dhdt: A photohypsometric Python library to estimate glacier elevation change via optical remote sensing imagery

14 November 2022

Summary

Extracting historical and present day measurements of mountain glacier change is challenging. Their locations are remote and their population is vast[1]. Consequently, satellite Earth observation are an ideal means to collect information about their geometry and surface characteristics. Such information can be used for contemporary climate reconstruction and improve future projection of fresh water availability or their contribution to sea-level.

The archive of satellite missions with geometric mapping capabilities is very limited in space and time. While optical missions with a single telescope are more abundant. Hence, methodologies that are able to get at least some information out are worth exploring.

`dhdt` is Python library that has the functionality to automatically extract glacier elevation change from large collections of satellite imagery. The different functions are organised in modules, and each function is documented in a readthedocs.

Statement of need

Regional or global scale surface mass-balance modeling is mainly forced through atmospheric circulation models (Lenaerts et al. 2019; Zekollari et al. 2022). Where in-situ stake measurements are used for calibration, or validation. While remote sensing data is able to provide additional observations of the (sub)-surface.

Such remote sensing observations can be for longer time periods, where the length changes from glacier outlines, can sometimes date back up a century or more. While at decadal timescales the use of elevation models can help to constrain a time-spans, as it delivers an estimate of volume changes (a.k.a. geodetic mass balance) (Hugonnet et al. 2021). While, velocity fields can be used

[1] current estimates of glaciers on Earth come to almost 200'000

to infer current ice thickness (Millan et al. 2022). At annual time scales, the transient snowline can be linked to the surface mass balance (Davaze et al. 2018; Barandun et al. 2021). Enhancing the ability to glacier specific mass-balance functions.

Extraction of elevation data from shadow cast has been around for some time in the computer vision community. But recently this has gone into the wild, exploiting real world data. Here we have dubbed this technique photohypsometry, but other terms like shape-from-shadowing (Daum and Dudek 1998) and heliometric stereo (Abrams, Hawley, and Pless 2012) have also been used for this methodology. This methodology has gotten some interest in the cryospheric research community [Rada Giacaman (2022);], though up to now it has stayed at the level of a proof of concept.

Here, we present a Pyhton library that includes a complete and highly automated processing pipeline. The procedures of such a photohypsometic pipeline are illustrated in Figure 1.

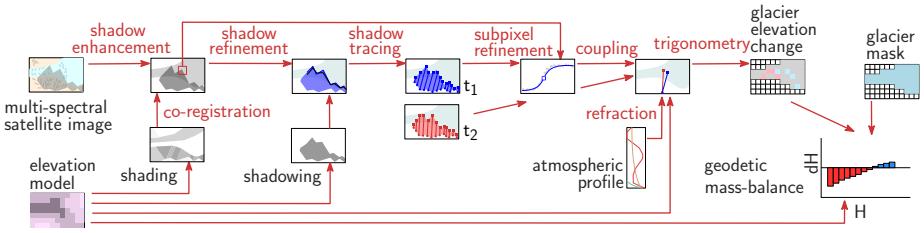

Figure 1: Schematic of the photohypsometric pipeline.

Design and data sources

The data structure of the `dhdt` library follows a data processing pipeline, where general functions call more detailed and refactured components. The top level structure is as follows:

```
dhdt
├── input
├── preprocessing
├── processing
├── postprocessing
├── presentation
├── generic
└── auxilary
```

When a specific function is based upon a given methodology, the literature that is at its root is given in the doc-string. Hence the references in this work are by far not comprehensive.

input

A suit of high resolution[2] optical satellite systems are currently in space, of which many have adopted an open access policy. Hence, a large selection of these satellite data is supported by the `dhdt` library. Though an emphasis is given to the Senintel-2 system (`read_sentinel2`), which is in operation since XXXX.

Though support is also given to the Landsat (`read_landsat8`), SPOT legacy () and ASTER (`read_aster`), as well as, commercial satellite systems (`read_planetscope`, `read_rapideye`) and demonstration missions like VENuS.

