.297978,18.980070,7.805862,2022-01-20 12:55:57.594848,gt1l,0.129534
-96.297981,18.980101,7.607521,2022-01-20 12:55:57.595348,gt1l,0.129534
-96.297984,18.980127,12.460285,2022-01-20 12:55:57.595748,gt1l,0.129534
-96.297984,18.980133,7.872645,2022-01-20 12:55:57.595848,gt1l,0.129534
```

Figure 2.3: Example results for the ICESat-2 download.

3 Filtering & Corrections

This section will describe a number of optional steps in the DSM mosaic process, each with the goal of improving vertical accuracy of the final product. The following filtering/corrections steps are described: finding DSM strips that are cloudy or over water, filtering the ICESat-2 points for vegetation and inland water, and applying corrections to the DSM strips before including them in the mosaic.

3.1 Optional: Finding Clouds & Water in DSM Strips

Script: `Find_Cloudy_DEMs.py` in DEM repository.

As the DSM is created from optical imagery, clouds can be an issue. Slow moving clouds may still be seen in approximately the same place in both stereo image acquisitions, and this manifests as an artificially high region in the DSM. In many (sub)tropical regions this is an inevitable occurrence and some clouds are still acceptable. However, in some cases a strip is simply too cloudy, which also prevents proper alignment with adjacent strips, acquired on different days when conditions were perhaps less cloudy. This section describes a simple cloud detection algorithm, which can be quickly run for a particular region. Strips that were identified as too cloudy can then be skipped in the mosaic algorithm.

After running this script, use the `--cloud_water_filter` flag for the `Mosaic_Strips.py` script to apply the filter.

3.1.1 Input File

This script can use an input file to run many locations sequentially. Such a file will require a single header line with the locations in which the strips are located and the associated coastline files to delineate between water and land. If the coastline column is left empty, the default OpenStreetMap coastline will be assumed, which will then be subset to spatially match the extent of the input files. Figure 3.1 shows an example input file.

3.1.2 Flags

- `-h,--help` : Display all the options.
- `--input_file` : Path to input file with locations, see Figure 3.1 for an example. One can select either an input file, a specific input directory (`--input_dir`) or a specific list of files to run (`--list`); selecting multiple will throw an error. Defaults to `None`.
- `--input_dir` : Path to a single input directory in which to run the algorithm. Defaults to `None`.
- `--list` : Path to a list containing the desired strips on which to run the algorithm; these paths must be full paths, and no header line should be included in this list file. Defaults to `None`.
- `--loc_name` : Specify a location name when using a specific list. Defaults to `None`.
- `--machine` : Select which machine you are running this script on. The machine name only changes path names, as paths to a particular directory are slightly different from one machine to the other. Defaults to `t`.

Figure 2.4: Example spatial extents for the ICESat-2 download. The red points are the ICESat-2 ATL03 geolocated photons (their high along-track resolution means individual points appear as lines), the white line represents the subsetted OpenStreetMap coastline, and the basemap is from Bing images.

```
(geospatial) eheijkoop@ubuntu:~/Scripts/DEM$ head ~/INPUTS/Africa_Cloud_Water_Input.txt
Location,Coastline
/BhaltosMount/Bhaltos/NASA_SEALEVEL/PRODUCTS/Africa/South_Africa_PortElizabeth
/BhaltosMount/Bhaltos/NASA_SEALEVEL/PRODUCTS/Africa/Sudan_PortSudan
/BhaltosMount/Bhaltos/NASA_SEALEVEL/PRODUCTS/Africa/Tanzania_Dar-es-Salaam
/BhaltosMount/Bhaltos/NASA_SEALEVEL/PRODUCTS/Africa/Togo_Lome
/BhaltosMount/Bhaltos/NASA_SEALEVEL/PRODUCTS/Africa/Tunisia_Bizerte
/BhaltosMount/Bhaltos/NASA_SEALEVEL/PRODUCTS/Africa/Tunisia_Sousse
/BhaltosMount/Bhaltos/NASA_SEALEVEL/PRODUCTS/Africa/Tunisia_Tunis
```

Figure 3.1: Example input for the cloud/water detection algorithm. Note the many directories in alphabetical order, which were processed sequentially.

- `--dir_structure` : How is the directory the strips are in structured? If `simple` is selected, it is assumed that all strips are located directly in the input directory. If `sealevel` is selected, the structure of the `NASA_SEALEVEL` directory is assumed: strips are located inside their respective `WV*/strips/` directory, with or without a `UTM*` directory in between the main directory and the `WV*` directories. Default is `sealevel`.
- `--N_cpus` : How many CPU cores do you want to use to run the cloud/water detection. As each process is independent, the only limit is the number of strips in a location. For example, if a given location contains 6 strips, but 8 CPU cores are requested, only 6 will be used. Default is 1.
- `--a_priori` : Which a priori DEM do you want to download for comparison? Options are `srtm`, `aster` and `copernicus` (all lowercase); as a reminder, downloading SRTM and ASTER requires a NASA EarthData username & password, while downloading Copernicus requires the AWS CLI utility. Default is `copernicus`.
- `--coastline` : Which coastline do you want to use to detect water? Each DSM strip will be clipped to this coastline and anything outside will be considered water, so an accurate coastline is recommended. If this flag is not toggled, it will default to the main OpenStreetMap file as defined by your configuration file, and it will subset that file to the extent of your location of interest.
- `--vertical_threshold` : What (absolute) vertical difference between the *a priori* DEM and each strip is allowable, above which an area is defined as cloudy/erroneous. Default is 50 m.
- `--keep_diff` : Do you want to keep the difference files (between the selected *a priori* DEM and the strips). Default is `False`.
