Using the Google Earth Engine for Global Glacier Change Assessment

University of Zurich
Faculty of Science

Master's thesis
GEO 511

for obtaining the academic degree
Master of Science in Geography

presented by

Noah Zeltner
10752590

Supervised by	PD Dr. Michael Zemp Dr. Frank Paul
Faculty Member	Prof. Dr. Andreas Vieli

Date of submission: 31 October 2016

Summary

For an independent detection of climate trends, the monitoring of glaciers and their changes through time is important. Along with their hydrologic importance from regional to global scales they can also cause or trigger natural hazards, requiring a monitoring system that offers the possibility for frequent observations. Since satellite imagery covers large regions of the globe, it is possible to monitor glacier change at a large spatial and temporal resolution. Over the past years, these resolutions have even been pushed to new boundaries, resulting in a growing publicly available archive of satellite data. Despite the positive effects in regard to the density of information available for glacier monitoring this leads to an increase in manual workload and processing time on servers and computers. Using the Google Earth Engine (GEE) for detection, quantification and visualisation of glacier distribution displays a valuable tool to automatically deal with this increasing workload.

The objective of this thesis is to implement fully-automated algorithms for glacier monitoring within GEE which combines a multi-petabyte catalogue of satellite imagery and geospatial datasets with planetary-scale analysis capabilities.
In a first step, a scene selection algorithm has been created, which is able to select the most suitable images for glacier mapping in regard to a minimum of cloud cover and seasonal snow. Different approaches have been applied and compared to a manually performed scene selection in terms of quality of the scene selection and computation time. Although GEE has certain usage limitations which lead to restrictions in either spatial or temporal resolution and scale, the automated scene selection algorithm performs well enough to present a set of most suitable images for further processing and is applicable at large scales.
The second algorithm automatically computes glacier outlines and calculates the area based on an image composite of a prior scene selection. The creation of glacier outlines shows promising results and can be applied over a large spatial scale. Nevertheless, future applications should additionally deal with shadows within the individual scenes and the detection of debris-covered areas, as they both illustrate sources of error and reduce the quality of the results.
Third, the implementation of an automated thresholding function enables to classify snow and ice surfaces which provide good results even under the changing conditions of the resulting image composite. Assuming that the snow covered area roughly represents the accumulation area, it is tried to approximate the accumulation area ratio (AAR) and compare the trend of the two measures. Although it displays a tool which should be further refined by using reflectance values that are corrected for topographic effects, it offers the possibility to compute a fair estimate of values over large regions.
Furthermore, the objective to complement already available glacier inventories with a mask

of debris-covered regions was achieved by creating an image composite for the entire regions of West-Central-Asia and the Swiss Alps respectively. Despite the dependency on the quality of the prior scene selection the results provide promising insights for a better understanding of ablation processes and their implications within a specific region.

Contents

List of Figures

List of Tables

1 Introduction

1.1 Monitoring of global glacier change and -distribution

Monitoring of glaciers and their changes through time is important for an independent detection of climate trends, as well as for their hydrologic importance from regional (run-off, hydro-power) to global (sea-level rise) scales (IPCC 2013). They can also cause or trigger natural hazards (ice avalanches, glacier lake outburst floods) requiring frequent observation of their changes (Paul 2004). As glaciers are distributed globally and the huge and growing archive of Landsat data is commonly used for the above purposes (but at regional scales), using the Google Earth Engine (GEE) for detection, quantification and visualisation of worldwide glacier distribution would be a most valuable asset.

1.2 Growing importance of satellite imagery

Applying glacier monitoring on a global scale is possible since satellite imagery covers large regions of the globe. Before that, the application on a global level could not be pursued due to the high costs of accessing images and keeping the acquired datasets up to date on a regular basis. Although, in-situ measurements can't be spared to gain information about the processes on, in and around glaciers, satellite imagery offers a tremendous opportunity to monitor glacier change at a larger spatial and temporal scale. Over the past decades more and more satellite missions have been launched with the aim to survey and monitor the Earth's surface and near-surface systems. Thereby, spatial and temporal resolution are pushed forward to new boundaries. Especially the (public) availability of Landsat Thematic Mapper (TM) since 1984 (U.S. Geological Survey 2008) and the launch of the Sentinel satellite mission by the European Space Agency (ESA) within the Copernicus programme in 2014 both feature a promising outlook and represent important milestones for monitoring services based on satellite imagery. Until now, a single Landsat satellite crosses every point on earth once every 16 days. This revisit frequency is set to a completely new level by the Sentinel satellite mission (Sentinel-2A and Sentinel-2B) which record images of the same location every 5 days with a three times higher spatial resolution of 10 m. This also illustrates the massive amount of data which will be available for processing in future. On one hand, increasing spatial and temporal resolution have a positive effect regarding density of information available for glacier monitoring. On the other hand, it leads to an increase in manual workload and processing time on servers and computers. As in many other disciplines there is no way around automation processes as the amount of data is increasing and the datasets themselves are getting larger and richer in detail. In combination with the availability of accurate and robust mapping methods for clean to slightly dirty glacier ice (Albert 2002, Paul & Kääb 2005) automated processes would free the way to a more or less complete global monitoring of glacierized areas.

1.3 Process of selecting suitable scenes for glacier mapping applications

Particularly the automation of the process of selecting suitable scenes for glacier mapping will be fundamental for future global monitoring of glaciers. Until today there have only been a few attempts to automate glacier mapping processes which rely on multi-temporal image processing and use Landsat-type sensors (Selkowitz & Forster 2016, Winsvold et al. 2016). Cloud cover, seasonal snow cover and shadows are the most deciding factors whether a satellite scene is suitable or not for glacier mapping applications. The laborious work of glacier analysts to manually browse through the constantly growing, already voluminous archive of satellite images to globally map glaciers, is already unsustainable now and will be unbearable in the near future if all available data should be exploited in regard to the different applications in environmental studies and glaciology in particular. The aim here is to make use of the computation power and direct data access within GEE and create a fully automated work flow of selecting the most suitable scene(s) for glacier mapping applications.

1.4 Contribution to monitoring applications of glacierized regions

Providing scientifically sound and concise information is a critical function in today's world as glacier changes are among the most prominent indicators of climate change (WGMS 2008). The creation of glacier outlines and the possibility to compute the corresponding area of glacierized terrain is thus an important task for glacier monitoring. With the automated scene selection process described in Section 1.3 serving as starting point, the fully automated creation of glacier outlines and computation of their area would contribute substantially to glacier monitoring. As a result, a global assessment at a high temporal and spatial resolution could be pursued with considerably reduced effort.

Apart from scene selection, the creation of glacier outlines and the subsequent area calculation, there are various additional applications which can profit from the possibilities of automated processing helping to provide the improved access to significant environmental information and data. This includes information on the Accumulation Area Ratio (AAR) of a specific glacier (approximated by mapping the snow covered area), or complementing the existing glacier inventories with extents of the debris-covered parts of a glacier.

1.5 Research objectives

Implementation of the already available methods for data processing in GEE would allow going truly global, thereby overcoming current restrictions due to the workload related to image selection and scene-by-scene processing. A modular extension with already existing vector datasets would allow to provide information about on-going glacier changes at an unprecedented frequency and spatial coverage.

In this thesis the main objective is to elaborate on algorithms to automate the process from image selection and scene-by-scene processing (named as Algorithm 1) to glacier classification and area calculation to create glacier outlines (Algorithm 2). The objectives here are to review whether it is possible to access necessary satellite data, compute needed measures and hence automatically evaluate the quality of individual satellite scenes or not. The aim also is to find the best or a set of most suitable scenes for glacier mapping to create precise glacier outlines for a given time period.

In a second step, the focus will be on implementing an automated method to map snow- and ice cover on glaciers (Algorithm 3) to approximate the Accumulation Area Ratio (AAR). Being able to calculate time-series of such AAR approximations could serve as a meaningful instrument in glacier monitoring, especially if used on remote glaciers with a low density of measurements throughout the past (Barandun et al. 2015). In this part of the thesis the objectives are to assess the potential to retrieve a reasonable approximation of the AAR from the mapped snow covered area within GEE based on a preceding automatic scene selection for a specified region. A special focus is on the implementation of the Otsu method to automatically find the critical threshold to distinguish ice and snow (Ostu 1979).

In a last step of the thesis, the goal is to combine the above designed methods within GEE to separate clean ice from debris-covered ice (Algorithm 4). Already available glacier inventories could then be provided with a mask of the debris-covered parts for each glacier.

1.6 Structure of the thesis

This thesis constitutes several chapters following a common thread. First, in Chapter 2, the scientific background to the essential methods in glaciology and the data used within this thesis are given and datasets are described. In Chapter 3, the used methods and algorithms which have been developed to automatically select suitable scenes (Algorithm 1), map the extent and calculate the area of glaciers (Algorithm 2), map snow- and ice cover to approximate the accumulation area ratio (AAR) (Algorithm 3) and separate clean- and debris-covered ice (Algorithm 4) are illustrated. In Chapter 4, the output and overall results of the individual methods are shown and initial interpretations are illustrated. In Chapter 5, the results are discussed and set in relation to the current literature in these fields. In a last step some general conclusions are drawn in Chapter 6 with the perspectives and the insights which have been gained.

2 Scientific background and data

In this Chapter the scientific background is given and the data for the computations are described. The basic functionality of GEE is described as most of the calculations are done within the GEE itself striving for automated approaches to achieve the research objectives. This Chapter serves as foundation for the methodical Chapter 3.

2.1 Technical background on glacier monitoring

2.1.1 Distribution of glaciers

A total of 96.6% of the Earth's glacierized area is represented by the large ice sheets of Antarctica (85.7%) and Greenland (10.9%). Within the remaining 3.4% only 1.3% concern mountain glaciers while the rest are covered by ice caps at higher latitudes (WGMS 1989). Although mountain glaciers are only a small fraction of the overall glacierized area, they represent the perennial surface ice that is most closely connected to human activities which is why their monitoring is of such an importance in regard to serving as a climate indicator, as water resource or for reasons of tourism.

2.1.2 Glacier inventories

To a greater understanding of our climate system, climate change and the effects of global warming, one needs to understand how and why glaciers are important. Hence, glacier inventories are an indispensable source of historical data and base on in-situ mass balance measurements, as well as air- and spaceborne imaging systems (Kargel et al. 2014).
Here, the latest Randolph Glacier Inventory (RGI) released July 20, 2015 (Version 5.0) has been used as database for glacier polygon imports into the GEE. The RGI is a publicly available product provided by the Global Land Ice Measurements from Space (GLIMS, `http://glims.org/RGI/rgi50_dl.html`) and was motivated by the Intergovernmental Panel on Climate Change (IPCC 2013) and designed to mostly get a globally complete snapshot of glacier coverage from optical satellite images. Methods have primarily been tested with the following regions of the RGI (Pfeffer et al. 2014):

- 06 Iceland
- 08 Scandinavia
- 11 Central Europe
- 13 Central Asia
- 14 South Asia West

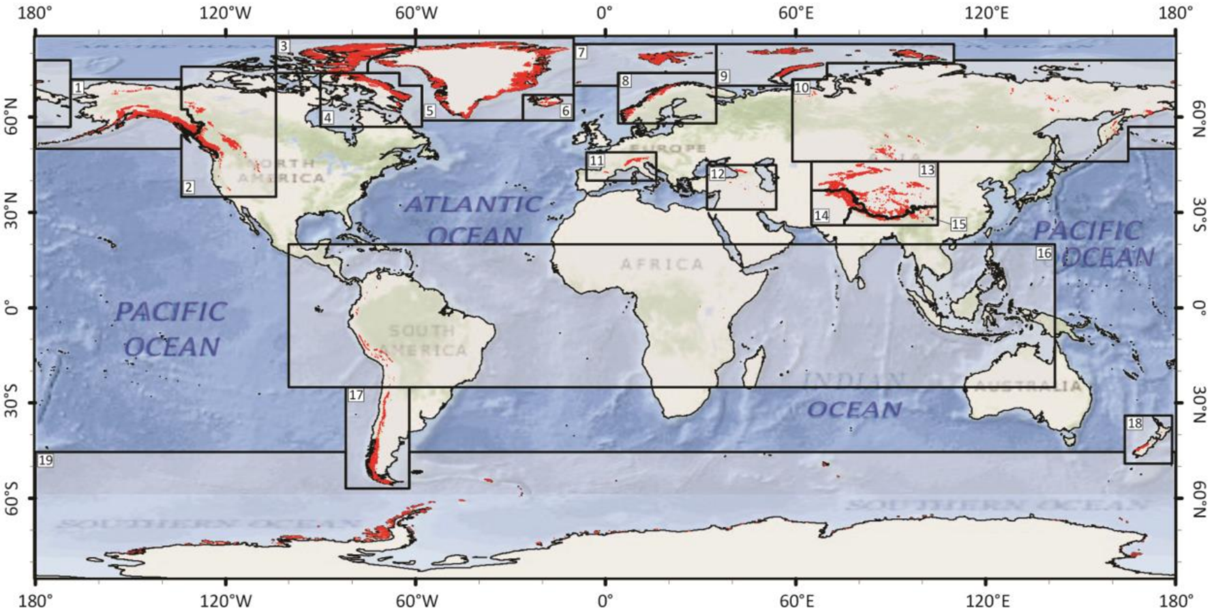


Figure 2.1: First-order regions by the RGI (Randolph Glacier Inventory 2015).)

2.1.3 Satellite imagery

The focus lies on optical rather than on microwave sensors and especially on Landsat TM/ETM+ type sensors which are most frequently used for glacier mapping (Paul, 2014). All algorithms described in Chapter 3 have been developed using Landsat TM/ETM+ type sensor data but work as well with the Operational Land Imager (OLI) sensor from Landsat 8 or Sentinel 2 Multispectral Image (MSI) sensor imagery (Paul et al. 2016). Other sensor types have not been tested but will work similarly if the sensor has individual bands recording in similar spectral ranges.

2.1.4 Scene selection

To achieve the best satellite scene selection, at least the following three prerequisites have to be given (Paul 2004):

1. cloud-free conditions
2. near the end of the ablation period (e.g. mid-August to mid-October)
3. minimum extent of seasonal snowfields

Prerequisite (1) can be evaluated using a cloud-score algorithm which sorts pixels by their relative cloudiness including various indicators of a pixel to be cloudy. By filtering the dates using built-in functions prerequisite (2) is considered. In Section 3.3.1 more technical information is given. However, the recognition of seasonal snow in (3) has proven to be more challenging and is presented in more detail in Section 3.2.

Finding a suitable scene can be very challenging not only by trying to implement the above prerequisites but also because most glacierized areas around the globe are rarely recorded under decent conditions. Although one can find the most suitable scenes in a certain region, they might not be good enough to be taken as a basis for further analysis and calculations. In particular in maritime regions such as Svalbard, parts of Scandinavia and Greenland it is difficult to find suitable scenes for glacier mapping even within a few years.

2.1.5 Classification of glacierized area

Segmentation of ratio images

In terms of the classification of snow and ice different methods have been investigated in previous studies. One of the most promising methods still is the segmentation of ratio images which already has been used extensively until today (Carturan et al. 2013, Gardent et al. 2014). Especially the one of near infrared and short-wave infrared (corresponds to e.g. Landsat TM4 and TM5) but also ratio images based the band covering red light and short-wave infrared (corresponds to e.g. TM3 and TM5) have been implemented successfully as the latter seem to better perform in zones of shadow but is mapping more of the turbid lakes as water (Rott & Markl 1989, Paul & Kääb 2005). For the segmentation of ratio images visual interpretations of ratio images revealed that a threshold around 2 works well in most cases to separate glacierized areas from non-glacierized regions (see Section 4.2.1).