Typical functions include loading of the imagery, and associated meta-data. These are functions about the instrument specifics or the acquisition situation.

```
input
├── read_sentinel2.py
├── read_landsat8.py
├── read_spot4.py
├── read_spot5.py
├── read_aster.py
├── read_hyperion.py
├── read_venus.py
├── read_rapideye.py
├── read_planetscope.py
└── read_era5.py
```

preprocessing

Prior to the coupling of imagery and the extraction of elevation change data, the satellite data needs to be made ready. Typically, shadows are not the main interest for such imagery, hence the functions in this sub-directory are taylored towards enhancing the shadow imagery and getting the proper geometric information needed. The structure of the folder is as follows,

```
preprocessing
├── handler_multispec.py
├── color_transforms.py
├── image_transforms.py
├── shadow_transforms.py
├── shadow_filters.py
├── shadow_geometry.py
├── acquisition_geometry.py
└── atmospheric_geometry.py
```

Most optical remote sensing instruments record multiple spectral intervals at (nearly-)simultaneously. Such spectral information can de used to separate out the surface reflectivity (albedo) from the illumination component. This illumination component is a combination of shading and shadowing. A multitude

[2]here high resolution denotes imagery with a pixel spacing in the order of 5 to 30 meters

of methods have been developed (Tsai 2006),(Tsai 2006), (Tsai 2006) and these and more are implemented in `shadow_transforms`.

An even richer collection can be implemented when analysis in the temporal domain are included. Though such methods enforce a stable acquisition setup, which is not always guaranteed for spaceborne acquisitions. However, apart from the spectral methods mentioned above, the spatial neighbourhood can also be used to enhance the performance of the shadow localisation. Such methods are included within `shadow_filters` Where local or zonal neighbourhood filters can reduce speckle noise in the classification, see also an example in Figure 3.

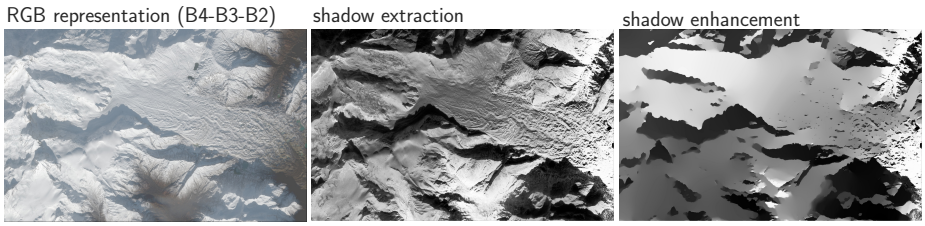

Figure 2: Example of the different preprocessing functions.

Apart from the signal separation to enhance shadows, a suit of functions is present in `dhdt` to deal with parsing of geometric information. The metadata of the instrument needs to be combined with terrain properties. Hence in `acquisition_geometry` coupling of the acquisition geometry of the satellite scenes are coupled to different terrain angles.

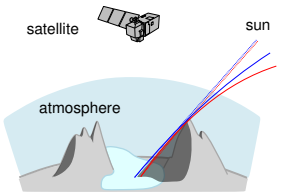

Figure 3: Acquisition geometry with an emphasis on refraction.

The atmosphere is also an element within the photohypsometric chain to take into account, since the sun traces are bended through the atmosphere. This refraction is typically not present in the meta data of the satellite, and only the geometric line of sight is given, see also Figure 3. However, correct sun angles are essential, especially for scenes with low sun angles (such situations occur especially often at high latitudes, where many glaciers are situated). Therfore, `atmospheric_geometry` has functions to correct for such angles given specific wavelengths and atmospheric compositions.

processing

An important building block of `dhdt` is image correspondence, this is an ill-posed problem. Hence, a suit of methodologies are implemented in this library.

Image correspondence

Generic .

```
processing
├── coregistration.py
├── coupling_tools.py
├── network_tools.py
├── matching_tools_organization.py
├── gis_tools.py
├── matching_tools.py
├── matching_tools_frequency_filters.py
├── matching_tools_frequency_correlators.py
├── matching_tools_frequency_subpixel.py
├── matching_tools_frequency_metrics.py
├── matching_tools_spatial_correlators.py
├── matching_tools_spatial_subpixel.py
├── matching_tools_spatial_metrics.py
├── matching_tools_differential.py
├── geometric_precision_describtion.py
├── matching_tools_geometric_temporal.py
└── photohypsometric_image_refinement.py
```

In the `dhdt` library the different approaches to implement image correspondence [3] is subdivided into three domains. The first major subdivision, are implementation based on Fourier based methods. Here, imagery are transformed into the frequency domain, which is computational efficient. The second subdivision are methods formulated in the spatial domain, which are mostly based upon convolution. The last subdivision is also formulated in the spatial domain, but is based upon differential methods. These optical flow methods work best, when displacements are within sub-pixel level.