- `--quiet` : Suppresses output. Default is `False`.

3.1.3 Configuration File

The `dem_config.ini` configuration file has a number of paths that should be changed:

- `GENERAL_CONSTANTS/earthdata_username` : Enter your NASA EarthData username here.
- `GENERAL_PATHS/osm_shp_file` : Path to the full OpenStreetMap shapefile.
- `GENERAL_PATHS/EGM96_path` : Path to the EGM96 geoid file with which the SRTM/ASTER DEMs can be converted to WGS 84 heights.
- `GENERAL_PATHS/EGM2008_path` : Path to the EGM2008 geoid file with which the Copernicus DEM can be converted to WGS 84 heights.
- `GENERAL_PATHS/tmp_dir` : Path to directory to hold temporary files.

3.1.4 Example Output

Figure 3.2 shows the results after running the script for one location (albeit in an input file). When the `--quiet` flag is omitted, the results are printed to the screen, and in any case the results are saved as a csv in the directory containing the strips.

```
(geospatial) eheijkoop@ubuntu:~/Scripts/DEM$ python Find_Cloudy_DEMs.py --input_file /home/eheijkoop/INPUTS
/Africa_Cloud_Water_Input.txt --machine t --dir_structure sealevel --a_priori copernicus --vertical_thresho
ld 50 --N_cpus 4
Working on Tunisia_Tunis.
Using default coastline file. Clipping it to DEM extents...
Downloading copernicus...
Download complete.
WV01_20140724_10200100329DA800_1020010032E41D00_seg1_2m_dem_smooth: 7.6% cloudy, 0.1% over water.
WV01_20120317_102001001A7C4800_102001001B2D9F00_seg1_2m_dem_smooth: 0.1% cloudy, 0.1% over water.
WV01_20151013_1020010042B39F00_10200100447B9F00_seg1_2m_dem_smooth: 13.9% cloudy, 0.2% over water.
WV01_20150304_1020010039C1C700_102001003D607C00_seg1_2m_dem_smooth: 11.2% cloudy, 0.2% over water.
WV01_20130412_10200100213BBE00_10200100218 3D600_seg1_2m_dem_smooth: 1.4% cloudy, 0.0% over water.
WV02_20110413_103005000BF96700_103005000BF96800_seg1_2m_dem_smooth: 14.0% cloudy, 0.2% over water.
WV03_20141001_10400100020E8600_104001000264E700_seg1_2m_dem_smooth: 9.6% cloudy, 0.8% over water.
WV01_20140409_102001002DACE600_102001002D956B00_seg1_2m_dem_smooth: 2.8% cloudy, 0.0% over water.
WV01_20170319_102001005FCB8A00_102001005FD39100_seg1_2m_dem_smooth: 0.6% cloudy, 0.1% over water.
Finished Tunisia_Tunis in 7 minutes, 23 seconds.
```

Figure 3.2: Example results for the cloud/water detection algorithm.

3.2 Optional: Filtering ICESat-2 with NDVI & (A)NDWI

Script: `Filter_ICESat2_GEE.py` in Utilities repository.

The ATL03 product's built-in classification for land and water performs poorly in coastal regions, as both datasets to denote land and water are buffered offshore and inland, respectively. Thus, photons over coastal regions are generally classified as both land and water. The landmasking algorithm used to apply an a posteriori mask works well when the coastline file describing the coast is accurate. The OpenStreetMap dataset is generally the best performing product, but in some cases its data is lacking. This product is a living product, with many contributions every day, so oftentimes locations get updated; be sure to regularly re-download the file for it to be as up-to-date as possible.

However, the OSM coastline file often does not include bodies of inland water, like lakes and rivers. ICESat-2 will return measurements over these locations, but the DSM often produces poor results over water. Additionally, as the DSMs are created from optical measurements, they only describe the tops of canopies, buildings, etc (hence DSM and not DTM, or Digital Terrain Model). The photons emitted by ICESat-2, however, can and do penetrate vegetation and return a vertical distribution (between the ground and the top of the canopy) over vegetated areas. In these areas, the vertical distribution of ICESat-2 will introduce a negative bias between it and the DSM.

Thus, it is apparent that a finer classification for surface water and vegetated area. For these purposes we use multispectral Sentinel-2 imagery, obtained over the same region in space and as close as possible in time, from which we compute the the Augmented Normalized Difference Water Index (ANDWI) [Rad et al., 2021] and the Normalized Difference Vegetation Index (NDVI) [de Griend and Owe, 1993]. We use ANDWI rather than the standard NDWI because it shows better performance in detecting surface water. This product has the added benefit of being able to mask out photons over the sea when OpenStreetMap's performance is lacking.

3.2.1 Extra Python Packages

As this script accesses your Google Drive programmatically, extra authentication through Google is needed. Additionally, Google Earth Engine must be installed. To do this, run the following command (in the right environment):

- `conda install -c conda-forge earthengine-api google-auth google-api-python-client google-auth-oauthlib=0.7.1`

For more information on installing and activating Google Earth Engine's Python API see this link and for more information on the Google Drive API see this link.

3.2.2 Input File

No input file is used for this script; instead, path to the ICESat-2 csv with the `--input_file` flag. To process multiple files, copy the full command into a shell script (each line having different ICESat-2 csv files) and run that.

3.2.3 Flags

- `--input_file` : Path to the input ICESat-2 csv file.
- `--machine` : Select which machine you are running this script on. The machine name only changes path names, as paths to a particular directory are slightly different from one machine to the other. Defaults to `t`.
- `--N_cpus` : How many CPU cores do you want to use to run the filtering? The ICESat-2 file will be split up by acquisition date, and each date is then independent. Default is 1.