Classification of clean- and debris-covered ice

Similar to other classification work based on segmentation of ratio images, the detection of debris-covered ice poses a difficult task (Racoviteanu et al. 2009, Paul et al. 2009). Here, debris-covered ice is defined as the part of the glacier which is covered by a critical amount of dirt and rock which is not recognized by the classifier any more. However, if the portion of debris-cover within one pixel exceeds a certain threshold the pixel will not be classified as glacier (Paul et al. 2004). In contrast, clean ice often includes some dirty ice (sand, gravel, small stones, single boulders) but is still recognized as clean ice and thus classified as glacier. A visual interpretation of a suitable scene by an experienced viewer can correctly classify such glacier parts.

Sources of error

In addition to misclassification of debris-covered glaciers there are other sources of error. As mentioned in Section 2.1.4, occurrence of seasonal snow can pose a problem as the eventual classification of glacierized area is supposed to be larger than it should be. Water bodies such as the ocean, lakes or rivers are also partly classified as glaciers and need to be recognized and removed by either visual interpretation or additional intersection with respective datasets or classifiers in a second step. Evaluating the result of the classification in the other direction is the case for shadow being cast on parts of the glacier which usually occurs near steep mountainsides (Paul 2000).

2.1.6 Mapping snow on glaciers to approximate the accumulation area ratio

Glacier mass balance is a response to annual atmospheric conditions and provides essential estimates for the contribution of glaciers to sea level (Kaser et al. 2006). The mass balance is highly dependent on precipitation and air temperatures throughout the year. A popular proxy to estimate the mass balance is the determination of the equilibrium line of altitude (ELA) which is marking the position where ablation is balanced by accumulation in the respective year. Similarly, the Accumulation Area Ratio (AAR), which defines the ratio of the accumulation area to the total area (Dyurgerov et al. 2009), is also a good indicator for the mass balance of a glacier. As the snow covered area at the end of the ablation period approximates the region with mass gain (accumulation area) it was decided to map the snow covered area over several years within GEE as described in Section 3.4.

2.1.7 Automation processes

A high quality inventory of glaciers and ice caps from both aerial photographs and satellite images still needs to be created by a well-trained glaciologist. Having experience and expert knowledge in creating a glacier inventory with all glacier parameters from satellite imagery is crucial for its quality. Additionally, every satellite scene should be analysed individually as atmospheric conditions are changing from scene to scene and even within a single scene. This knowledge is still applicable when working with semi-automated methods as has been done in other studies before (Selkowitz & Forster 2016, Robson et al. 2016, Winsvold et al. 2016).

Automated approaches often lack the required flexibility in such context. It is thus a goal to build algorithms in a way that the computer can work with satellite data like an expert would do. This includes taking decisions on the further processing based on interim results. Still, achieving the same level of flexibility and reliability is a difficult task. Inaccuracies which need to be limited are dependent on the spatial and temporal resolution of the available scenes which might not meet the requirements of a suitable scene in each particular application. Furthermore, the general assumptions made to simplify the algorithm are sources of error.

Nevertheless, working towards automation has a lot of advantages, despite the fact that it is difficult to automate a complex process of selecting the most suitable scene(s) in all the different locations of the world with distinct atmospheric conditions and constantly varying conditions in regard to cloud-, seasonal snow- and shadows. As described in Section 1.2 the increase of workload and processing time on servers and computers is calling for new alternatives on handling the shear endless and constantly increasing data flow. With automation processes the true power of computational processing can be played off and create new possibilities regarding spatial coverage of calculations. Automation processes are carried out within the GEE which handles all data inquiries on its own servers (cloud computing) which leads to a decrease in processing time on the user's computer which is only prompted the final results of the data inquiry. Consequently, it should be possible to retrieve glacier outlines on at large scales or to monitor glacier change over several years.

2.2 The Google Earth Engine (GEE)

The Google Earth Engine is a platform combining a multi-petabyte catalogue of satellite imagery and geospatial datasets with planetary-scale analysis capabilities and is available for everyone to detect changes, map trends, and quantify differences on the Earth's surface (Google Earth Engine Team n.d.). It is thus a tool for analyzing geospatial information with an application programming interface (API) which is available in Python and JavaScript. Here, the methods are built on using JavaScript.

2.2.1 Background

As stated above, it would be truly valuable for environmental change studies to automatically map glaciers on a global level. In theory GEE could give the opportunity to handle such an amount of data and calculate multi temporal changes of glacier extents within two points of time. To reach this point though, it is important to outline the key functionalities that have to be achieved within GEE and check whether they are existing or if they can be obtained with additional work.

Terminology of data types

The following terms and expressions are used here as they exist within the GEE (Google Earth Engine Team n.d.). The special notation indicates that one deals with the respective data type:

`Image`
: `Images` are composed of one or more bands and each band has its own name, data type, pixel resolution and projection. In addition, each `Image` has metadata stored as a set of properties.

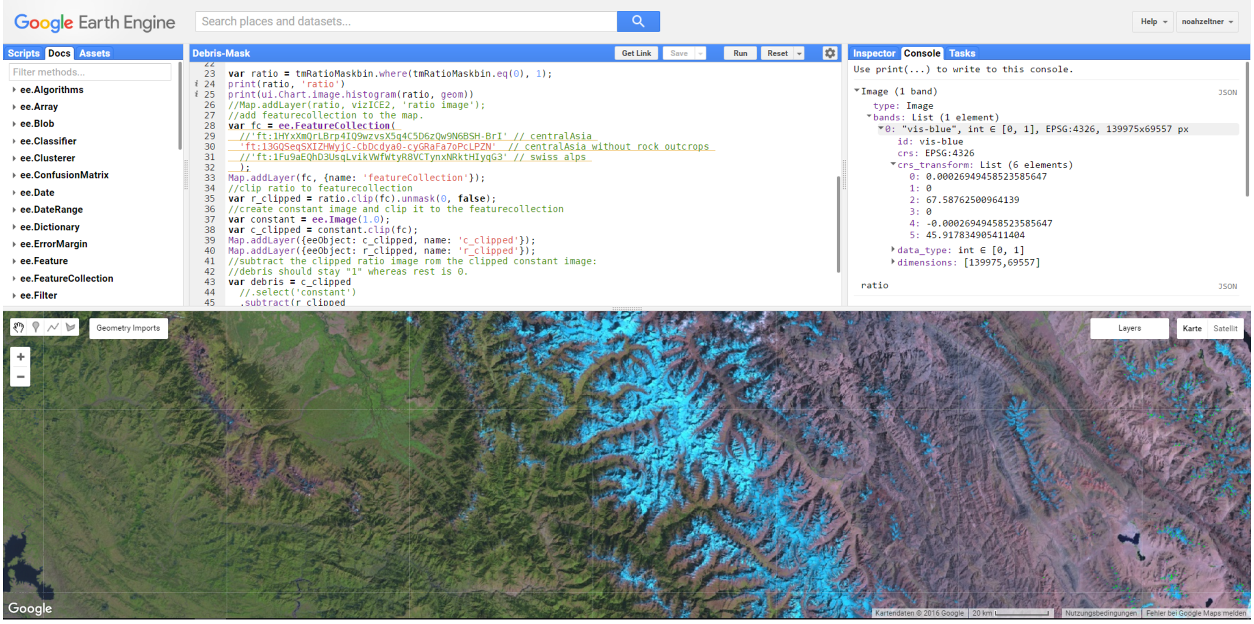
`ImageCollection`
: An `ImageCollection` is a stack or time series of Images.

`Feature`
: A `Feature` is defined as a GeoJSON feature and includes geometry properties which can be added to a map in the GEE or outside of it when exported.

`FeatureCollection`
: A `FeatureCollection` is a collection of `Features`. Imported *.shp*-files are saved as `FeatureCollection` within GEE containing single glacier polygons as single `Features`.

Data availability

The GEE data catalogue shows a variety of different datasets. The user has access to images from different satellite images providers such as MODIS, LANDSAT, SENTINEL or digital elevation data from the Shuttle Radar Topography Mission (SRTM). In addition to that,



```
var ratio = tmRatioMaskbin.where(tmRatioMaskbin.eq(0), 1);
print(ratio, 'ratio')
print(ui.Chart.image.histogram(ratio, geom))
//Map.addLayer(ratio, vizICE2, 'ratio image');
//add featurecollection to the map.
var fc = ee.FeatureCollection(
  //'ft:1HYxXmQrLBrp4IQ9wzvsX5q4C5D6zQw9N6BSH-BrI' // centralAsia
  'ft:13GQSeqSXIZHWyjC-CbDcdya0-cyGRaFa7oPcLPZN'  // centralAsia without rock outcrops
  //'ft:1Fu9aEQhD3UsqLvikVWfWtyR8VCTynxNRktHIyqG3' // swiss alps
  );
Map.addLayer(fc, {name: 'featureCollection'});
//clip ratio to featurecollection
var r_clipped = ratio.clip(fc).unmask(0, false);
//create constant image and clip it to the featurecollection
var constant = ee.Image(1.0);
var c_clipped = constant.clip(fc);
Map.addLayer({eeObject: c_clipped, name: 'c_clipped'});
Map.addLayer({eeObject: r_clipped, name: 'r_clipped'});
//subtract the clipped ratio image rom the clipped constant image:
//debris should stay "1" whereas rest is 0.
var debris = c_clipped
  //.select('constant')
  .subtract(r_clipped
```

```
Use print(...) to write to this console.

▾Image (1 band)                                              JSON
   type: Image
  ▾bands: List (1 element)
    ▾0: "vis-blue", int ∈ [0, 1], EPSG:4326, 139975x69557 px
       id: vis-blue
       crs: EPSG:4326
      ▾crs_transform: List (6 elements)
         0: 0.00026949458523585647
         1: 0
         2: 67.58762500964139
         3: 0
         4: -0.00026949458523585647
         5: 45.917834905411404
      ▸data_type: int ∈ [0, 1]
      ▸dimensions: [139975,69557]

 ratio                                                       JSON
```

Figure 2.2: Earth Engine's application programming interface (API) with the documentation tab, the code editor, the output tab and the direct visual feedback on the map (from top left corner, clockwise)

the GEE data catalog includes satellite data in raw format, orthorectified imagery, top-of-atmosphere- and surface reflectance data.
The GEE data catalog serves as a basis for most of the calculation within the GEE environment as one's codes work best when staying inside this environment. Importing additional data is possible but often limits the use of built-in functions especially designed to work with data from the GEE data catalog. In these cases suitable workarounds have been created.

Limits and restrictions

During the building of the entire algorithms and fragments of them testing has been crucial to see whether the queries are hitting the usage limit on Google's servers. As a first restriction, Google is only allowing 3 queries per second which helps to ensure that resource-intensive algorithms do not affect the overall availability of GEE's services. In addition to that, client-side computations time-out after three minutes, whereas exports get the same error message only after a longer time: they hit the precompute limits after two hours of computation time or the *on-the-fly limits* after 10 minutes per `Feature`.

Depending on the size of the initial `ImageCollection` and the imported `FeatureCollection` these limits can be reached. There are ways to bypass these limits by adapting ones data and parameters. Particularly the scale parameter of regional counts of pixels instantly reduces the computation time but unfortunately at an expense of accuracy and detail. However, if ones algorithms provide extraordinary and promising results, there is the possibility to ask for additional computation power, although this has only been granted for less than a handful of projects such as for "Global Forest Watch". Depending on the further applications it could be worth to apply for such an increase in data usage.

2.2.2 Methodical feasibility: basic functionalities

Access satellite raw data

Spectral band info can be retrieved by calling respective band names on a specific image or image collection within GEE. A specific band normally contains DN values or spectral radiance per pixel.

Subsetting

GEE contains methods to filter a collection or list by date ranges, metadata (e.g. path and row information) or based on boolean operators. Items are excluded from further calculations using filters to create subsettings. Filters are crucial methods to use to reduce the amount of data in a pre-processing step to downsize the overall processing.

Masking

Masks make it possible to hide specific values or value ranges. It is often used to set particular values to the value of zero thus acting as a sort of filter.

Cloud detection

For scoring pixels by their relative cloudiness, Earth Engine provides a cloud-scoring algorithm and an additional method to create a composited image out of single pixels chosen by their relative cloud-score. This cloud-score has been slightly modified to the conditions in high mountain areas.

Ratio images

By using simple mathematical operations ratio images are calculated and saved as new images.

Pixel counting

A simple pixel counting method is implemented and enables the area calculation for glacierized area within a certain region in combination with internal methods of GEE.

2.2.3 Import of external datasets

To import vector datasets such as the one from the RGI Google Fusion Tables need to be created. The recommended tool to convert .shp-files into Google Fusion Tables is *Shpescape* (Livni 2016) which directly uploads the resulting file into one's Google Drive folder ready to be queried by the GEE.

2.2.4 Export and further calculations

Exports are again saved to one's Google Drive where they can be viewed or downloaded for further use. Only Images (multi-band GeoTIFF), tables (.csv-file) and FeatureCollections (.csv-, .kml- and GeoJSON-files) can be exported.

2.3 Study regions and data of comparison

As stated before, the aim would be to go truly global, by overcoming current restrictions due to the high workload of scene-by-scene processing. Therefore, at least some of the results are related to a larger scale. To be able to test the algorithms, a few different test regions have been selected where data from other studies and investigations are available for comparison. The testing for Algorithm 1 (automated scene selection) has been done with a subset of the Swiss Alps, mainly in the region of the Bernese- and Valais Alps, whereas Algorithm 1.2 has been tested for Central Europe. Algorithm 2 (creating glacier outlines) was examined within Central Europe and Scandinavia. The results were compared to data from the Swiss Glacier Inventory from 2003 (Paul 2004) based on satellite data and the one from 2010 (Fischer et al. 2014) derived by manual digitization from high-resolution (25 cm) aerial orthophotographs. These datasets were retrieved from the data catalogue of the Randolph Glacier Inventory (`http://www.glims.org/RGI/rgi50_dl.html`). Algorithm 3 has been tested for the glaciers *Findelen* (Switzerland), *Kesselwand* (Austria) and *Storglaciaeren* (Sweden). The testing of Algorithm 4 (separation of clean- and debris-covered ice) has been done within the Swiss Alps, with a focus on the Bernese- and Valais Alps, and for Central- and West Asia focusing on the Karakoram mountains.

3 Methods

3.1 General remarks

The various scripts consist of different functions which have been built to match the purpose of this study. In the following sections the main functions are explained, keeping it on a conceptual level. Advanced GEE users and curious readers can examine the specific code in the appendix or within the GEE itself by using the provided links to the specific algorithms in Chapter 7.
As a methodical “road map” the approach of the glacier classification and mapping of (Paul 2004) used in the Swiss Glacier Inventory (SGI) is pursued. The code is based on RGI vector data retrieved directly on the website of the RGI containing glacier polygons on a global scale. The tool *shpescape* (Livni 2016) was used to convert *.shp* file format into the format of a `FeatureCollection` used in GEE. To simplify coding and lower the time of processing various RGI regions were fragmented into smaller units or even single glacier polygons. All RGI regions and fragments imported as mentioned above are publicly available within GEE.
This chapter explains the different algorithms, divided in four parts: First, the scene selection which serves as basis to identify the suitable scenes according to the prerequisites explained in Section 2.1.4. It is divided into Algorithms 1.1 and 1.2 representing two different approaches of scene selection (single scene selection and multi scene selection. Second, the calculation of a glacierized area is done by Algorithms 2.1 (automatic approach) and 2.2 (semi-automatic approach). Third, Algorithm 3 is mapping snow- and ice cover and trying to approximate the AAR of a glacier and covers the implementation of the Otsu method in detail. Algorithm 4 deals with the separation process of clean- and debris-covered ice and shows the possibility to retrieve a debris-cover mask for existing inventories.

3.2 Scene selection

3.2.1 Algorithm 1: Automated scene selection

Two approaches of automated scene selection have been pursued: Algorithm 1.1 handles the selection of the most suitable scenes for a specific path and row, whereas Algorithm 1.2 handles the selection of the most suitable scenes for a set of glaciers over a wider region. The latter one takes a `FeatureCollection` (a set of geometries imported to GEE) instead of path row information. These two approaches needed two separate algorithms and are conceptually described in Figure 3.1 and 3.2 respectively.