The general procedure of frequency based image correspondence estimation is illustrated in Figure 4. After the image templates are transformed to the frequency domain, the phase imagery can be adjusted via different filters (`matching_tools_frequency_filters`). Similarly, the calculation of the cross-power spectrum can be done via different means (`matching_tools_frequency_correlators`). The resulting phase plane can be used to estimate the subpixel displacement (`matching_tools_frequency_subpixel`). While the quality of match can be deduced from the cross-power spectrum (`matching_tools_frequency_metrics`).

Image correspondence via the spatial domain is very similar, see also

[3] also known as, pattern matching, feature tracking, image velocimetry

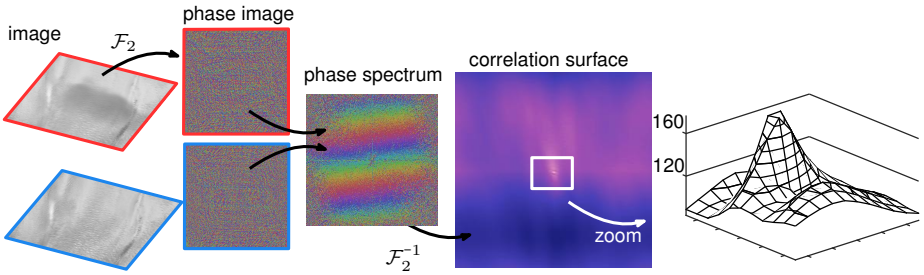

Figure 4: Illustration of finding image correspondence via the frequency domain.

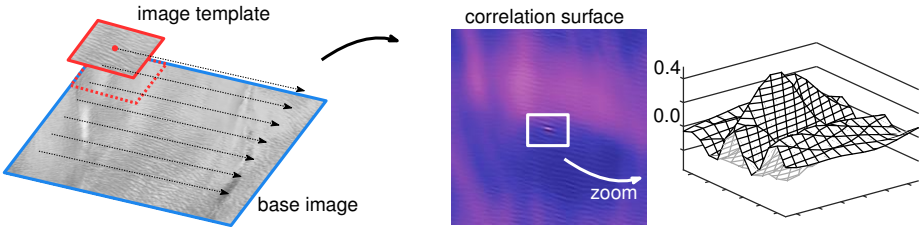

Figure 5: Illustration of finding image correspondence via the spatial domain.

Figure 5. Though here a smaller image subset is used and is slided over the other image subset. Again specific image operators can be used to put emphasis on certain image structures, but these functions are already present in the preprocessing folder. While different correspondence metrics can be found in `matching_tools_spatial_correlators`. This results in a two dimensional correlation function, where its peak can be localised by different methods (`matching_tools_spatial_subpixel`), as well as its metrics (`matching_tools_spatial_metrics`). Other descriptions of the matching quality are implemented in `geometric_precision_describtion`, where functions are situated that give indicators for the precision of an individual match.

While image correspondence is an essential component of a typical geodetic imaging pipeline, many functions are needed to support this building block. See for a schematic of such a pipeline also Figure 6. For example, functions in `matching_tools` and `coupling_tools` are generic functions to create for example a sampling setup. While `netwerk_tools` deals with large organization of imagery over time, similarly multi-temporal functions can be found in `matching_tools_geometric_temporal`. Lastly, most of the matching methods implemented have been extended to also cope with multi-spectral data such that stacking of cross-power correlation spectra (Altena and Leinss 2022) is possible.