3.2.4 Configuration File

The `utils_config.ini` configuration file has a number of paths that should be changed:

- `GENERAL_PATHS/tmp_dir` : Path to directory to hold temporary files.
- `GENERAL_CONSTANTS/SCOPES` : Scope variable needed to authenticate credentials for Google Drive, along with token and credentials json files. Default is https://www.googleapis.com/auth/drive.
- `GENERAL_PATHS/landmask_c_file` : Path to the C code to do the landmasking. The repository comes with this code, so it should be the path to <my_dir>/Utilities/C_Code/pnpoly_function.c when you clone the repository into <my_dir>.
- `GDRIVE_PATHS/token_json` : Path to the token json file used to authenticate Google Drive, *e.g.* /home/<user>/.config/googledriveapi/token.json
- `GDRIVE_PATHS/credentials_json` : Path to the credentials json file used to authenticate Google Drive, *e.g.* /home/<user>/.config/googledriveapi/credentials.json

Constants (some of which were derived from GEE's s2cloudless tutorial):

- `GEE_CONSTANTS/DT_SEARCH` : Maximum allowable difference between ICESat-2 acquisition and Sentinel-2 image. Default is 7 days.
- `GEE_CONSTANTS/CLOUD_FILTER` : Maximum image cloud cover percent allowed in image collection. Default is 60%.
- `GEE_CONSTANTS/CLD_PRB_THRESH` : Cloud probability; values greater than are considered cloud. Default is 50%.
- `GEE_CONSTANTS/NIR_DRK_THRESH` : Near-infrared reflectance; values less than are considered potential cloud shadow. Default is 0.15 (15%).
- `GEE_CONSTANTS/CLD_PRJ_DIST` : Maximum distance to search for cloud shadows from cloud edges. Default is 1 km.
- `GEE_CONSTANTS/BUFFER` : Distance with which to buffer the edge of cloud-identified objects. Default is 50 m.
- `GEE_CONSTANTS/SR_BAND_SCALE` : Sentinel-2 Surface Reflectance (SR) band scale. Default is 10^4.
- `GEE_CONSTANTS/NDVI_THRESHOLD` : Threshold above which NDVI pixels are classified as vegetated. Default is 0.5.
- `GEE_CONSTANTS/NDWI_THRESHOLD` : Threshold above which ANDWI pixels are classified as water. Default is 0.0.

3.2.5 Example Output

Figure 3.3 shows the results of running the script for the city of Dakar in Senegal. The final output is a file in the same directory as the original ICESat-2 csv, with `_Filtered_NDVI_NDWI` inserted, *i.e.* the new file is now `Senegal_Dakar_ATL03_high_conf_masked_Filtered_NDVI_NDWI.txt`.

Summing all individual times yields 4029 seconds, or 67 minutes and 9 seconds, *i.e.* the total run time using 8 CPU cores (11 minutes and 59 seconds) yields an increase in speed of a factor of 5.6 compared with processing everything sequentially.

```
(geospatial) eheijkoop@ubuntu:~/Scripts/Utilities$ python Filter_ICESat2_GEE.py --input_file /Bhaltos
Mount/Bhaltos/EDUARD/Projects/DEM/ICESat-2/Senegal_Dakar/Senegal_Dakar_ATL03_high_conf_masked.txt --c
pus 8 --machine t
Working on 0...
Working on 2...
Working on 4...
Working on 6...
Working on 8...
Working on 10...
Working on 12...
Working on 14...
No suitable Sentinel-2 data for 0.
Working on 1...
Processing Sentinel-2 for 6 took 41.4 s.
Applying filter for 6 took 1.9 s.
Working on 7...
Processing Sentinel-2 for 8 took 49.1 s.
Applying filter for 8 took 7.9 s.
Working on 9...
Processing Sentinel-2 for 10 took 81.6 s.
Applying filter for 10 took 1.1 s.
Working on 11...
Processing Sentinel-2 for 4 took 92.2 s.
Applying filter for 4 took 3.5 s.
Working on 5...
Processing Sentinel-2 for 1 took 103.7 s.
Applying filter for 1 took 2.0 s.
Working on 16...
Processing Sentinel-2 for 12 took 110.2 s.
Applying filter for 12 took 16.0 s.
Working on 13...
Processing Sentinel-2 for 2 took 151.3 s.
Applying filter for 2 took 2.0 s.
Working on 3...
Processing Sentinel-2 for 14 took 155.2 s.
Processing Sentinel-2 for 7 took 140.1 s.
Processing Sentinel-2 for 9 took 139.0 s.
Applying filter for 14 took 65.2 s.
Working on 15...
Processing Sentinel-2 for 13 took 98.2 s.
Applying filter for 13 took 4.2 s.
Working on 18...
Processing Sentinel-2 for 11 took 153.4 s.
Processing Sentinel-2 for 5 took 145.4 s.
Applying filter for 7 took 59.5 s.
Working on 20...
Applying filter for 11 took 14.2 s.
Working on 22...
Applying filter for 5 took 11.1 s.
Working on 24...
Processing Sentinel-2 for 16 took 162.5 s.
Processing Sentinel-2 for 3 took 115.5 s.
Processing Sentinel-2 for 15 took 51.4 s.
Applying filter for 3 took 7.9 s.
Working on 26...
Applying filter for 15 took 6.5 s.
Working on 28...
Applying filter for 9 took 107.0 s.
Working on 30...
Processing Sentinel-2 for 24 took 60.3 s.
Processing Sentinel-2 for 18 took 89.1 s.
Applying filter for 24 took 6.7 s.
Working on 25...
No suitable Sentinel-2 data for 25.
Working on 32...
Processing Sentinel-2 for 22 took 88.1 s.