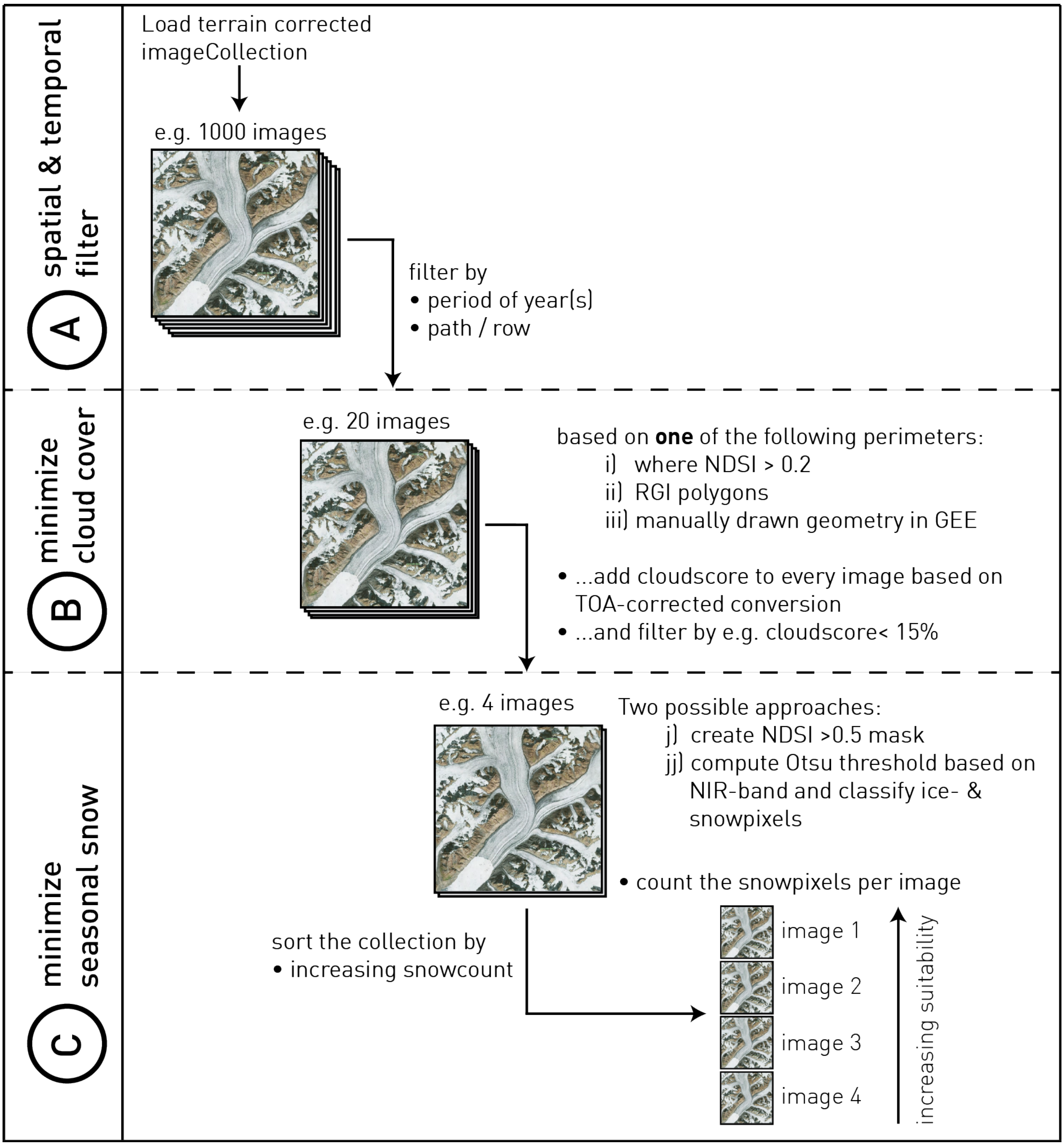

A spatial & temporal filter
Load terrain corrected imageCollection
e.g. 1000 images
filter by
• period of year(s)
• path / row
B minimize cloud cover
e.g. 20 images
based on **one** of the following perimeters:
i) where NDSI > 0.2
ii) RGI polygons
iii) manually drawn geometry in GEE
• ...add cloudscore to every image based on TOA-corrected conversion
• ...and filter by e.g. cloudscore< 15%
C minimize seasonal snow
e.g. 4 images
Two possible approaches:
j) create NDSI >0.5 mask
jj) compute Otsu threshold based on NIR-band and classify ice- & snowpixels
• count the snowpixels per image
sort the collection by
• increasing snowcount
image 1
image 2
image 3
image 4
increasing suitability

Figure 3.1: Concept of Algorithm 1.1 with the three sequential steps: (A) selection of the end of ablation period, (B) minimization of cloud cover and (C) minimization of seasonal snow cover. For (B) (and subsequently for (C) too) the three different approaches define the perimeter of analysis: (i) a snow cover mask based on the Normalized Difference Snow Index (NDSI) is generated, (ii) the imported RGI polygons are taken as reference area or (iii) a specifically sketched geometry is used. Within (C) two possible approaches can be used: (j) uses a NDSI>0.5 mask whereas (jj) computes the Otsu threshold based on the near infrared (NIR) band to classify and quantify snow cover.

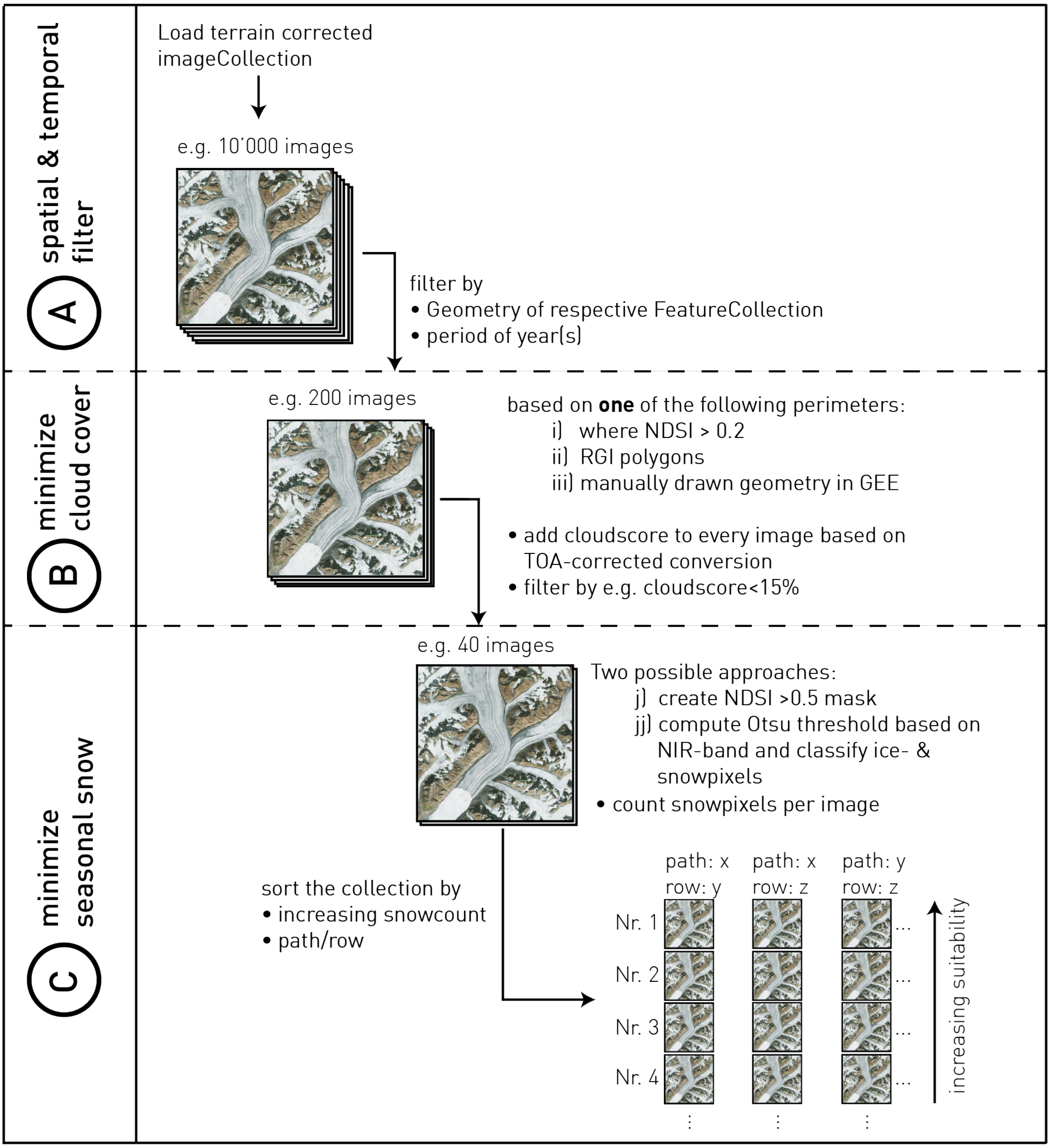

A spatial & temporal filter
Load terrain corrected imageCollection
e.g. 10'000 images
filter by
• Geometry of respective FeatureCollection
• period of year(s)
B minimize cloud cover
e.g. 200 images
based on **one** of the following perimeters:
i) where NDSI > 0.2
ii) RGI polygons
iii) manually drawn geometry in GEE
• add cloudscore to every image based on TOA-corrected conversion
• filter by e.g. cloudscore<15%
C minimize seasonal snow
e.g. 40 images
Two possible approaches:
j) create NDSI >0.5 mask
jj) compute Otsu threshold based on NIR-band and classify ice- & snowpixels
• count snowpixels per image
sort the collection by
• increasing snowcount
• path/row
path: x row: y
path: x row: z
path: y row: z
Nr. 1
Nr. 2
Nr. 3
Nr. 4
increasing suitability

Figure 3.2: Concept of Algorithm 1.2 with the three sequential steps: selection of the end of ablation period (A), minimization of cloud cover (B) and minimization of seasonal snow cover (C). For (B) (and subsequently for (C) too) there are three different approaches to define the perimeter of analysis: (i) a snowcover mask based on the Normalized Difference Snow Index (NDSI) is generated, (ii) the imported RGI polygons are taken as reference area or (iii) a specifically sketched geometry is used. Within (C) two possible approaches can be used: (j) uses an NDSI>0.5 mask whereas (jj) computes the Otsu threshold based on the near infrared (NIR) band to classify and quantify snow cover.

User-defined variables

Several user-defined variables are required to successfully run Algorithm 1. A selection of the important ones is given in Table 3.1:
Changing these variables on one hand, offers a quick and easy handling of otherwise hidden variables. On the other hand, it offers some flexibility to adapt to error messages. The latter ones can occur when the time horizon is set to a wide time range, causing time-out messages when exporting the results. The two scale parameters then need to be increased accordingly.

Spatial and temporal filter of ImageCollections

To reduce the amount of data to be analysed a spatial filter is applied on the terrain corrected `ImageCollection` which has been loaded in advance. In part (A) of this algorithm this filter is based on the specific path and row information for Algorithm 1.1 (Figure 3.1). Whereas, in Algorithm 1.2 (Figure 3.2) the `ImageCollection` is filtered by the imported `FeatureCollection`.
In a second step the `ImageCollection` is filtered by a date range which should correspond to the end of ablation time in the region of analysis. For example, in the test region of the Swiss Alps the date range has been specified from 15th July to 15th October.

Minimal cloud cover

In part (B) the images are checked on their cloudiness. This cloud-score is calculated based on top-of-atmosphere (TOA) corrected conversions of the respective image. In many cases, satellite scenes already come with some pre-calculated factor for cloudiness such as the "Automatic Cloud Cover Assessment" (ACCA) for Landsat 7 images (Irish 2000). The ACCA, however, does not perform well over snow and ice due to the cold surfaces. This is why another approach has been pursued to better match the specific application within this environment of cold surfaces.
To reduce the perimeter where computations of the minimization of cloud cover (and subsequently also of seasonal snow) are based on, three different approaches (see Figures 3.1 & 3.2) are compared: this spatial limitation can either be done (i) by calculating the normalized-difference snow index (NDSI) for the entire image, (ii) by clipping the images to the defined RGI polygon or (iii) by clipping them to a manually drawn polygon.
Primarily, this is done to prevent the algorithm from falsely discarding images without clouds over the glacierized area, but still achieving a high cloud-score. Hence, a mask is generated in advance to focus the computations on where glaciers actually exists. In addition to that, it also lowers the computation time.
In regard to computation time, the approach using the NDSI mask (i) is faster but has the disadvantage of sometimes underestimating the seasonal snow cover in the next step due to clouds covering the glacierized area resulting in a small NDSI-mask. The approach using the RGI polygons (ii), however, takes a lot more computation time as the polygons can be quite complicated (large amount of vertices). If a specific region of interest exists and can

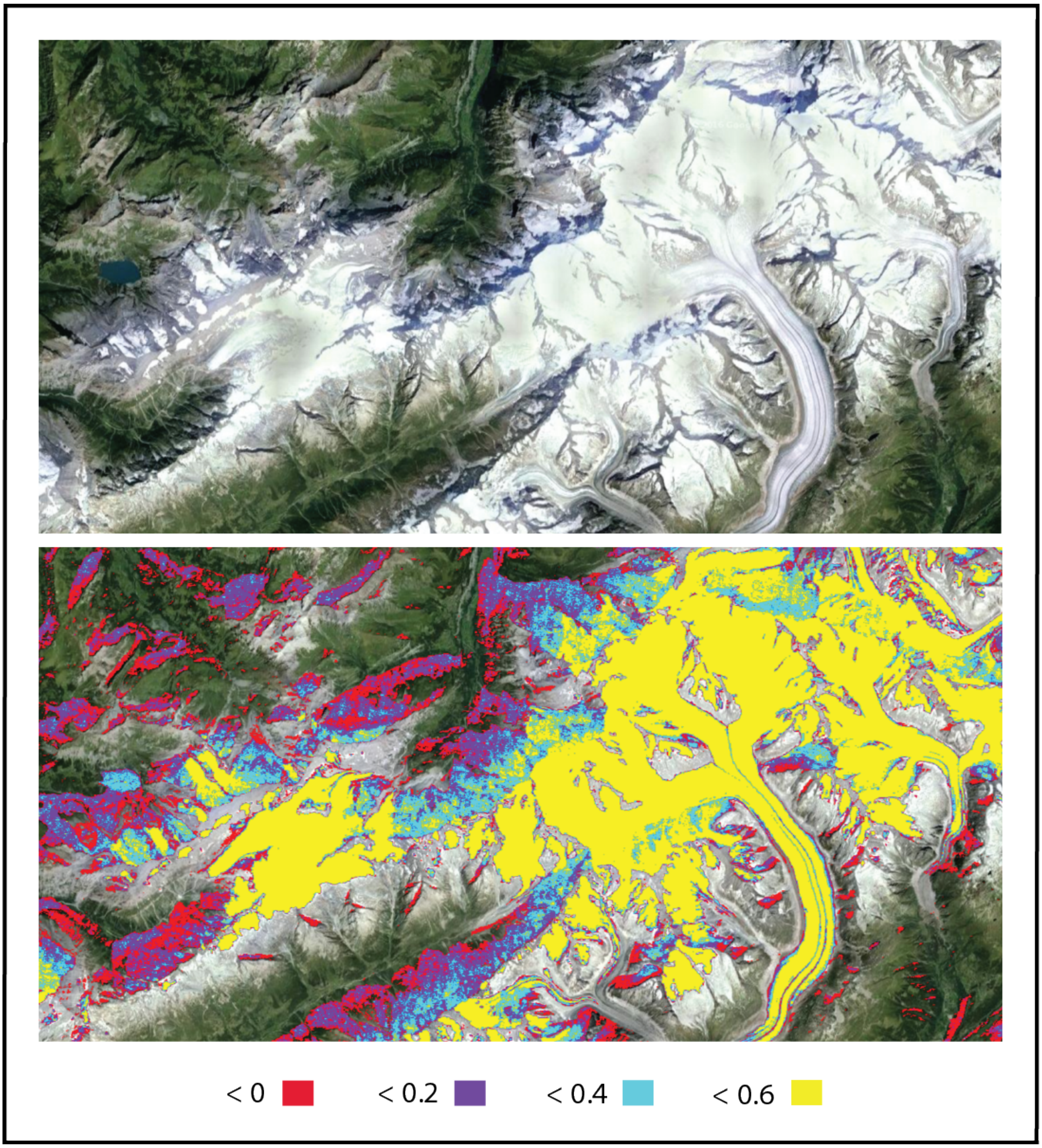


Figure 3.3: Use of different thresholds for the "Normalized Difference Snow Index" (NDSI) to define the perimeter on which the cloud-score is based. The NDSI is calculated based on raw digital numbers of a Landsat 5 scene (ID: LT5195028201127 9MOR00). Every pixel whose value lies above the respective threshold is added to the perimeter of further computation. Every classification color also includes the color of the thresholds above (e.g. violet classification includes turquoise and yellow pixels too).