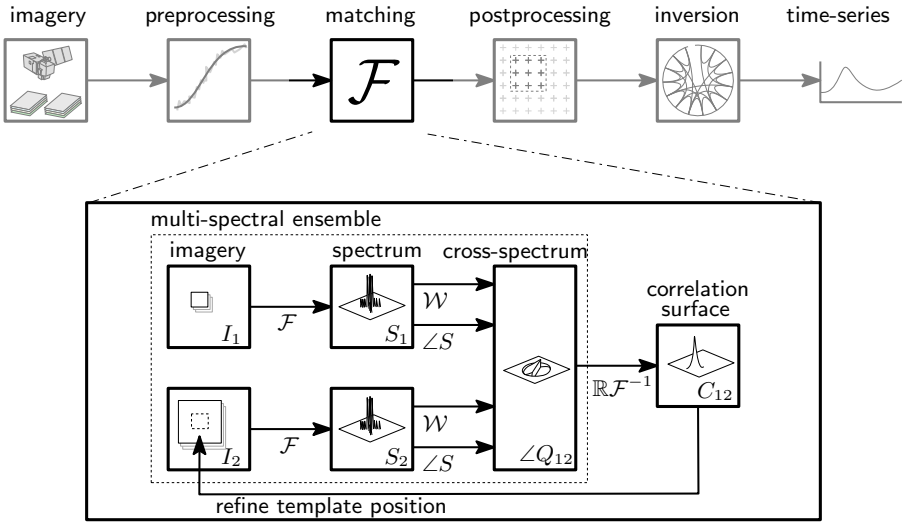

Figure 6: Schematic of the processing steps within a geodetic imaging pipeline, using a frequency based matching approach.

postprocessing

The functions within this folder are mostly concerned with product creation for specific application domains. It is composed of the following file structure:

```
postprocessing
├── mapping_io.py
├── displacement_filters.py
├── group_statistics.py
├── adjustment_geometric_temporal.py
├── photohypsometric_tools.py
├── atmospheric_tools.py
├── snow_tools.py
├── glacier_tools.py
├── terrain_tools.py
├── solar_tools.py
└── solar_surface.py
```

The products generated from image matching are typically noisy. The error distribution has elements of normally distributed noise, but a large part of the sample can also have outliers. Typically sampling of the neighbourhood is used to clean such data, and these function can be found in `displacement_filters` and more generic in `group\_statistics.py`. Multiple displacement and velocity products over time create redundancy and make it possible to apply inversion. This results in harmonised and evenly sampled data. Such functions can be found in `adjustment_geometric_temporal`.

The spatialtemporal coupling of can be found in `photohypsometric\_tools`

Further specific application domain functions, that relate to glaciology, hydrology, meteorology can be found in the other packages.

presentation

Functions within the presentation directory of the library are associated to specific data representation, typically with a spatial component. The file structure of this directory is as follows:

```
presentation
├── image_io.py
├── terrain_tools.py
├── displacement_tools.py
├── glacier_tools.py
└── velocity_tools.py
```

Generic functions are present in `image_io`. to create (geo-referenced) imagery of the (intermediate) results presented in the former directories. Since elevation data is needed for the photohypsometric pipeline, a suit of functions is present in the `terrain_tools`. Where mountain specific shading functions are incorporated. Other map making and data visualization functions specificly for surface kinematics can be found in `displacement_tools` and `glacier_tools`. Think of displacement vectors and their associated precision or strain rate maps. While fucntions within 'velocity_tools') are focussed on animations, of moving particles and flow paths.

generic

Many spatial functions are used in several . Especially terrain tools based on the elevation model used.

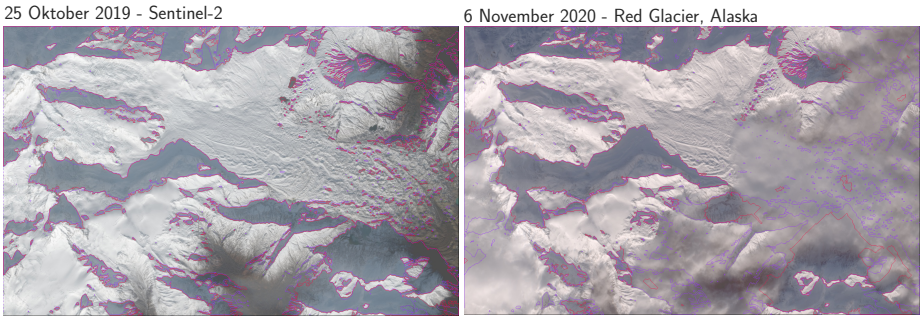

Figure 7: Example of shadow refinement (red) starting from an initial shadowing (purple) that is based on an elevation model.

auxilary

CopernicusDEM, Randolph Glacier Inventory, ERA-5

Localization of the glaciers is needed.

The XXX

For the estimation of the refraction, the state of the atmosphere needs to be known. Hence, atmospheric variables such as temperature and humidity are XX

```
auxilary
├── handler_coastal.py
└── handler_rgi.py
```

Functionality

Surface displacement estimation nice glacier velocity plot...?

Spatial temporal elevation change geodetic mass balance, maybe get some CryoSAT data?