Processing Sentinel-2 for 28 took 69.7 s.
Processing Sentinel-2 for 20 took 106.2 s.
Applying filter for 22 took 14.0 s.
Working on 23...
Applying filter for 18 took 35.1 s.
Working on 19...
Applying filter for 28 took 6.3 s.
Working on 29...
Applying filter for 20 took 8.8 s.
Working on 21...
Processing Sentinel-2 for 26 took 82.4 s.
Applying filter for 26 took 6.5 s.
Working on 27...
Processing Sentinel-2 for 23 took 47.6 s.
Processing Sentinel-2 for 29 took 47.8 s.
Applying filter for 23 took 6.4 s.
Applying filter for 29 took 9.5 s.
Processing Sentinel-2 for 32 took 96.0 s.
Processing Sentinel-2 for 30 took 116.8 s.
Applying filter for 30 took 12.0 s.
Working on 31...
Applying filter for 32 took 18.5 s.
Processing Sentinel-2 for 27 took 98.7 s.
Applying filter for 27 took 8.5 s.
Processing Sentinel-2 for 19 took 155.4 s.
Processing Sentinel-2 for 21 took 157.6 s.
Applying filter for 16 took 247.6 s.
Working on 17...
Applying filter for 21 took 17.6 s.
Processing Sentinel-2 for 31 took 114.0 s.
Applying filter for 31 took 19.9 s.
Processing Sentinel-2 for 17 took 57.9 s.
Applying filter for 19 took 81.2 s.
Applying filter for 17 took 42.4 s.
Senegal_Dakar took 11 minute(s), 59.2 s.
```

Figure 3.3: Results of running the ICESat-2 NDVI/ANDWI filter on an ICESat-2 csv.

3.3 Optional: Correcting DSMs with ICESat-2

Script: `Correct_DEM_Strip_ICESat2.py` in DEM repository.

3.3.1 Input File

No input file is used for this script; instead, path to the ICESat-2 csv with the `--icesat2` flag. To process multiple locations, copy the full command into a shell script (each line having different paths to directories and ICESat-2 csv files) and run that.

3.3.2 Flags

- `--input_file` : Path to input DSM to correct. Use this when only doing a single file. Default is `None`.
- `--input_dir` : Path to directory containing DSM strips to correct. Use this when you want to correct a whole directory in one go. Default is `None`.
- `--list` : Path to input list of DEMs to correct; list must contain full paths. Default is `None`.
- `--machine` : Select which machine you are running this script on. The machine name only changes path names, as paths to a particular directory are slightly different from one machine to the other. Defaults to `t`.
- `--dir_structure` : How is the directory the strips are in structured? If `simple` is selected, it is assumed that all strips are located directly in the input directory. If `sealevel` is selected, the structure of the `NASA_SEALEVEL` directory is assumed: strips are located inside their respective `WV*/strips/` directory, with or without a `UTM*` directory in between the main directory and the `WV*` directories. Default is `sealevel`.
- `--N_cpus` : Number of CPUs to use. Each DSM strip correction is an independent process, so the script will find the minimum value between this input and the number of strips to process. Default is 1.
- `--icesat2` : Path to ICESat-2 file used to correct DSMs. Default is `None`.
- `--mean` : Toggle this flag to aim for zero-mean alignment with ICESat-2. Default is `False`.
- `--median` : Toggle this flag to aim for zero-median alignment with ICESat-2. Default is `False`.
- `--sigma` : How many standard deviations away from the mean/median do you want to discard? Larger values will discard fewer points, but will likely end up with a higher uncertainty., while lower values will discard more points. Default is 2.
- `--threshold` : What value for an iterative alignment of DSM strip to ICESat-2 constitutes convergence? This is set to avoid infinite loops. Default is 0.02 m.
- `--print` : Toggle to print statistics of alignment and corrections upon completion. Default is `False`.
- `--keep_files` : Toggle to keep intermediate files, such as the jitter correction file. Default is `False`.
- `--a_priori` : Filter with the Copernicus DEM? Default is `False`.
- `--coastline` : Apply a coastline filter to the DSM & a priori DEM? Default is `None`.

3.3.3 Configuration File

The `dem_config.ini` configuration file has a number of paths that should be changed:

- `GENERAL_PATHS/tmp_dir` :
- `GENERAL_CONSTANTS/earthdata_username` :
- `GENERAL_PATHS/EGM2008_path` :

Constants:

- CORRECTIONS_CONSTANTS/N_coverage_minimum :
- CORRECTIONS_CONSTANTS/N_photons_minimum :
- CORRECTIONS_CONSTANTS/faulty_pixel_height_threshold :
- CORRECTIONS_CONSTANTS/faulty_pixel_pct_threshold :

3.3.4 Example Output

4 Mosaic

Script: Mosaic_Strips.py in DEM repository.

4.1 Input File

Figure 4.1 shows an example input file for the mosaic code, called `MOSAIC_Input.txt`. This file requires one header line and requires an input and output directory. An optional third column can be used to denote either old or new versions of DSM strips (previous versions of `scenes2strips.py` produced `*dem_smooth.tif` and `*dem_browse.tif` files rather than `*dem.tif` and `*dem_10m.tif` for high and low resolution, respectively); leaving this column empty will simply find all types and that is the recommended method. The first location must point to a directory containing DSM strips and the second column will contain the output (in a separate `/Mosaic` directory). If this latter directory does not yet exist, it will be created. Similar to other input files, multiple locations on separate lines will be processed sequentially.