Variable	Description
startJulian	defines the starting (julian-) day of analysis
endJulian	defines the ending (julian-) day of analysis
startYear	defines the starting year of analysis
endYear	defines the ending year of analysis
path	defines the path-information of the scene to analyse
row	defines the row-information of the scene to analyse
ss_LowThreshold	defines the lower threshold for values to accept in regard to the *snowcount* deviation value
ss_HighThreshold	defines the upper threshold for values to accept in regard to the *snowcount* deviation value
scale_otsu	defines the scale on which fnOtsu() are based on

Table 3.1: User-defined variables of Algorithm 1 for quick adjustments.

be outlined without much effort, it can be useful to (iii) manually draw a simple polygon with a lower amount of vertices within GEE.

When using (i), values above a certain threshold are masked out and ignored by the calculations. The threshold to separate snow and ice based on NDSI from other terrain is usually set around 0.6 (Choi & Bindschadler 2004). However, for this application a lower threshold of 0.2 has been chosen to widen the perimeter of analysis because clouds immediately next to the glacierized area can affect the suitability of a respective satellite scene in a negative way. Figure 3.3 illustrates the calculation of this perimeter with different thresholds for the NDSI (0, 0.2, 0.4 and 0.6 respectively) and illuminates the decision to take 0.2 as threshold within this method. For approaches (ii) and (iii) the perimeter of computation is defined by their respective geometry.

Images with a cloud-score higher than 15% within this perimeter are then discarded. This results in an `ImageCollection` only containing images with a low value for cloudiness.

Minimal seasonal snow cover

As for part (C) in Figures 3.1 and 3.2, the aim is to minimize the occurrence of seasonal snow. Two approaches are applied: A less accurate but much faster approach is to create an NDSI mask and classify pixels greater than 0.6 as snow. The more accurate approach is to calculate the Otsu threshold (Ostu 1979) based on the near infrared band for each image of the `ImageCollection` individually. Doing so, snow can even be separated from ice leading to a more specific count of snow pixels. However, this approach takes much more computation time. The function to retrieve the Otsu threshold is described in detail in Section 3.4. The *snowcount* is calculated for each image individually and stored as additional property of the image. Images with a high *snowcount* are discarded and ignored for further application. Images with a low *snowcount* are expected to be good as they indicate that during this period of time a low amount of seasonal snow is affecting the detection of glacierized area.

This entails the risk of falsely including images with reasonably high cloud cover as this can lead to an underestimated amount of seasonal snow cover. This effect is taken up again in Section 4.1.1.
At this point of the process there are again two separate ways to go depending on using either Algorithm 1.1 or Algorithm 1.2: Whereas in Algorithm 1.1 only scenes from an identical path and row are assembled, the `ImageCollection` first needs to be sorted by path and row to secondly sort it by the *snowcount* for Algorithm 1.2. For both the result is an `ImageCollection` only containing the most suitable scenes for glacier mapping, sorted in descending order of suitability.

3.2.2 Manual scene selection

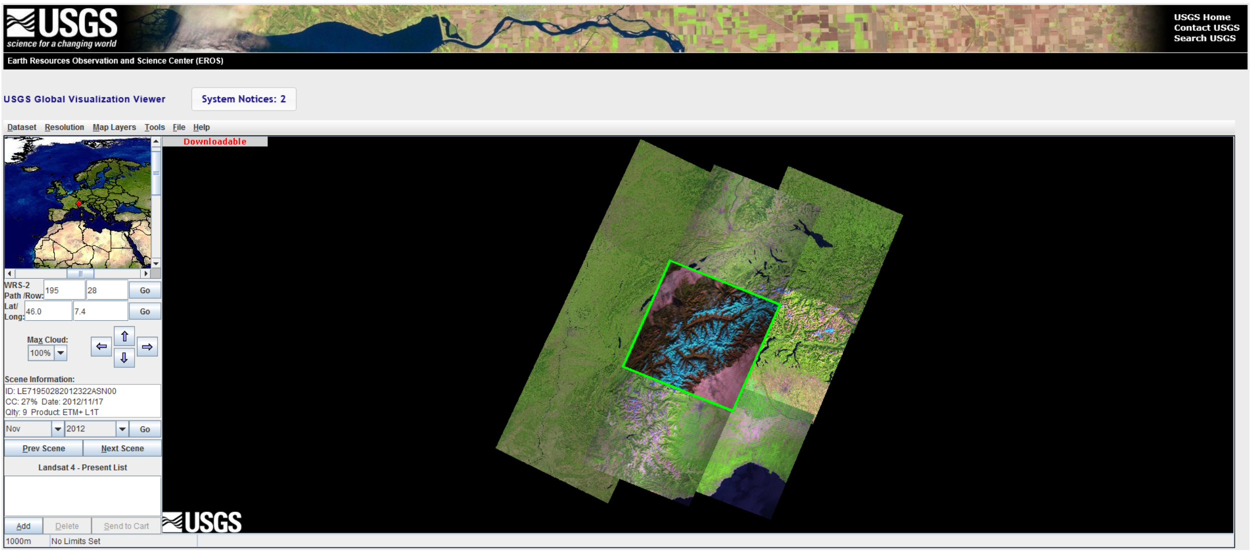
For the comparison of manual and automated scene selection the USGS Global Visualization Viewer (GLOVIS, see Figure 3.4) is used (U.S. Geological Survey 2005). It's containing different datasets such as from Landsat, ASTER, MODIS and Sentinel satellite missions. With GLOVIS one can preview images from specific locations and dates with all its metadata and eventually download the wanted scenes for further use in its original format and resolution. In the first level the pixels have a resolution of 1000 m which only provides a first glance at the images. The resolution can be set to 240 m per pixel resulting in a higher visibility with more details but slightly slower loading times. Eventually, images can be downloaded in orthorectified *.jpg*-format or in their full resolution including all the band information. After the pre-selection and the download of these images they are loaded into a GIS software (in this case QGIS Desktop 2.10.1) where they can be compared visually. Eventually, the best scenes can be determined by cloud cover- and seasonal snow cover comparison. This needs some experience of the observer but offers opportunities to select the scenes based on the particular application.

3.3 Area calculation and glacier extent

In this Section two main approaches are stated: First, Algorithm 2.1 is explained, which results in different area calculations for the set of glaciers it analyses over a given time range. The aim here is to import geometries from the RGI as references indicating the occurrence of glaciers and then calculate the glacierized area within and around them.
Algorithm 2.2 is set up in a similar way but ends with the calculation of the ratio image over a large scale and creates a mask of all glacierized areas within the region of analysis. This region is set far wider, resulting in longer computation time and the need (or possibility) to apply some post-processing steps to refine the the result.

3.3.1 Algorithm 2.1: Area calculation

In Algorithm 2.1 a slightly different approach in the process of finding suitable scenes for glacier mapping is used: in terms of minimal cloud cover the `qualityMosaic` method is applied in combination with a cloud scoring method, which creates one single image out of



WRS-2 Path /Row: 195 28

Lat/ Long: 46.0 7.4

Max Cloud: 100%

Scene Information:
ID: LE71950282012322ASN00
CC: 27% Date: 2012/11/17
Qlty: 9 Product: ETM+ L1T

Nov 2012

1000m No Limits Set

Figure 3.4: Interface of the USGS Global Visualization Viewer (GLOVIS). Retrieved from http://glovis.usgs.gov/

the least cloudy pixels of an `ImageCollection`. Therefore, the order of the two minimizing processes (cloud cover & occurrence of seasonal snow) are switched in comparison to Algorithms 1.1 & 1.2.
Image composites are helpful to get the most suitable pixels over a given time period for a specific application. By doing this, the occurrence of non-suitable data points decreases and results in a better accuracy. However, they represent an artificial consolidation and not an actual state of an environment. Luckily, most glaciers and their environment do not change quickly and only move very slowly. This allows creating image composites over several years (e.g. three to five) to achieve the most accurate representation of the state of a glacier.
The aim of Algorithm 2.1 is to create glacier outlines based on a selection of suitable scenes within a time period of several years and then calculate the area of this set of glaciers.

User-defined variables

Several user-defined variables are required to successfully run Algorithm 2.1. A selection of the important and easily adjustable ones is given in Table 3.2.
Changing these variables on one hand, offers a quick and easy handling of otherwise hidden variables. On the other hand, it offers some flexibility to adapt to error messages.

Spatial and temporal subsetting

As for Algorithm 2.1 (Figure 3.5) the filtering on spatial and temporal basis (A) are working in the same way as in Algorithm 1.2 (Section 3.2). On a spatial level, the geometry of the `FeatureCollection` is intersected with the starting `ImageCollection` and on temporal level a period of years is defined by which the `ImageCollection` is reduced.

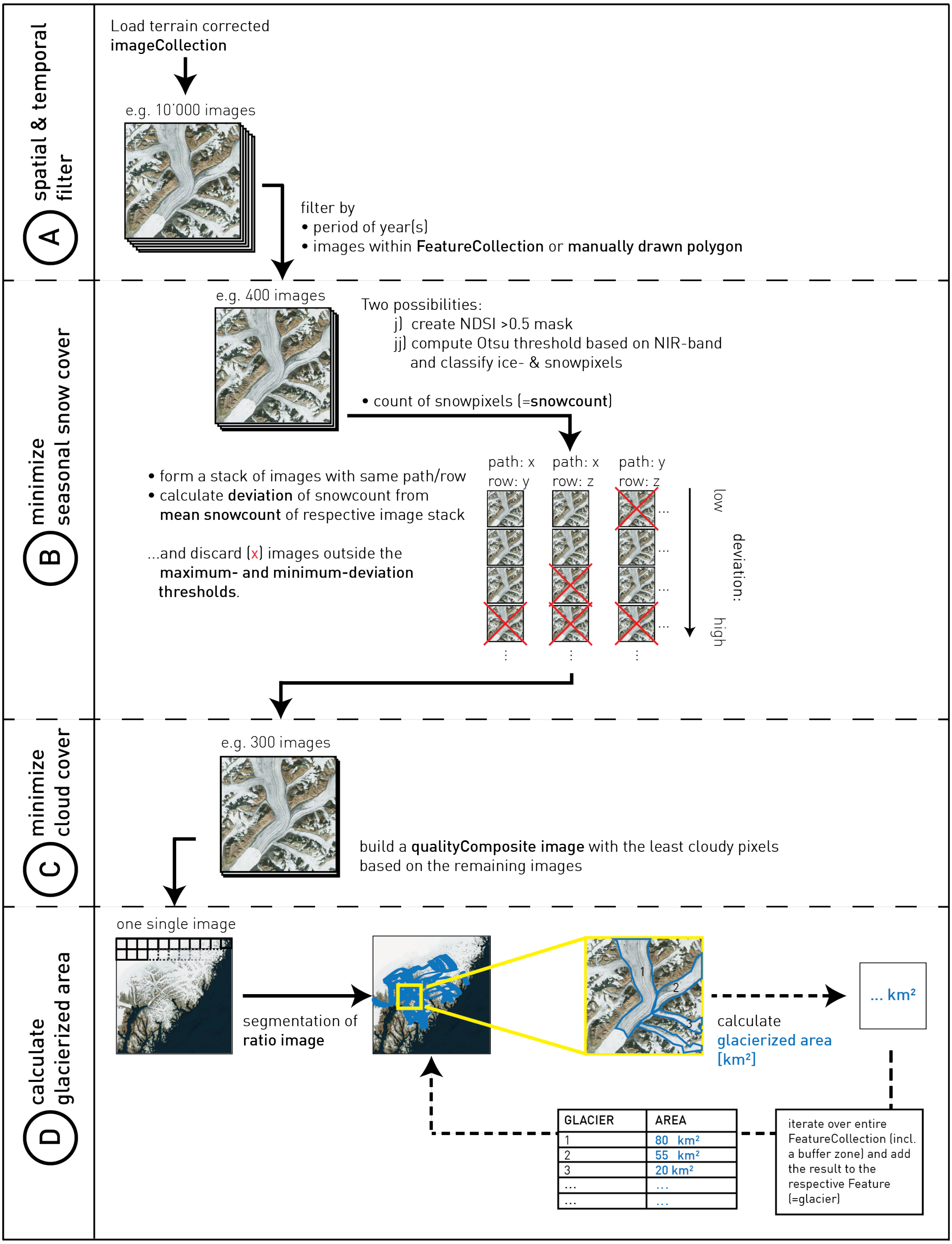

A spatial & temporal filter
Load terrain corrected imageCollection
e.g. 10'000 images
filter by
• period of year(s)
• images within FeatureCollection or manually drawn polygon
B minimize seasonal snow cover
e.g. 400 images
Two possibilities:
j) create NDSI >0.5 mask
jj) compute Otsu threshold based on NIR-band and classify ice- & snowpixels
• count of snowpixels (=snowcount)
• form a stack of images with same path/row
• calculate deviation of snowcount from mean snowcount of respective image stack
...and discard (x) images outside the maximum- and minimum-deviation thresholds.
path: x row: y
path: x row: z
path: y row: z
deviation:
low
high
C minimize cloud cover
e.g. 300 images
build a qualityComposite image with the least cloudy pixels based on the remaining images
D calculate glacierized area
one single image
segmentation of ratio image
1
2
3
calculate glacierized area [km²]
... km²
GLACIER | AREA
1 | 80 km²
2 | 55 km²
3 | 20 km²
... | ...
... | ...
iterate over entire FeatureCollection (incl. a buffer zone) and add the result to the respective Feature (=glacier)

Figure 3.5: Concept of Algorithm 2.1 with four sequential steps: selection of the end of ablation period (A), minimization of seasonal snow cover (B), minimization of seasonal snow cover (C) and the iteration process over an imported collection of glacier polygons (D).

Variable	Description
startYear	defines the starting year of analysis
endYear	defines the ending year of analysis
startJulian	defines the starting (julian-) day of analysis
endJulian	defines the ending (julian-) day of analysis
gte_ndsi	values above this threshold will be part of the ndsi mask generated to define the perimeter of computations (e.g. to minimize seasonal snow)
buffersize	size of the buffer around the polygons (in metres)
ss_LowThreshold	defines the lower threshold for values to accept in regard to the *snowcount* deviation value
ss_HighThreshold	defines the upper threshold for values to accept in regard to the *snowcount* deviation value
iceAndSnowThreshold	defines the threshold to use for the segmentation of the ratio image
scale_snowcount	defines the scale for the minimization of seasonal snow
scale_otsu	defines the scale on which the calculation of the maximum between-class-variance is calculated

Table 3.2: User-defined variables of Algorithm 2.1 for quick adjustments.