Other software

Dissemination of satellite imagery is typically provided at different processing levels. The nomenclature is not standardised, but mostly level-0 data is raw sensor readings. Level-1 is georeferenced, while Level-2 is transformed to a physical measure. Lastly, Level-3 data are spatiotemporal consistent datasets. Since `dhdt` is mostly interested in the geometrical aspect, the lowest level is of most importance. Hence, for Sentinel-2 Level-1T is used. This is not the most common level, and repositories such as SentinelSat have routines for downloading Sentinel-2 Level-2 imagery.

A generic remote sensing library is Orfeo Toolbox (Grizonnet et al. 2017) this toolbox has a rich code base which has its emphasis on typical remote sensing methods, such as classification.

MicMac(Rupnik, Daakir, and Pierrot Deseilligny 2017) is another open-source software library, which stems from photogrammetry, where displacements are directly related to disparity, resulting in a topographic reconstruction. Though with some adjustments of the pipeline, it is also possible to extract surface displacements.

More glacier specific software can be found in Open Global Glacier Model (OGGM) (Maussion et al. 2019). While this initiative started out as a modular glacier flowline model, it now entails a large ecosystem with highly automated access to atmospheric and geospatial datasets. This modelling infrastructure has an approach based upon individual glaciers, which is also adopted in `dhdt`. Hence integration with this framework is foreseen in the future. Lastly, Quantum

Geographic Information System (QGIS) is a general system to combine and catalogue geographical datasets of all sorts.

Other datasets and repositories

Photogrammetry[4] is another technique that is able to get elevation products over time. However, an additional off-nadir telescope is needed to be able to create a systematic mapping configuration. It is also possible to stare towards a target region, but such systems are typically commercial and have limited coverage. Nonetheless, photogrammetry is able to create a large spatial coverage of a region. Similar to `dhdt`, the postprocessing procedures for glacier specific workflows are also captured by a repository such as xDEM. Since elevation models have specific systematic errors within (Hugonnet et al. 2022).

Apart from elevation products from optical remote sensing instruments, is it possible to estimate the geodetic mass balance from scattered laser altimetry data (Kääb 2008). Such systems are relatively limited in space, but their detailed footprint in the order of tens of meters and their repeated overflights make the generated products spatially sparse but consistent. Currently, ICESAT-2 is operational and tools for data processing can be found in the icepyx repository.

In the same realm one can use of microwave altimetry can be used for geodetic mass balance. Such altimeters have a footprint in the order of kilometers, thus are less useful for rough mountain terrain. Though interferometric altimeters, such as CryoSAT, are able to cope with complex mountain terrain (Gourmelen et al. 2018). Products from such satellites are ideal for flatter upper regions of glaciers and ice caps. Data from such satellites are currently available via the CryoTEMPO-EOLIS project.

Description of software

`dhdt` has other important dependencies, namely

Acknowledgements

We acknowledge contributions from Ou Ku, and Meiert Grootes, and support from Bert Wouters, Yifat Dzigan and Michiel van den Broeke during the genesis of this project. This project received funding from the Dutch research council (NWO) and the Netherlands Space Office (NSO) under their joint research progamme GO (grant agreement No. ALWGO.2018.044 - Eratosthenes).

[4] also known as, structure from motion, shape from motion

References

Abrams, Austin, Christopher Hawley, and Robert Pless. 2012. "Heliometric Stereo: Shape from Sun Position." *Lecture Notes in Computer Science* 7573: 357–70. https://doi.org/10.1007/978-3-642-33709-3_26.

Altena, Bas, and Silvan Leinss. 2022. "Improved Surface Displacement Estimation Through Stacking Cross-Correlation Spectra from Multi-Channel Imagery." *Science of Remote Sensing* 6: 100070. https://doi.org/10.1016/j.srs.2022.100070.

Barandun, Martina, Eric Pohl, Kathrin Naegeli, Robert McNabb, Matthias Huss, Etienne Berthier, Tomas Saks, and Martin Hoelzle. 2021. "Hot Spots of Glacier Mass Balance Variability in Central Asia." *Geophysical Research Letters* 48 (11): e2020GL092084. https://doi.org/10.1029/2020GL092084.

Daum, Michael, and Gregory Dudek. 1998. "On 3-d Surface Reconstruction Using Shape from Shadows." In *Proceedings. 1998 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 461–68. IEEE.