```
(geospatial) eheijkoop@ubuntu:~/Scripts/DEM$ head ~/INPUTS/MOSAIC_Input.txt
Location of strips,Output location,Output type:
/BhaltosMount/Bhaltos/NASA_SEALEVEL/PRODUCTS/Africa/Tunisia_Tunis/,/BhaltosMount/Bhaltos/EDUARD
/Projects/DEM/Africa/Tunisia_Tunis/
```

Figure 4.1: Example input for the Mosaic code.

4.2 Flags

- `-h,--help` : Display all the options.
- `--input_file` : Specify a different input file than the default one. Default is `None`.
- `--list` : Specify a list of (full paths to) strips to mosaic. If the input file is not empty, it will do this list as a separate entry at the end, otherwise it only does the list. Default is `None`.
- `--output_dir` : Specify an output directory for the strips in the list. Default is `None`.
- `--loc_name` : Specify a name for the strips in the list. Default is `None`.
- `--machine` : Select which machine you are running this script on. The machine name only changes path names, as paths to a particular directory are slightly different from one machine to the other. Defaults to `t`.
- `--dir_structure` : How is the directory the strips are in structured? If `simple` is selected, it is assumed that all strips are located directly in the input directory. If `sealevel` is selected, the structure of the `NASA_SEALEVEL` directory is assumed: strips are located inside their respective `WV*/strips/` directory, with or without a `UTM*` directory in between the main directory and the `WV*` directories. Default is `sealevel`.
- `--N_cpus` : How many CPU cores do you want to use to do alignment of strips in parallel. The number of cores actually used will depend on how many independent alignments can be done per step. See Figure 4.2, each individual "generation" only has one alignment, so only 1 CPU core will be used, despite 6 being requested. Default is 1.
- `--horizontal` : Toggle to include horizontal shifting when aligning one strip to another to find the best alignment. Default is `False`.
- `--cloud_water_filter` : Toggle to turn on cloud/water filter. It will find the relevant `*_Threshold_Exceedance_Values.txt` file in the input directory and filter strips based on values in `dem_config.ini`. Default is `False`.
- `--corrected` : Toggle to create the mosaic with the corrected strips instead of the original ones. Default is `False`.
- `--all_strips` : Toggle to force include all strips and skip the geometric filtering (e.g. when two strips fully overlap another, older one). Default is `False`.

- `--gsw` : Path to different surface water file, e.g. derived from Sentinel-2. Usually incompatible with a batch of different locations, as the surface water file is assigned to every location. Default is `False`.

- `--no_gsw` : Toggle to skip the GSW filter, which may be useful for regions where there is a lot of water, *e.g.* tropical islands. Default is `False`.

- `--simplify` : Flag to toggle simplification of the strips' outline geometry. Simplification will speed up masking of points for strip-to-strip alignment, but at the expense of accuracy (though a 10 m simplification radius when points are sampled every 20 m should not cause any issues). Default is `False`.

4.3 Configuration File

The `dem_config.ini` configuration file has a number of paths that should be changed:

- `GENERAL_PATHS/tmp_dir` : Path to directory to hold temporary files.

- `GENERAL_PATHS/gsw_dir` : Path to directory containing Global Surface Water [Pekel et al., 2016] extent tiles.

- `GENERAL_PATHS/landmask_c_file` : Path to the C code to do the landmasking. The repository comes with this code, so it should be the path to `<my_dir>/DEM/C_Code/pnpoly_function.c` when you clone the repository into `<my_dir>`.

- `GENERAL_PATHS/osm_shp_file` : Path to the full OpenStreetMap shapefile.

- `GENERAL_PATHS/EGM96_path` : Path to the EGM96 geoid file with which the SRTM/ASTER DEMs can be converted to WGS 84 heights. (Deprecated and has since been replaced by EGM2008 for the Copernicus DEM.)

- `GENERAL_PATHS/EGM2008_path` : Path to the EGM2008 geoid file with which the Copernicus DEM can be converted to WGS 84 heights.

- `GENERAL_CONSTANTS/earthdata_username` : Enter your NASA EarthData username here.

- `MOSAIC_PATH/input_file` : Path to the mosaic input file.

Constants:

- `MOSAIC_CONSTANTS/POLYGON_AREA_THRESHOLD` : When a strip's true outline is created, a number of small pockets may appear. This value represents the threshold for these individual polygons' surface area to be included in the final GeoDataFrame for a single strip. Defaults to 250.0 m^2.

- `MOSAIC_CONSTANTS/POLYGON_SIMPLIFY_VALUE` : Simplify radius if the `--simplify` flag is on. Defaults to 10.0 m.

- `MOSAIC_CONSTANTS/STRIP_AREA_THRESHOLD` : Threshold area for a strip to be included. Any strip with a smaller surface area will be discarded (*i.e.* removed from consideration, not deleted). Defaults to 4×10^6 m^2.

- `MOSAIC_CONSTANTS/GSW_POCKET_THRESHOLD` : Minimum percentage of the total surface water area that will get included in the final surface water shapefile, reducing computational complexity by decreasing the number of polygons. For example, many small pockets of surface water are much smaller than 1% of the total area, and therefore these will get discarded. Defaults to 0.01 (1%).

- `MOSAIC_CONSTANTS/GSW_CRS_TRANSFORM_THRESHOLD` : Minimum percentage of polygon area with respect to the total area after transformation from EPSG:4326 to the local UTM zone. Sometimes this transformation introduces small square pockets that just increase the final polygon's complexity. Defaults to 0.05 (5%).

- `MOSAIC_CONSTANTS/GSW_OVERLAP_THRESHOLD` : Percentage that defines whether or not a strip is entirely covered by the surface product. Such strips will be removed from consideration. Defaults to 0.95 (95%).