Minimal seasonal snow cover

As for part (B), the aim is to minimize the occurrence of seasonal snow. Like before, for each image individually in the `ImageCollection`, the Otsu threshold (see Section 3.4) is computed based on the near infrared band. After classifying the snow pixels the count can be computed for each image (*snowcount*). In the next step, the remaining satellite scenes are sorted by path and row and divided into sub-collections, each of them only containing images from identical path and row. Within these sub-collections the deviation of each image's *snowcount* from the *mean snowcount* of its respective sub-collection is then calculated (*snowcount-deviation-value*). Images with a deviation outside the defined maximum- and minimum-threshold are discarded. In a last step in part (B) these sub-collections are merged again for further calculations.

Minimal cloud cover

In part (C) the aim is to get the least possible cloud cover over the area of interest. The method `qualityMosaic` creates a composite image from all the remaining images in the `ImageCollection` based on a so-called quality band. In this case the quality band is defined as cloud score. Standard TOA calibration is applied to the images of the `ImageCollection` and then a cloud score is assigned to each pixel using the `simpleCloudscore` method described in Section 3.2.1. It selects the lowest possible range of cloud-scores at each pixel and computes per-band percentile values from the accepted pixels. The output is a single image covering the entire area of interest based on the imported FeatureCollection in the beginning.

Calculate glacierized area

In part (D) the single image from (C) serves as foundation to create a ratio image of two bands using Digital Numbers (DN). In the context of this study the bands 3 and 5 are used for Landsat 5 and 7. By using a manually defined threshold of 2 the output is then classified in *glacierized-* (above threshold) and *other terrain* (below threshold). For every feature of the FeatureCollection the pixels (containing the area in m^2) of this segmented ratio image, which intersect with the feature, are summed up. Like this, every glacier feature is provided with the information of its represented glacierized area within the given parameters of the algorithm.

Examples

Figure 3.6 shows a comparison of several Landsat 5 TM images recorded over summer 2003. All images have a distinct amount and spatial distribution of cloud cover which results in different suitability. Additionally, depending on the weather conditions of the respective time period the image was recorded, the amount of seasonal snow around the glacierized area varies a lot. Based on the amount of seasonal snow included within an image it is sorted out and ignored for the segmentation of the ratio image (see Section 3.3.1). Based on that resulting collection the pixel with the least cloud score is chosen for the image composite.

3.3.2 Algorithm 2.2: Semi-automatic method with additional manual post-processing in conventional GIS

As Figure 3.8 shows, Algorithm 2.2 is based on the same three (A-C) sequential steps of Algorithm 2.1. The difference lies, first, in a larger buffer zone around the imported FeatureCollection, which results in a larger area of analysis for the processes of seasonal snow cover (B) and cloud cover (C) minimization. Second, the last step (D) does not calculate the area within GEE but stops after the segmentation of the ratio image, which results in a detection of the glacierized area. However, testing of Algorithm 2.2 has proven to be more CPU-intensive than Algorithm 2.1 as the more effortful calculations lie in parts where pixels need to be counted. These sorts of computations mainly lie in part B and C and are intensified by the selection of large areas of analyses and time ranges.
The aim of this Algorithm, again, is to create glacier outlines based on a selection of suitable scenes within a time period of several years. But this time with additional manual post-processing with help of conventional GIS software.

Regional and global scale

This Algorithm has been tested with data from the RGI as well, mainly with its regional divides (see Section 2.1.2). As for most of the 19 RGI regions the export queries timed-out

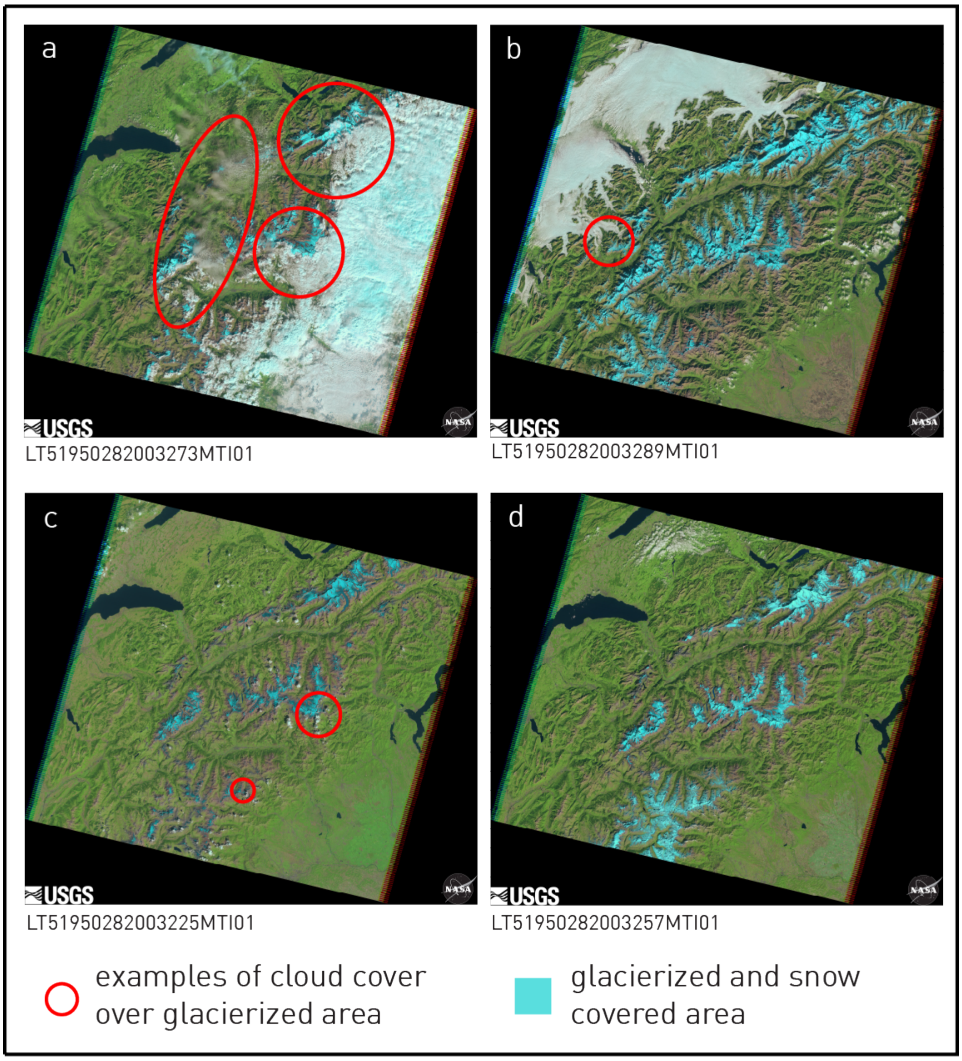


Figure 3.6: Comparison of several Landsat 5 TM scenes (pseudocolor representation with bands 5, 4 and 3) recorded over summer 2003 (path: 195, row: 28). Examples of cloud cover are highlighted red. Glacierized area is colored in cyan and usually includes a fair amount of seasonal snow. a) shows a good example for a scene with a very high cloud-score over the perimeter of interest, whereas b) shows that a scene can have a lot of cloud cover but not over the perimeter of interest. Scenes c) and d) are scenes with a low resulting cloud score but with c) being more suitable as it contains less amount of seasonal snow around the glacierized area.

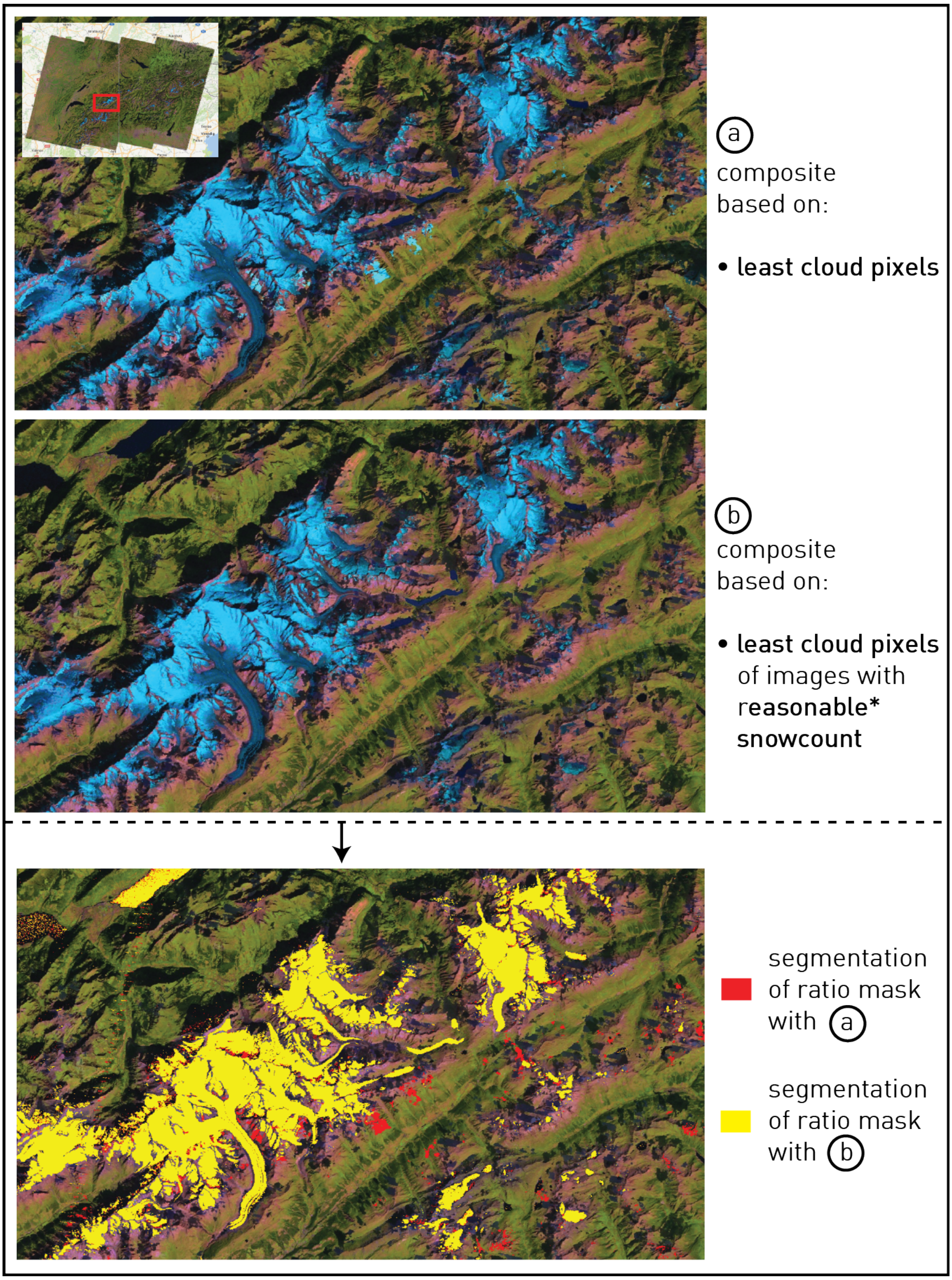


Figure 3.7: Creation of image composites in the Bernese and Valais Alps. Composite (a) is based on the least cloudy pixels of the `ImageCollection` whereas in (b) the images with reasonable* *snowcount* are discarded from the `ImageCollection` before the least cloudy pixels are selected for the composite.
*: only images with a *snowcount-deviation-value* between 0.5 and 1.2 are retained.

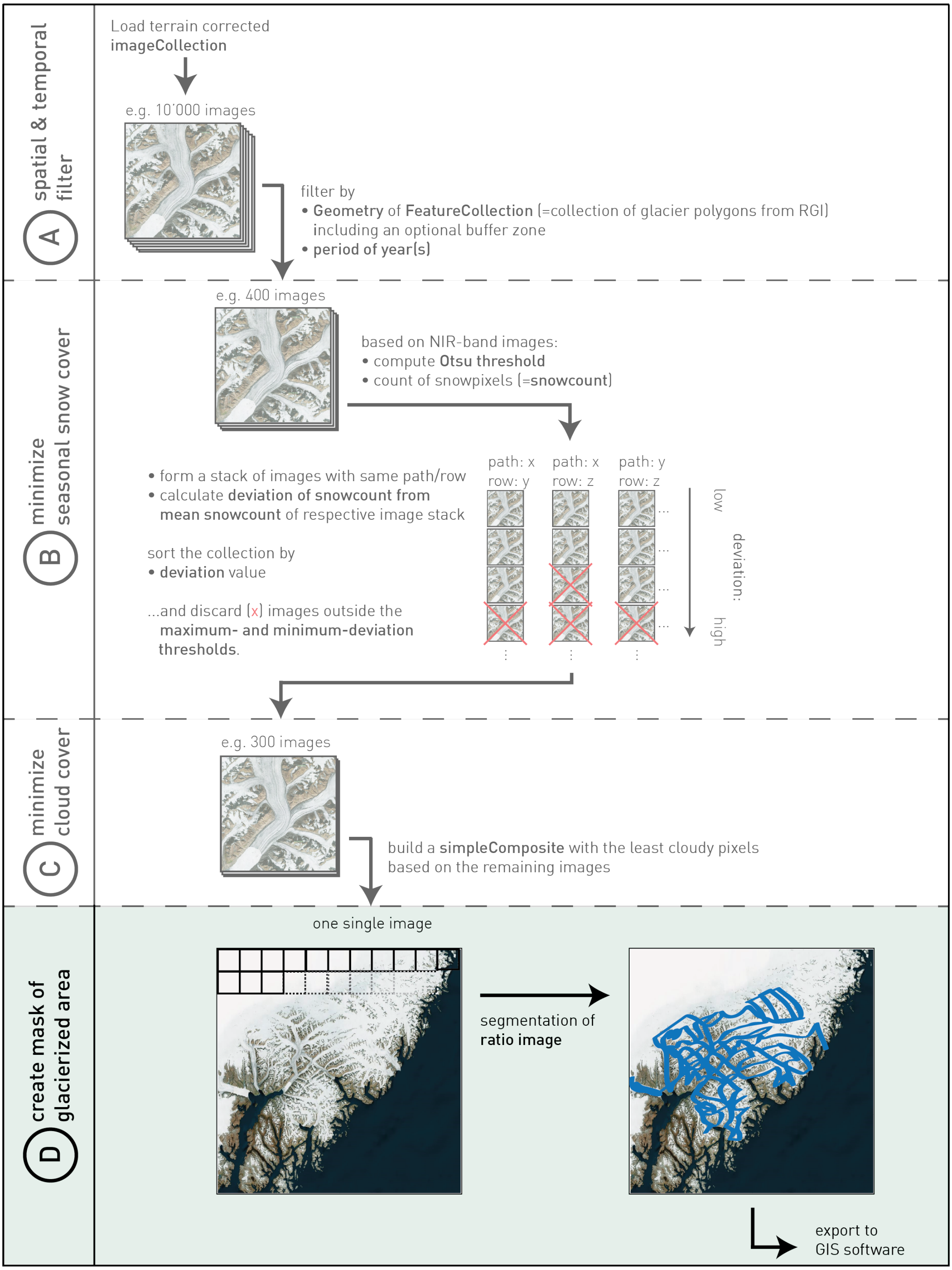

A spatial & temporal filter
Load terrain corrected imageCollection
e.g. 10´000 images
filter by
• Geometry of FeatureCollection (=collection of glacier polygons from RGI) including an optional buffer zone
• period of year(s)
B minimize seasonal snow cover
e.g. 400 images
based on NIR-band images:
• compute Otsu threshold
• count of snowpixels (=snowcount)
• form a stack of images with same path/row
• calculate deviation of snowcount from mean snowcount of respective image stack
sort the collection by
• deviation value
...and discard (x) images outside the maximum- and minimum-deviation thresholds.
path: x
row: y
path: x
row: z
path: y
row: z
deviation:
low
high
C minimize cloud cover
e.g. 300 images
build a simpleComposite with the least cloudy pixels based on the remaining images
D create mask of glacierized area
one single image
segmentation of ratio image
export to GIS software

Figure 3.8: Concept of Algorithm 2.2 with the three identical sequential steps (A), (B), (C) from Algorithm 2.1: selection of the end of ablation period (A), minimization of seasonal snow cover (B), minimization of seasonal snow cover (C). The difference is in (D) which is designed to calculate the ratio image over a large scale and create a mask of all glacierized area.

with a scale of 30m per pixel this parameter was increased to values between 120 m and 240 m per pixel to be able to finish the exports and generate outputs. This shows that the computations within Algorithm 2.2 are quite CPU-intensive.