Davaze, Lucas, Antoine Rabatel, Yves Arnaud, Pascal Sirguey, Delphine Six, Anne Letreguilly, and Marie Dumont. 2018. "Monitoring Glacier Albedo as a Proxy to Derive Summer and Annual Surface Mass Balances from Optical Remote-Sensing Data." *The Cryosphere* 12 (1): 271–86. https://doi.org/10.5194/tc-12-271-2018.

Gourmelen, Noel, MariaJose Escorihuela, Andrew Shepherd, Luca Foresta, Alan Muir, Albert Garcia-Mondéjar, Monica Roca, S. G. Baker, and Mark R. Drinkwater. 2018. "CryoSat-2 Swath Interferometric Altimetry for Mapping Ice Elevation and Elevation Change." *Advances in Space Research* 62 (6): 1226–42. https://doi.org/10.1016/j.asr.2017.11.014.

Grizonnet, Manuel, Julian Michel, Victor Poughon, Jordi Inglada, Mickaël Savinaud, and Rémi Cresson. 2017. "Orfeo Toolbox: Open Source Processing of Remote Sensing Images." *Open Geospatial Data, Software and Standards* 2 (1): 1–8. https://doi.org/10.1186/s40965-017-0031-6.

Hugonnet, Romain, Fanny Brun, Etienne Berthier, Amaury Dehecq, Erik Schytt Mannerfelt, Nicolas Eckert, and Daniel Farinotti. 2022. "Uncertainty Analysis of Digital Elevation Models by Spatial Inference from Stable Terrain." *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing* 15: 6456–72. https://doi.org/10.1109/JSTARS.2022.3188922.

Hugonnet, Romain, Robert McNabb, Etienne Berthier, Brian Menounos, Christopher Nuth, Luc Girod, Daniel Farinotti, et al. 2021. "Accelerated Global Glacier Mass Loss in the Early Twenty-First Century." *Nature* 592 (7856): 726–31. https://doi.org/10.1038/s41586-021-03436-z.

Kääb, Andreas M. 2008. "Glacier Volume Changes Using ASTER Satellite Stereo and ICESat GLAS Laser Altimetry. A Test Study on Edgeøya, Eastern Svalbard." *IEEE Transactions on Geoscience and Remote Sensing* 46 (10): 2823–30. https://doi.org/10.1109/TGRS.2008.2000627.

Lenaerts, Jan TM, Brooke Medley, Michiel R van den Broeke, and Bert Wouters. 2019. "Observing and Modeling Ice Sheet Surface Mass Balance." *Reviews of Geophysics* 57 (2): 376–420. https://doi.org/10.1029/2018RG000622.

Maussion, Fabian, Anton Butenko, Nicolas Champollion, Matthias Dusch, Julia Eis, Kévin Fourteau, Philipp Gregor, et al. 2019. "The Open Global Glacier Model (OGGM) v.1." *Geoscientific Model Development* 12 (3): 909–31. https://doi.org/10.5194/gmd-12-909-2019.

Millan, Romain, Jérémie Mouginot, Antoine Rabatel, and Mathieu Morlighem. 2022. "Ice Velocity and Thickness of the World's Glaciers." *Nature Geoscience* 15 (2): 124–29. https://doi.org/10.1038/s41561-021-00885-z.

Rada Giacaman, Camilo Andrés. 2022. "High-Precision Measurement of Height Differences from Shadows in Non-Stereo Imagery: New Methodology and Accuracy Assessment." *Remote Sensing* 14 (7): 1702. https://doi.org/10.3390/rs14071702.

Rupnik, Ewelina, Mehdi Daakir, and Marc Pierrot Deseilligny. 2017. "MicMac – a Free, Open-Source Solution for Photogrammetry." *Open Geospatial Data, Software and Standards* 2 (1): 1–9. https://doi.org/10.1186/s40965-017-0027-2.

Tsai, Victor J. D. 2006. "A Comparative Study on Shadow Compensation of Color Aerial Images in Invariant Color Models." *IEEE Transactions on Geoscience and Remote Sensing* 44 (6): 1661–71. https://doi.org/10.1109/TGRS.2006.869980.

Zekollari, Harry, Matthias Huss, Daniel Farinotti, and Stef Lhermitte. 2022. "Ice-Dynamical Glacier Evolution Modeling — a Review." *Reviews of Geophysics* 60 (2): e2021RG000754. https://doi.org/https://doi.org/10.1029/2021RG000754.