- MOSAIC_CONSTANTS/STRIP_TOTAL_AREA_PERCENTAGE_THRESHOLD : Minimum area percentage of a polygon within a larger set of polygons describing a strip's outline to be considered, otherwise it is discarded. This speeds up computations of intersection and overlap. Defaults to 0.01 (1%).
- MOSAIC_CONSTANTS/STRIP_CONTAINMENT_THRESHOLD : Area (as a percentage) of a particular strip that needs to be covered by the union of N newer strips, where N is equal to N_STRIPS_CONTAINMENT. All N strips must be newer, how much newer is defined next by STRIP_DELTA_TIME_THRESHOLD. If, for example, $N = 2$ and we have two strips from 2018 and 2020, which together cover more than 75% of an older strip's area, that older strip will be removed from consideration. This removes unnecessary strip-to-strip alignments, each of which introduces uncertainty. Defaults to 0.75 (75%).
- MOSAIC_CONSTANTS/STRIP_DELTA_TIME_THRESHOLD : Difference in time (in days) for a strip to be considered newer. Non-negative values (*e.g.* 100) mean that there must be more of a temporal gap for a strip to be considered newer, when assessing overlap of one strip by N others. Defaults to 0 days.
- MOSAIC_CONSTANTS/STRIP_CLOUD_THRESHOLD : Percentage of strip that is allowed to be "cloudy" (off by more than 50 m, or whatever was user-defined when the algorithm was run, from an *a priori* DEM); a higher value will cause the strip to be discarded. Defaults to 0.15 (15%).
- MOSAIC_CONSTANTS/STRIP_WATER_THRESHOLD : Percentage of strip that is allowed to be over water (when clipped by a coastline); a higher value will cause the strip to be discarded. Defaults to 0.75 (75%).
- MOSAIC_CONSTANTS/N_STRIPS_CONTAINMENT : Number of strips of which the outlines' union must be larger than a single, older strip for that older strip to be removed from consideration. Defaults to 2.
- MOSAIC_CONSTANTS/AREA_OVERLAP_THRESHOLD : Minimum overlapping area between two strips for an overlap to be considered valid. A smaller area will lead to fewer points sampled in this overlapping area, which reduces the accuracy of the strip-to-strip alignment. Defaults to 2.5×10^5 m^2.
- MOSAIC_CONSTANTS/GSW_INTERSECTION_THRESHOLD : Maximum percentage that the overlapping area of two strips is allowed to be covered by the surface water product. This ensures that not too much of this overlapping area is covered by water, as the water masking reduces the number of points available for alignment. Defaults to 0.667 (66.7%).
- MOSAIC_CONSTANTS/X_SPACING : Spacing of points in x direction to sample primary and secondary strips in their respective overlapping areas. Defaults to 20.0 m.
- MOSAIC_CONSTANTS/Y_SPACING : Spacing of points in y direction to sample primary and secondary strips in their respective overlapping areas. Defaults to 20.0 m.
- MOSAIC_CONSTANTS/X_MAX_SEARCH : Maximum distance in x direction to shift when applying --horizontal. Defaults to 12.0 m.
- MOSAIC_CONSTANTS/Y_MAX_SEARCH : Maximum distance in y direction to shift when applying --horizontal. Defaults to 12.0 m.
- MOSAIC_CONSTANTS/MOSAIC_TILE_SIZE : Intermediate tile size to build up the mosaic. Defaults to 25000.0 m, leading to a 25×25 km^2 tile.

4.4 Example Output

Figure 4.2 shows the results of running the mosaic code for Tunis, Tunisia, taking less than an hour. The statistics of strip-to-strip alignment are also saved to a `*Statistics.txt` file in the `Mosaic` directory. The final mosaic file is then called `Tunisia_Tunis_Full_Mosaic_0_32632.tif`, where the 0 represents the first mosaic (most often only one mosaic is produced, but in the case of a lack of overlap, multiple mosaics may be produced) and 32632 represents the EPSG code for the projection.

```
(geospatial) eheijkoop@ubuntu:~/Scripts/DEM$ python Mosaic_Strips.py --horizontal --machine t
--cloud_water_filter --simplify --cpus 6

Working on Tunisia_Tunis
EPSG:32632
Loading strips...
[====================] 100%
Applying cloud/water filter...
Building mosaic(s) from 1 group(s) & 0 single strip(s).

Mosaic 0:
Starting at: 0
/BhaltosMount/Bhaltos/NASA_SEALEVEL/PRODUCTS/Africa/Tunisia_Tunis/UTM32S/WV01_20170319_1020010
05FCB8A00_102001005FD39100/strips/WV01_20170319_102001005FCB8A00_102001005FD39100_seg1_2m_dem_
smooth.tif

Generation: 0
Strip 1 to strip 0

Generation: 1
Strip 2 to strip 1

Generation: 2
Strip 3 to strip 2

Linking 1 (WV01) to 0 (WV01)...
Optimal horizontal shift: x = 4.0 m, y = 2.0 m
RMSE (zero shift): 0.62 m, RMSE (optimal shift): 0.59 m
Relative RMSE: 0.95
Results for 1 (WV01) to 0 (WV01):
Retained 76.7% of points.
Vertical shift: 2.83 m
RMSE: 0.59 m
Linking 2 (WV01) to 1 (WV01)...
Optimal horizontal shift: x = 4.0 m, y = 2.0 m
RMSE (zero shift): 0.64 m, RMSE (optimal shift): 0.59 m
Relative RMSE: 0.93
Results for 2 (WV01) to 1 (WV01):
Retained 74.8% of points.
Vertical shift: 1.73 m
RMSE: 0.59 m
Linking 3 (WV01) to 2 (WV01)...