Post-processing steps

The outputs of GEE are in the format of *geoTIFF* and can thus be imported into regular GIS applications. For this thesis the open source GIS software *QGIS* is used to import and display data in this format. The outputs are intersected with datasets containing water bodies which are subtracted from the end result. Additionally, obvious mis-classifications are visually corrected and removed.

3.3.3 Usage limits

During the computation process error messages can occur when

1. the time range is too big, resulting in a huge collection of images
2. the `FeatureCollection` is too big or contains an enormous amount of vertices
3. both of the above together.

In this case it can cause time-out messages when exporting the results. However, there are possibilities to evade them (see Figure 3.9). The scale parameters of either the snow cover minimization method or the one from the segmented ratio image can be increased accordingly (resulting in a loss of accuracy) or the time range and/or the `FeatureCollection` has to be chopped up leading to loads of manual work. Another possibility in regard to the minimization of snow cover is to bypass the effortful Otsu method by a less time-consuming NDSI calculation or even bypass it completely depending on the amount of data to be processed.

3.4 Mapping of snow- and ice cover

3.4.1 Algorithm 3: Using the Otsu thresholding function to map snow cover and approximate the accumulation area ratio (AAR)

A truly valuable asses in glacier monitoring would be to automatically provide estimates related to glacier mass balance. The concept of accumulation area ratio (AAR) is often applied as a proxy to a glacier's mass balance. Nevertheless, the calculation of the AAR is not easy to implement in an automated manner as the outlines of the accumulation area and the ablation area respectively would need to be known. One could assume that by mapping snow- and ice cover of a glacier these areas could at least be approximated. In the following, this approach to approximate the AAR within GEE is presented by using the Otsu-thresholding function (Ostu 1979).

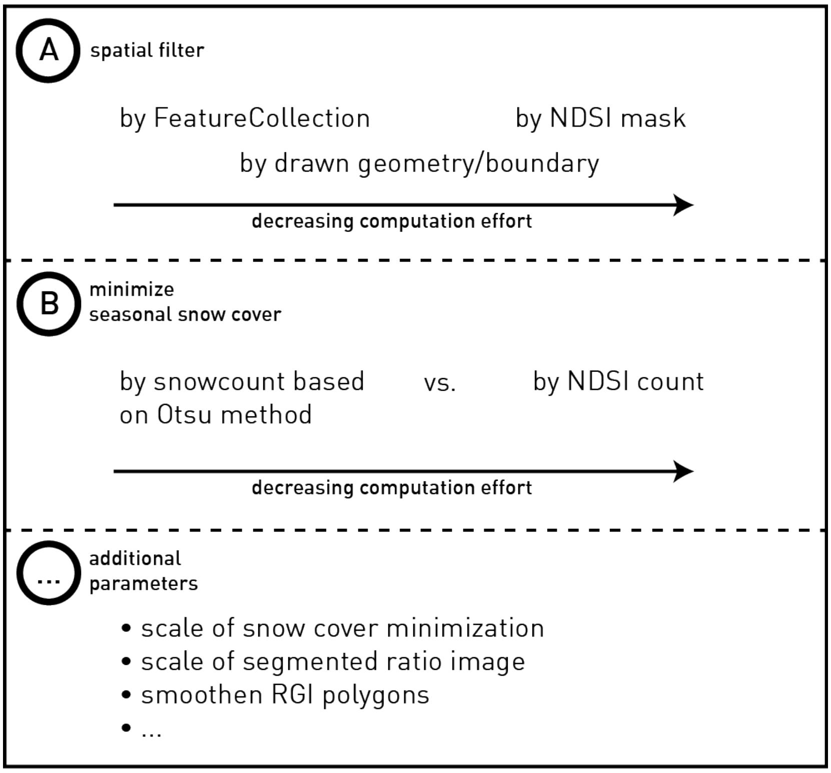
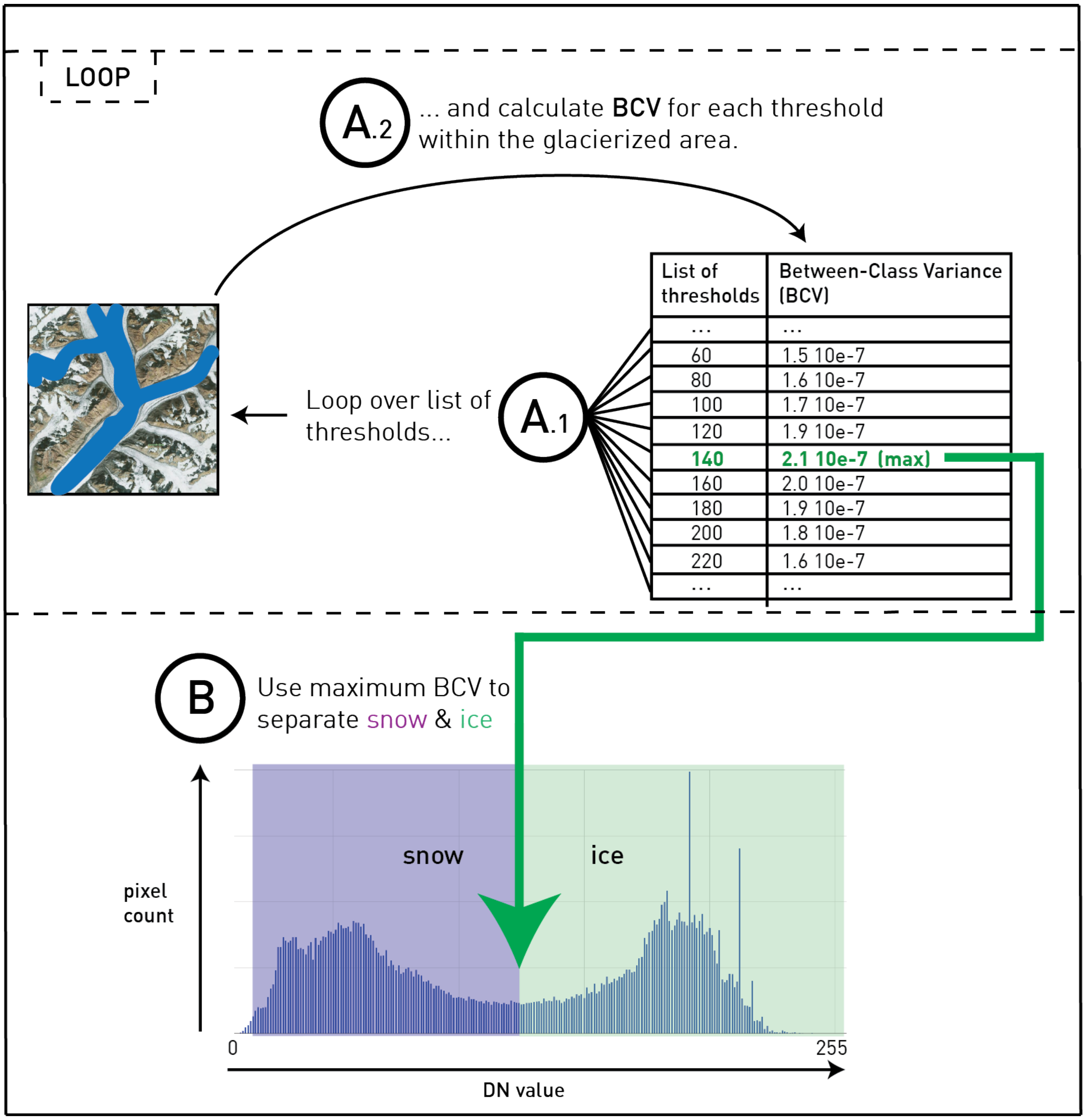

A
spatial filter
by FeatureCollection
by NDSI mask
by drawn geometry/boundary
decreasing computation effort
B
minimize
seasonal snow cover
by snowcount based on Otsu method
vs.
by NDSI count
decreasing computation effort
...
additional parameters
• scale of snow cover minimization
• scale of segmented ratio image
• smoothen RGI polygons
• ...

Figure 3.9: Possible adaptions to error- and time out messages: (A) either a FeatureCollection or an NDSI mask can be used to spatially filter the images and localize further calculations. (B) In regard to the minimization of seasonal snow cover either the Otsu method or a more simple NDSI count can be used to try to relatively quantify seasonal snow cover. In addition to that it is possible to smoothen the RGI polygons before importing them or rising the scale of either the snow cover minimization method or the one from the segmented ratio image.


LOOP
A.2
... and calculate BCV for each threshold within the glacierized area.
Loop over list of thresholds...
A.1
B
Use maximum BCV to separate snow & ice
snow
ice
pixel count
0
255
DN value

List of thresholds	Between-Class Variance (BCV)
...	...
60	1.5 10e-7
80	1.6 10e-7
100	1.7 10e-7
120	1.9 10e-7
140	**2.1 10e-7 (max)**
160	2.0 10e-7
180	1.9 10e-7
200	1.8 10e-7
220	1.6 10e-7
...	...

Figure 3.10: Concept of the implemented Otsu method to separate snow from ice on a glacier. A.1 and A.2 define a loop, resulting in finding the threshold which corresponds to the maximum Between-Class Variance.

Implementation of Otsu-thresholding function

Thresholding is a simple and commonly used technique for image segmentation (Pal & Pal 1993). The Otsu method is an image thresholding function named after Nobuyuki Otsu (Ostu 1979), which allows to perform clustering-based image thresholding and to reduce an image to a binary image. The method separates the image into two classes (named foreground- and background pixels) with a threshold and then calculates the variance for both classes. It then iterates through all the possible threshold values with the aim to find the threshold value where the sum of foreground and background variance is at its minimum (Ostu 1979). Luckily, the threshold with the minimum *within*-class variance is also the threshold with the maximum *inter*-class variance (Greensted 2010). The latter is used within Algorithm 3 (see Figure 3.10) as it is easier to implement. The following equation serves as mathematical basis:

$$
\begin{aligned}
Between\,Class\,Variance \quad \sigma_B^2 &= \sigma^2 - \sigma_W^2 && (3.1)\\
&= W_b(\mu_b - \mu_f)^2 + W_f(\mu_f - \mu)^2 \\
&\quad (where\,\mu = W_b\mu_b + W_f\mu_f) && (3.2)\\
&= W_b * W_f(\mu_b - \mu_f)^2 && (3.3)
\end{aligned}
$$

$$
\begin{aligned}
\sigma_B^2 &= Between\,Class\,Variance\\
\sigma_W^2 &= Within\,Class\,Variance\\
W_f &= weight, foreground\,(values\,above\,threshold)\\
W_b &= weight, background\,(values\,below\,threshold)\\
\mu_f &= mean, foreground\,(values\,above\,threshold)\\
\mu_b &= mean, background\,(values\,below\,threshold)\\
\mu &= overall\,mean
\end{aligned}
$$

Snow cover mapping

The Otsu method can be used to distinguish between ice and snow within the glacierized area. In the fields of glaciology it performs well to separate ice and snow if the thresholding function is used on a near infrared (NIR) band (e.g. band 4 for Landsat 7) and has been tested in other studies as well (De Angelis 2014). Figure 3.11 shows an example of how these snow- and ice-masks are calculated based on a Landsat 7 raw image from summer 2015 with use of the implementation of the Otsu method in GEE. By having this separation of snow- and ice pixels and calculating the amount of pixels in each class an approximation to the AAR could be done. However, to correctly calculate the AAR one would need the exact outlines of the accumulation and ablation area. The algorithm is mapping snow- and ice cover respectively, which only serves as proxy to accumulation- and ablation area. Nevertheless, by computing such an approximated AAR (proportion of the count of snow

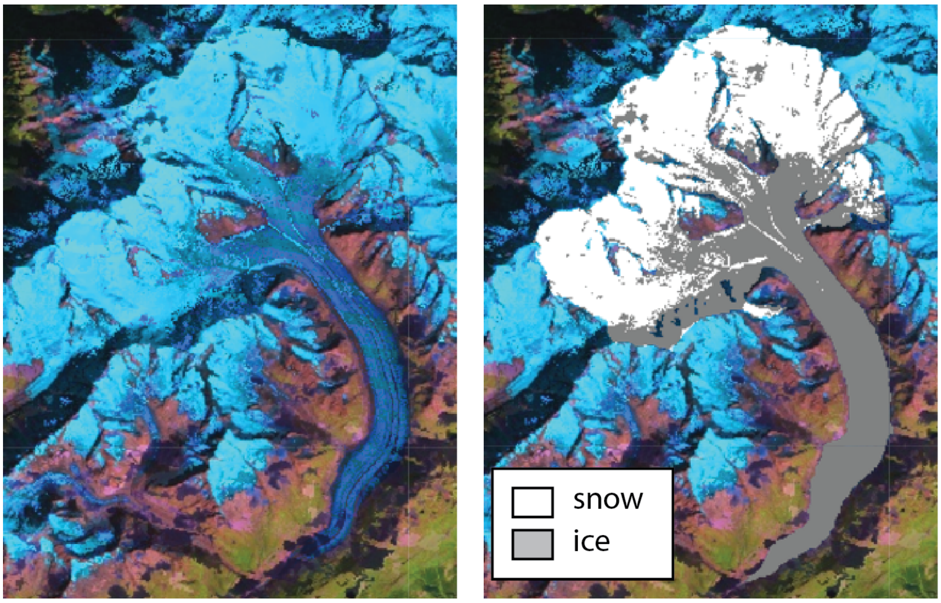


Figure 3.11: Example of snow and ice separation for *Aletschgletscher* based on the implementation of the Otsu method. The left shows a Landsat 5 and 7 image composite of an automated scene selection for 2010. The right shows the classification based on the same image composite.

pixels in relation to the total pixel count) a comparable trend could be identified in regard to manual AAR calculations from previous studies.

3.4.2 Trend comparison

To be able to compare the results from the approximated AAR calculation by mapping the snow covered area within GEE with data from manual AAR calculations, the series 'Fluctuations of Glaciers' (FoG) is consulted and used as relation. Prepared by the World Glacier Monitoring System (WGMS) it continuously publishes internationally collected and standardized data on changes in glaciers on global level at intervals of 5 years (WGMS 2008, 2012, e.g.). The glaciers which will serve as basis for that trend comparison are:

- Findelen (Switzerland)
- Kesselwand (Austria)
- Storglaciaeren (Sweden)

3.5 Separation of clean- and debris-covered ice

The RGI is a combination of new and previously-published glacier outlines (Pfeffer et al. 2014) and is publicly available in its fifth version by the time of this thesis (`http://www.glims.org/RGI`). The glacier outlines are usually acquired using automated segmentation

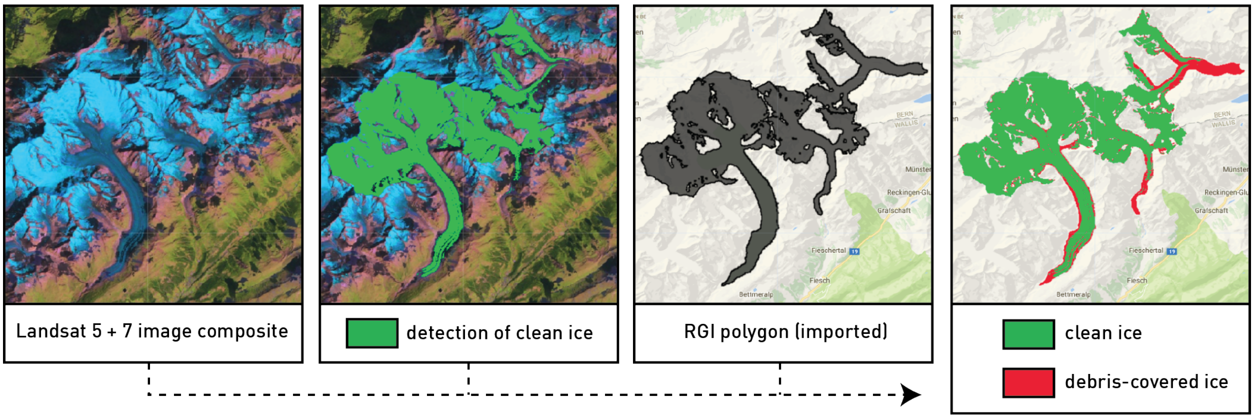


Figure 3.12: Clean-ice and debris-covered ice classification with use of imported RGI glacier polygons.

of ratio images too but with additional manual editing to deal with incorrect classification of glacierized area such as regions cast by shadow or covered by debris, thick enough to hide the spectral information of the ice underneath. In several regions outlines have been completely manually digitized, either from satellite data or topographic maps.
As findings of Chapter 4 will show, the algorithms presented in this thesis do not perform well either in shadow zones and debris-covered glacier parts. Especially debris cover remains a big issue when making use of fully automated approaches without any manual corrections. But although this might pose a problem to completely detect glacierized area, it poses the opportunity to supply the RGI with a debris mask, simply by intersecting the glacier outlines retrieved from GEE with the outlines of the RGI dataset. In the following the term *clean ice* will refer to the part of the glacierized area whose spectral information is not hidden under debris-cover (see Figure 3.12). However, *clean ice* is not referring to pure ice but can contain dirt. It rather indicates the part of the glacier which can be detected by the segmentation of a ratio image. From a spectral classification point of view, this means, that as long as a pixel (e.g. 30 m x 30 m) contains a certain share of (possibly) dirty ice it is classified as *clean ice* within the algorithm.

Keeping in mind that RGI data is not specifically acquired within a narrow time range but over a large time period, it is important to align the date range of data acquired within GEE to the dates of the respective dataset from the RGI. Without matching points in time it won't be meaningful to create such a debris mask. As for some regions throughout the world it is difficult to retrieve the exact dates the RGI is based on, debris masks are created here for two example regions only:

- Swiss Alps (with focus on Bernese and Valais Alps)
- Central- and West Asia (with focus on Karakoram mountains)

4 Results

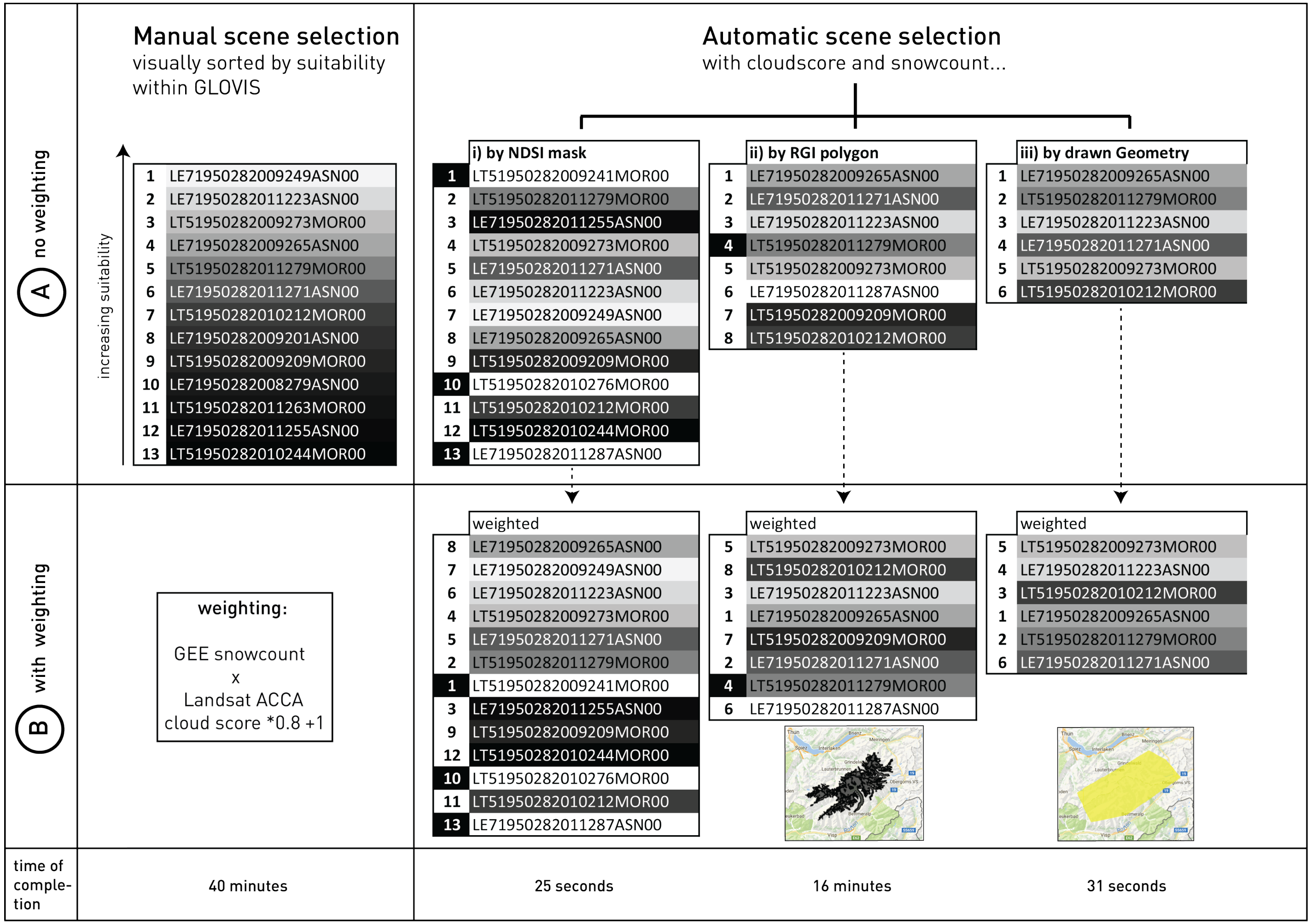
4.1 Scene selection

In the following sections the fully automated algorithms 1.1 and 1.2 are compared to the manual way of scene selection. According to the respective algorithm the manual scene selection is done within the same satellite imagery archives and within the same time range. The comparison is based on the resulting scene selection and the needed time to process the data. In this example Landsat 7 Enhanced Thematic Mapper Plus (ETM+) and Landsat 5 Thematic Mapper (TM) scenes between July 15th and October 15th from 2009 to 2011 have been explored. As for the different approaches to count seasonal snow (see j) and jj) in Figures 4.1 & 4.3) the one using the automatic setting of the Otsu threshold proved to be too time consuming when the set of images appears to be rather large. Consequently, the other approach is used in the following, using an NDSI mask for every image to determine the amount of seasonal snow.

4.1.1 Comparison of single scene selection (Algorithm 1.1): manual vs. automated method

Resulting scene selection

In figure 4.1 the comparison between the manual scene selection using GLOVIS (one specific path/row) and the automatic image selection within GEE and its three different approaches of perimeter determination for cloud-score and *snowcount* is illustrated (Algorithm 1.1, see figure 3.1). The result of the manual scene selection is taken as model. Hence, the automatic image selection should ideally result in the same image selection. The manual scene selection has been constructed to include a fair amount of images, whilst the least suitable rated images would not be useful on their own but still contain small regions of glacierized area which suit the respective conditions. The result of the original method (see Figure 3.1) is shown in (A) whereas (B) includes a weighting by the Landsat ACCA cloud-score of each scene (Irish 2000). The identical number of images selected manually and in (i) happened by coincidence. In approach (i) the automatic image selection selects 10 out of 13 identical scenes as with the manual selection but misses another 3 scenes in comparison with the manual image selection (all because of a mean cloud-score above the set threshold). However, the 3 additional images automatically selected in (i) all include a high amount of low hanging and thick clouds partially lying over the glacierized area of the image (see Figure 4.2). It can happen that the algorithm then classifies such scenes as suitable, even though they're not. Particularly if the clouds are thick enough to cover the spectral information beneath



A no weighting

Manual scene selection
visually sorted by suitability
within GLOVIS

1	LE71950282009249ASN00
2	LE71950282011223ASN00
3	LT51950282009273MOR00
4	LE71950282009265ASN00
5	LT51950282011279MOR00
6	LE71950282011271ASN00
7	LT51950282010212MOR00
8	LE71950282009201ASN00
9	LT51950282009209MOR00
10	LE71950282008279ASN00
11	LT51950282011263MOR00
12	LE71950282011255ASN00
13	LT51950282010244MOR00

Automatic scene selection
with cloudscore and snowcount...

i) by NDSI mask

1	LT51950282009241MOR00
2	LT51950282011279MOR00
3	LE71950282011255ASN00
4	LT51950282009273MOR00
5	LE71950282011271ASN00
6	LE71950282011223ASN00
7	LE71950282009249ASN00
8	LE71950282009265ASN00
9	LT51950282009209MOR00
10	LT51950282010276MOR00
11	LT51950282010212MOR00
12	LT51950282010244MOR00
13	LE71950282011287ASN00

ii) by RGI polygon

1	LE71950282009265ASN00
2	LE71950282011271ASN00
3	LE71950282011223ASN00
4	LT51950282011279MOR00
5	LT51950282009273MOR00
6	LE71950282011287ASN00
7	LT51950282009209MOR00
8	LT51950282010212MOR00

iii) by drawn Geometry

1	LE71950282009265ASN00
2	LT51950282011279MOR00
3	LE71950282011223ASN00
4	LE71950282011271ASN00
5	LT51950282009273MOR00
6	LT51950282010212MOR00

B with weighting

weighting:

GEE snowcount
x
Landsat ACCA
cloud score *0.8 +1

weighted

8	LE71950282009265ASN00
7	LE71950282009249ASN00
6	LE71950282011223ASN00
4	LT51950282009273MOR00
5	LE71950282011271ASN00
2	LT51950282011279MOR00
1	LT51950282009241MOR00
3	LE71950282011255ASN00
9	LT51950282009209MOR00
12	LT51950282010244MOR00
10	LT51950282010276MOR00
11	LT51950282010212MOR00
13	LE71950282011287ASN00

weighted

5	LT51950282009273MOR00
8	LT51950282010212MOR00
3	LE71950282011223ASN00
1	LE71950282009265ASN00
7	LT51950282009209MOR00
2	LE71950282011271ASN00
4	LT51950282011279MOR00
6	LE71950282011287ASN00

weighted

5	LT51950282009273MOR00
4	LE71950282011223ASN00
3	LT51950282010212MOR00
1	LE71950282009265ASN00
2	LT51950282011279MOR00
6	LE71950282011271ASN00

time of completion

40 minutes

25 seconds

16 minutes

31 seconds

Figure 4.1: Comparison of manual scene selection using GLOVIS and automatic scene selection within GEE (Algorithm 1.1) from 2009 to 2011 including Landsat 5 and 7 imagery data. Identical shadings of manually (left) and automatically (right) selected Landsat scene IDs show the same IDs. Numbers with black background indicate that these scenes are missing in the manual scene selection. Two different results are shown: (A) uses Algorithm 1.1 as described in figure 3.1 without weighting of the resulting scene selection which is sorted by the calculated *snowcount*. (B) uses a weighting by the Landsat ACCA cloud-score of each scene (Irish 2000) to deal with undervalued *snowcount* calculated due to high cloud cover.

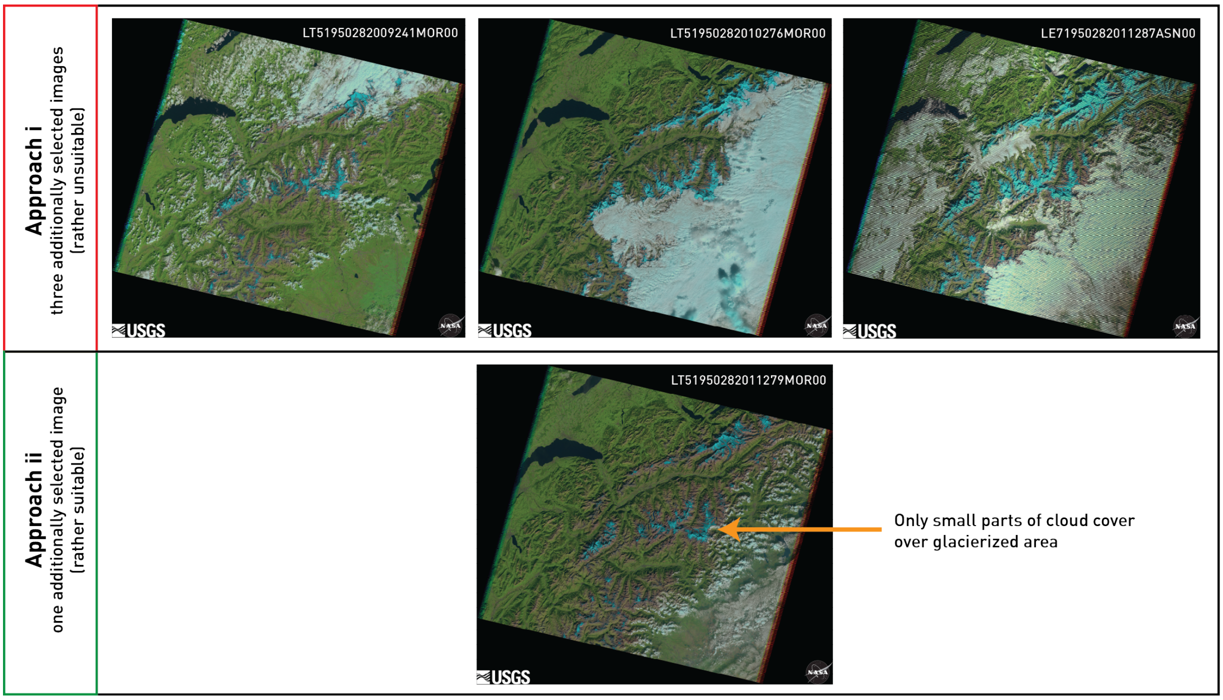


Figure 4.2: Additionally selected scenes of the automated approaches: i) rather unsuitable scenes due to high cloud cover near and partially over the glacierized area. ii) rather suitable scene which mistakenly wasn't selected manually.

them. As a result, the NDSI mask, which defines the perimeter of the cloud analysis, turns out to be too small. This setting results in a low *snowcount* and a mistakenly high estimated suitability. Because of the clear bounds of the clouds the mean cloud-score within the NDSI perimeter stays reasonably low and the scene is not discarded from further analysis.

Approach (ii) uses the imported RGI polygons to define the area of analysis for the cloud-score and the subsequent *snowcount*. It detects 7 of the most suitable images but includes one image (see Figure 4.2) with a considerable amount of clouds. However, this image should have been part of the manual scene selection. The cloud cover barely interferes with the glacierized area. This example indicates that by manually selecting suitable scenes it can happen to miss specific images, whereas the automatic selection algorithm shows more consistent results.

With approach (iii) a total of 6 images are selected. They appear to be the 6 best suitable ones as well when compared with the image selection performed manually.