Optimal horizontal shift: x = 8.0 m, y = 2.0 m
RMSE (zero shift): 0.93 m, RMSE (optimal shift): 0.64 m
Relative RMSE: 0.69
Results for 3 (WV01) to 2 (WV01):
Retained 89.1% of points.
Vertical shift: 1.54 m
RMSE: 0.64 m

Mosaicing...

Finished with Tunisia_Tunis in EPSG:32632.
It took:
0 hours, 50 minutes, 50.759836 seconds
```

Figure 4.2: Example results for the Mosaic code.

5 Co-registration

Script: `Simple_Coregistration.py` in Utilities repository.

5.1 Input File

5.2 Flags

5.3 Configuration File

5.4 Example Output

6 Inundation

Script: `Compute_Inundation.py` in DEM repository.

6.1 Methodology

6.2 Input File

No input file is used for this script; instead, path to the files with the relevant flags. To process multiple locations, copy the full command into a shell script (each line having different paths to files and extents) and run that.

6.3 Flags

- `--input_file` : Which file do you want to run the inundation computation on? It is highly recommended to run this on a file that has been co-registered to a vertical truth, *e.g.* ICESat-2.
- `--loc_name` : What name do you want for the output inundation files? If none is given (*i.e.* this flag is omitted) a name will be detected from the input file.
- `--machine` : Select which machine you are running this script on. The machine name only changes path names, as paths to a particular directory are slightly different from one machine to the other. Defaults to `t`.
- `--geoid` : Do you want to use a geoid file to correct everything from ellipsoidal to orthometric heights? Default is `None`, and no measurable difference in results were obtained when testing with the EGM2008 geoid.
- `--vlm` : What vertical land motion (VLM) file do you want to use to propagate the DSM forward in time? If a single value is given instead of a filename, this constant value will be applied to the whole DSM.
- `--clip_vlm` : Do you wish to clip the DSM extents to that of the VLM file? It may be the case that one only wants to consider part of the DSM that overlaps the VLM, in which case this flag should be toggled.
- `--icesat2` : Input ICESat-2 file of ocean heights. This functionality works, but deriving an ocean heights product from ICESat-2 is still experimental. Conflicts with `--sealevel_grid` and only one should be selected.
- `--sealevel_grid` : Input sea level grid to use as mean sea level. Conflicts with `--icesat2` and only one should be selected.
- `--grid_extents` : Extents with which to clip the input sea level grid, which is particularly useful when using global grids. Requires 4 arguments in the order of $x_{\min}$, $x_{\max}$, $y_{\min}$, $y_{\max}$, where x and y may be substituted by longitude and latitude, respectively, as long as these coordinates are in the same projection as the grid.
- `--coastline` : Path to coastline from which to compute inundation (and optionally connectivity). It is recommended to use the shapefile that is created in the ICESat-2 download process, as this will/should have the same extents as the DSM, unless the user can supply a more accurate file.
- `--clip_coast` : Toggle to clip the DSM to the aforementioned coastline. Generally this is a good idea as the DSM might include artifacts over water, which will influence the results. Default is `False`.
- `--years` : For which years should inundation be calculated? Requires either the `--rcp` or `--ssp` flag. Years must be divisible by 10, *e.g.* 2070 works, but 2054 will not; this is due to the temporal resolution of the regional sea level projections.
- `--rcp` : Use this with an argument to select which AR5 RCP (Representative Concentration Pathway) to use for sea level projections. Options are (select only one): `2.6`, `4.5` or `8.5`. Conflicts with `--ssp` and only one should be selected.

- `--ssp` : Use this with an argument to select which AR6 SSP (Shared Socioeconomic Pathway) to use for sea level projections. Options are (select only one): `1-1.9`, `1-2.6`, `2-4.5`, `3-7.0` or `5-8.5`. Inputs are allowed to have either the hyphen or dot, both, or none, and may be pre-appended with `SSP` in lower or upper case, *e.g.* `SSP119` and `1-1.9` will use the same projection. Conflicts with `--rcp` and only one should be selected.

- `--slr` : Select this flag to apply a uniform sea level rise value to the entire grid. Values should be in meters, and there is no limit to how many arguments are given to this flag, e.g. `--slr 0.3 0.5 0.7 1.0` is a valid argument, and will run inundation for 30 cm, 50 cm, 70 cm and 1.0 m.

- `--t0` : For either the `--rcp` or `--ssp` flags, set a t_0, which defaults to 2020. Note that this value is only used to propagate the VLM forward in time. Although the AR6 projections have a temporal baseline of 1995-2014, many sea level grids have similar baselines (such as DTU21's of 1993-2012), which means that the temporal component of propagating mean sea level to present day is already included with the AR6 projections.

- `--return_period` : Selected return period for sea level extreme values, derived from the CoDEC dataset [Muis et al., 2020]. Select one of 2, 5, 10, 25, 50, 100, 250, 500 or 1000 years. Conflicts with `--fes2014` and `--high_tide`, and only one should be selected.

- `--fes2014` : Use max tidal heights from the FES2014 ocean tide model [Lyard et al., 2021] as high tide values; these will be systematically lower than the sea level extremes from CoDEC. Conflicts with `--return_period` and `--high_tide`, and only one should be selected.

- `--mhhw` : Use mean higher high water (MHHW) instead of maximum tidal heights from the FES2014 model. Must be used in conjunction with the `--fes2014` flag.

- `--high_tide` : Select a single high tide value for a particular region, *e.g.* when local tide gauge data is used instead of CoDEC or FES2014. Value must be in meters. Conflicts with `--return_period` and `--fes2014`, and only one should be selected.

- `--connectivity` : Toggle to compute connectivity of inundation with the open ocean. This allows one to distinguish between pockets of low lying areas in the DSM further inland that are less prone to direct inundation from the sea, vs areas that are directly vulnerable to sea level rise. This will produce separate files with `_connected_GSW` appended to them, as the default method uses the Global Surface Water dataset, unless a coastline derived from ANDWI is used, in which case its inverse is used.

- `--uncertainty` : Compute uncertainty, based on quantiles of sea level change in the AR6 data. Only compatible with the `--ssp` flag and requires the `--sigma` flag with an integer 1, 2 or 3 as argument.

- `--sigma` : Number of standard deviations away from the median to compute uncertainty bounds. Requires `--uncertainty` to be toggled as well.

6.4 Configuration File

Paths:

- `INUNDATION_PATHS/SROCC_dir` : Path to the SROCC directory containing AR5 sea level projections.

- `INUNDATION_PATHS/AR6_dir` : Path to the directory containing AR6 sea level projections.

- `INUNDATION_PATHS/CODEC_file` : Path to the CoDEC netCDF file with sea level extremes. Available here.

- `INUNDATION_PATHS/fes2014_file` : Path to the FES2014 file with longitude, latitude, max tide heights and optionally MHHW. Use code in Utilites repository to create this, which will need the FES2014 constituent files.

Constants:

- `INUNDATION_CONSTANTS/ICESAT2_GRID_RESOLUTION` : If using an ICESat-2 file as a mean sea level input, at what resolution should the point cloud be gridded? Defaults to 500 m.
- `INUNDATION_CONSTANTS/N_PTS` : If using an ICESat-2 file as mean sea level, how many points in a grid cell are required for a cell to be valid? Default is 200. This works for a point cloud derived from the ATL03 product, which is denser than derived products, such as ATL12. When using a derived product, you should decrease this number.
- `INUNDATION_CONSTANTS/INTERPOLATE_METHOD` : If using an ICESat-2 file as mean sea level, what interpolation method method do you want to use to interpolate sea level onto the coast? Options are `Smooth` or `LSQ`, corresponding to the SmoothBivariateSpline and LSQBivariateSpline interpolation methods from scipy. Default is `Smooth`.
- `INUNDATION_CONSTANTS/GRID_NUM_THREADS` : How many threads to use when creating a grid from the coastal sea level points? Default is 4.
- `INUNDATION_CONSTANTS/GRID_INTERMEDIATE_RES` : What spatial resolution do you want to use for your temporary grid files? Default is 100 m.
- `INUNDATION_CONSTANTS/GRID_ALGORITHM` : What algorithm do you want to use to create a regular grid from the coastal sea level points, at the aforementioned resolution? Options are `nearest`, `invdist` and `invdistnn`. Default is `invdistnn`.
- `INUNDATION_CONSTANTS/GRID_SMOOTHING` :
- `INUNDATION_CONSTANTS/GRID_POWER` :
- `INUNDATION_CONSTANTS/GRID_NODATA` :
- `INUNDATION_CONSTANTS/GRID_MAX_PTS` :
- `INUNDATION_CONSTANTS/RETURN_PERIOD` :
- `INUNDATION_CONSTANTS/INUNDATION_NODATA` :
- `INUNDATION_CONSTANTS/GSW_BUFFER` :
- `INUNDATION_CONSTANTS/REGRID_INTERPOLATE_METHOD` :

6.5 Example Output

7 Other Useful Code

7.1 Convert Coordinates to a shapefile/geojson

Script: `Convert_Coords_to_File.py` in Utilities repository.

Use `Convert_Coords_to_File.py` to quickly turn a set of coordinates into a simple shapefile or geojson.

References

de Griend, A. A. and Owe, M.
On the relationship between thermal emissivity and the normalized difference vegetation index for natural surfaces.
International Journal of Remote Sensing, 14:1119–1131, 1993.

Lyard, F. H., Allain, D. J., Cancet, M., Carrère, L., and Picot, N.
FES2014 global ocean tide atlas: Design and performance.
Ocean Science, 17:615–649, 5 2021.
ISSN 18120792.
doi: 10.5194/OS-17-615-2021.

Muis, S., Apecechea, M. I., Dullaart, J., de Lima Rego, J., Madsen, K. S., Su, J., Yan, K., and Verlaan, M.
A High-Resolution Global Dataset of Extreme Sea Levels, Tides, and Storm Surges, Including Future Projections.
Frontiers in Marine Science, 7, 4 2020.
ISSN 2296-7745.
doi: 10.3389/fmars.2020.00263.

Neumann, T., Brenner, A., Hancock, D., Robbins, J., Saba, J., Harbeck, K., Gibbons, A., Lee, J., Luthcke, S., and Rebold, T.
Algorithm Theoretical Basis Document (ATBD) for Global Geolocated Photons ATL03 , 11 2021.
URL `https://nsidc.org/sites/default/files/icesat2_atl03_atbd_r005_0.pdf`.

Noh, M.-J. and Howat, I. M.
Automated stereo-photogrammetric DEM generation at high latitudes: Surface Extraction with TIN-based Search-space Minimization (SETSM) validation and demonstration over glaciated regions.
GIScience & Remote Sensing, 52:198–217, 2015.

OpenStreetMap contributors.
Planet dump retrieved from https://planet.osm.org , 2017.

Pekel, J.-F., Cottam, A., Gorelick, N., and Belward, A. S.
High-resolution mapping of global surface water and its long-term changes.
Nature, 540:418–422, 2016.

Rad, A. M., Kreitler, J., and Sadegh, M.
Augmented Normalized Difference Water Index for improved surface water monitoring.
Environmental Modelling & Software, 140:105030, 6 2021.
ISSN 1364-8152.
doi: 10.1016/J.ENVSOFT.2021.105030.