Comparing the different approaches presented in figure 4.1 (i-iii) one can recognise that (ii) and (iii) are more limiting on the resulting scene selection whereas approach (i) collects a wider series of images. Surprisingly, (iii) performs even better in regard to sorting. The best images tend to be at higher positions in the resulting order. In regard to computation time used to complete the respective image selection algorithm (i) and (iii) perform faster by far compared with the approach using the complete RGI polygons (ii). Whenever the region of interest is rather small it is thus recommendable to use either approach (i) or (iii).

As for part (B) in figure 4.1 a weighting of the *snowcount* resulting from GEE with the Landsat ACCA cloud-score (Irish 2000) is performed. With this weighting the above explained issue of undervalued *snowcounts* due to high cloud cover can be dampened. The ACCA score is a proper tool used on Landsat images to calculate the mean cloud cover of a scene, helping end-users to filter the Landsat archive by scenes with a reasonably low ACCA score. Scenes with a high ACCA score tend to perform slightly poorer than scenes with a low ACCA score. By comparing the resulting scene selection of part (B) one can tell that the sorting process is improved by applying the weighting.

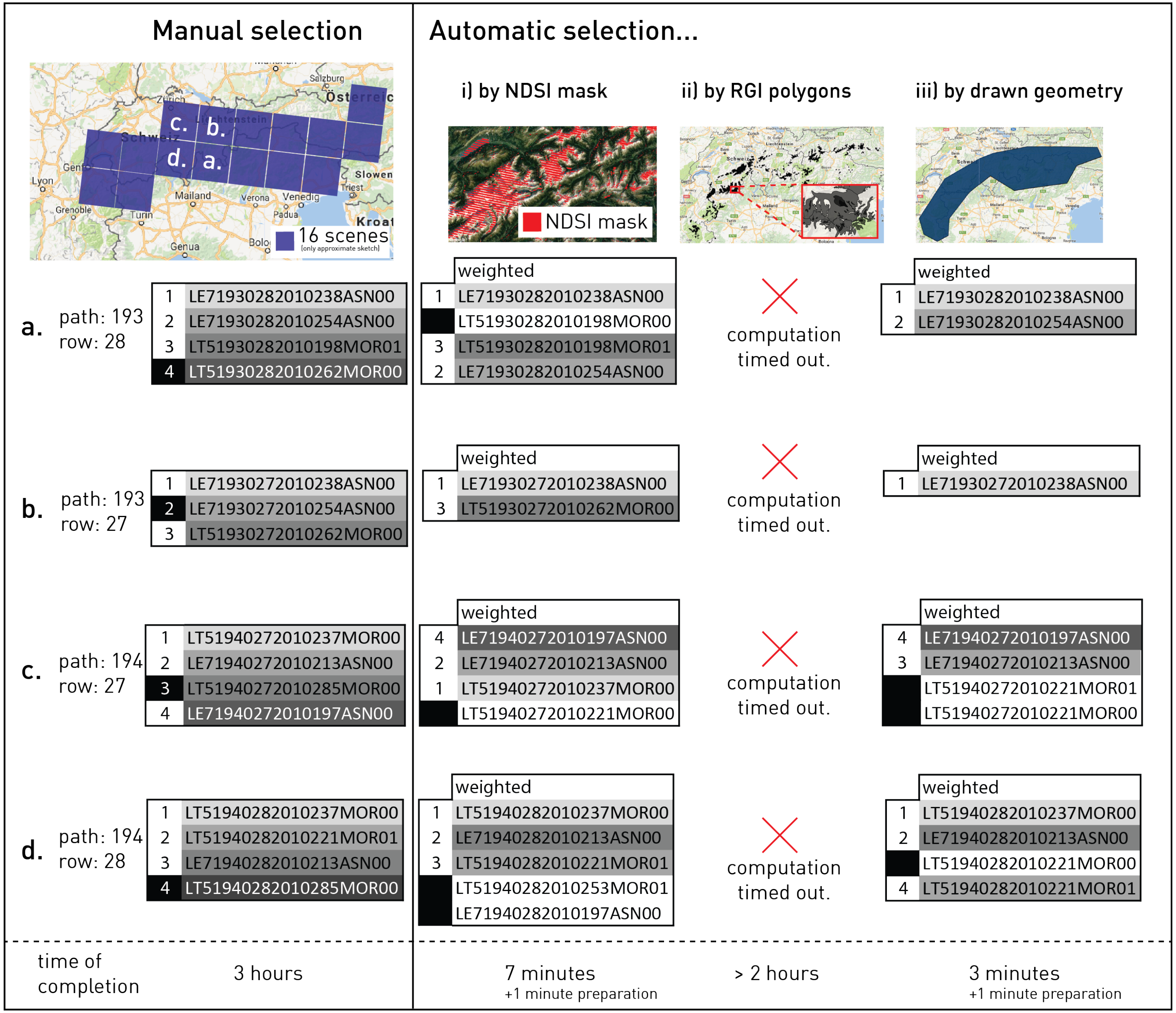
Processing time

In regard to processing time it took around 20 minutes to manually find suitable images using GLOVIS and another 20 minutes to take a closer look at a higher resolution and sort the resulting selection, making it approximately 40 minutes of total time to find suitable images to work with.
For the automated image selections within GEE it takes approximately one minute to enter the correct variables. However, that effort is not included in the different durations of the automatic image selection.
For (i) the runtime was 40 seconds. Compared to the manual image selection this approach is 60-times faster.
For (ii) the overall runtime of 16 minutes reveals that this approach takes longer to compute because of the mostly complex structure of the individual RGI polygons. The higher the amount of vertices of polygons the algorithm has to deal with, the longer it takes to compute the result.
That's also why approach (iii) completes a lot faster compared to (ii). With a total of six vertices the polygon is a lot simpler than the RGI polygons.
Eventually, approaches (i) and (iii) complete in a by far smaller period of time compared with the approach using the complete RGI polygons (iii). Whenever the region of interest is rather small it is thus recommendable to either use approach (i) or (iii). Nevertheless, as will be seen in the following Section 4.1.2 when the region of analysis gets larger it will get more and more time consuming to manually draw a geometry.
The observations mentioned above have been the same for different time periods and locations in the world. Nonetheless, it must not be forgotten that although all the automatic image selection approaches do deliver their own selection, the accuracy of the manual scene selection might still be arguably better compared to the automatic processing, as a trained glaciologist is easily able to interpret the changing conditions between individual scenes.

4.1.2 Comparison of multiple scene selection (Algorithm 1.2): manual vs. automated method

Resulting scene selection

As in Section 4.1.1, the results of the manual image selection are taken as role model. In Figure 4.3 only a selection of the scenes (path/row: 193/28, 193/27, 194/27 and 194/28) is



	Manual selection	i) by NDSI mask (weighted)	ii) by RGI polygons	iii) by drawn geometry (weighted)
a. path: 193 row: 28	1 LE71930282010238ASN00 2 LE71930282010254ASN00 3 LT51930282010198MOR01 4 LT51930282010262MOR00	1 LE71930282010238ASN00 LT51930282010198MOR00 3 LT51930282010198MOR01 2 LE71930282010254ASN00	computation timed out.	1 LE71930282010238ASN00 2 LE71930282010254ASN00
b. path: 193 row: 27	1 LE71930272010238ASN00 2 LE71930272010254ASN00 3 LT51930272010262MOR00	1 LE71930272010238ASN00 3 LT51930272010262MOR00	computation timed out.	1 LE71930272010238ASN00
c. path: 194 row: 27	1 LT51940272010237MOR00 2 LE71940272010213ASN00 3 LT51940272010285MOR00 4 LE71940272010197ASN00	4 LE71940272010197ASN00 2 LE71940272010213ASN00 1 LT51940272010237MOR00 LT51940272010221MOR00	computation timed out.	4 LE71940272010197ASN00 3 LE71940272010213ASN00 LT51940272010221MOR01 LT51940272010221MOR00
d. path: 194 row: 28	1 LT51940282010237MOR00 2 LT51940282010221MOR01 3 LE71940282010213ASN00 4 LT51940282010285MOR00	1 LT51940282010237MOR00 2 LE71940282010213ASN00 3 LT51940282010221MOR01 LT51940282010253MOR01 LE71940282010197ASN00	computation timed out.	1 LT51940282010237MOR00 2 LE71940282010213ASN00 LT51940282010221MOR00 4 LT51940282010221MOR01
time of completion	3 hours	7 minutes +1 minute preparation	> 2 hours	3 minutes +1 minute preparation

Figure 4.3: Region of analysis represents the area of central Europe's glaciers. It results in 16 different scenes covering the region. The manual and the different approaches of the automatic image selection are compared for four different scenes (a-d). The grey color gradient indicates the order of the manual image selection. Same grey tones indicate matching images in the respective manual and automatic image selection. Images which are missing in the opposing selection are highlighted black.

listed in comparison with the results of the manual selection. For all the automatic image selections (i-iii) the weighting presented earlier in Section 4.1.1 is directly applied within the algorithm. Using approach (i), a total of 58 images for 16 different scenes for the year 2010 have been selected. This makes an average of approximately 3.6 images for a period from 15th July to 15th October per scene and year. Approach (ii), which uses the original RGI polygons as reference for the cloud-score on every image, unfortunately timed out after two hours attempting to export the selection. As for approach (iii) a total of 22 images have been selected which makes an average of approximately 1.4 images per scene and year.

Overall, the automatic image selection processes recognise suitable scenes as such. As for the images the algorithm does not detect as suitable images, the deviation value of the *snowcount* is usually too high resulting in discarding the respective image from the selection (used upper threshold: 1.2).
As for example in approach (i) the following deviation values are calculated: 1.43 for (a), 1.39 (b), 1.25 (c) and 1.46 (d). In this example no image which has been discarded due to a too low value (threshold used: 0.5) has been found to be suitable after checking the image manually. Hence, at least the lower threshold seems to be set appropriately.
As for the images which are "falsely" selected from the algorithm, the issue with the thick cloud cover surrounding the perimeter of analysis as described in Section 4.1.1 comes into effect: The respective images all come with reasonably high cloud cover which affects the glacierized area only partially but is thick enough to cover the spectral information beneath them and significantly reduce the area classified by applying the NDSI classification. The algorithm thus falsely selects them as suitable images. As stated, the applied weighting with the ACCA cloud score should take care of this. However, if clouds almost exclusively occur in the region of glaciers and snow covered areas then even the weighting does not perform flawlessly.

Processing time

In regard to processing time it took around five minutes per scene to manually find suitable images using GLOVIS and another five minutes to take a closer look at a higher resolution and sort the resulting selection, making it 40 minutes of total time to find suitable images for each of the four scenes. For this example eight scenes have actually been investigated which are only partially illustrated in Figure 4.3. A total of 90 minutes was needed for the resulting selection due to second lookups and comparisons of similarly suitable images. Thus, extrapolated to the whole set of 16 scenes it would easily take 180 minutes to find the most suitable satellite images for the defined region.
For the automated image selection approach (i) it takes approximately one minute to enter the correct variables. The runtime for the setting described above was 7 minutes. This makes a total of 8 minutes for approach (i) which is using an NDSI mask for every image as reference to calculate its cloud-score. As mentioned the computation process of the approach using the RGI polygons (ii) timed out and could not complete. At the time of writing the export functionality is limited to two hours which is why it stays unclear for how long the

export would have taken to complete. Approach (iii) took three minutes to complete and output its image selection results. Compared with one another it takes a lot longer doing it the manual way. At the same time, again, the overall accuracy of the manual scene selection might be arguably better compared to the automatic processing.

4.2 Area calculation and glacier extent

4.2.1 Algorithm 2.1: Area calculation

Comparison of thresholds to create glacier masks

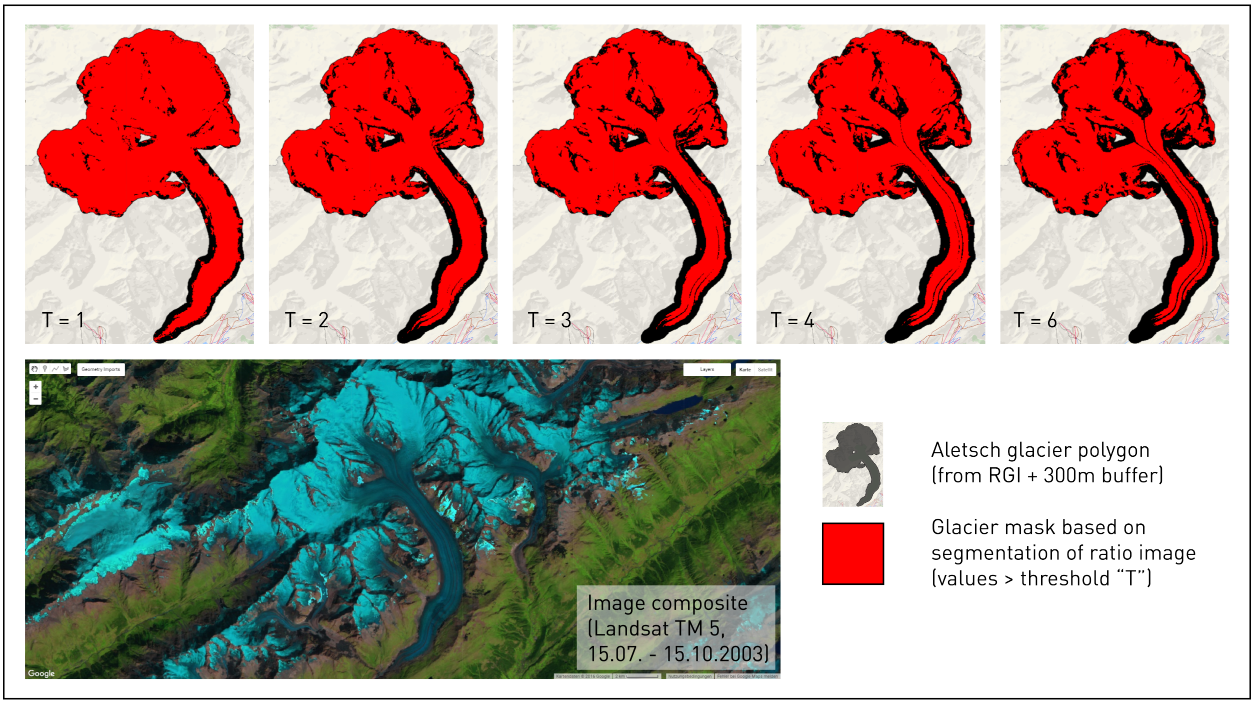


Figure 4.4: Comparison of various thresholds used to create glacier masks based on a ratio image of band 3 and 5 of of the image composite of Landsat TM 5 scenes recorded during the summer of 2003. Glacier masks are clipped to the RGI polygon of *Aletschgletscher* including a buffer zone of 300m. The higher threshold *T* is set, the less pixels are part of the glacier mask.

In Figure 4.4 the creation of various glacier masks based on different thresholds for a segmentated ratio image are shown. The ratio image is calculated using the red and the near infrared (NIR) band:

$$\frac{red}{NIR} \quad \Leftrightarrow \quad e.g. \frac{band\ 3}{band\ 5} \quad for\ Landsat\ 5\ Thematic\ Mapper$$

The glacier mask itself consists of values above the respective threshold. With *T=1* the glacier mask almost expands to the full extent of the buffered RGI polygon. The higher

the threshold T is set, the less pixels are included in the glacier mask. A large part of the glacier tongue and even some moraines with debris cover are already excluded with T=*2* and are excluded more and more for larger thresholds. For example, with T=*6* the medial moraines are continuously ignored. Especially in the lower parts of the glacier a lot of the (likely debris-covered) glacierized area seems to be left out.
Ideally, these thresholds should be set individually for each scene as the results can vary from image to image, but a dynamic setting of these thresholds has not been pursued here. Instead, the same threshold of 2.5 has been chosen for each segmented ratio image.

Results of area calculation for Central Europe and Scandinavia

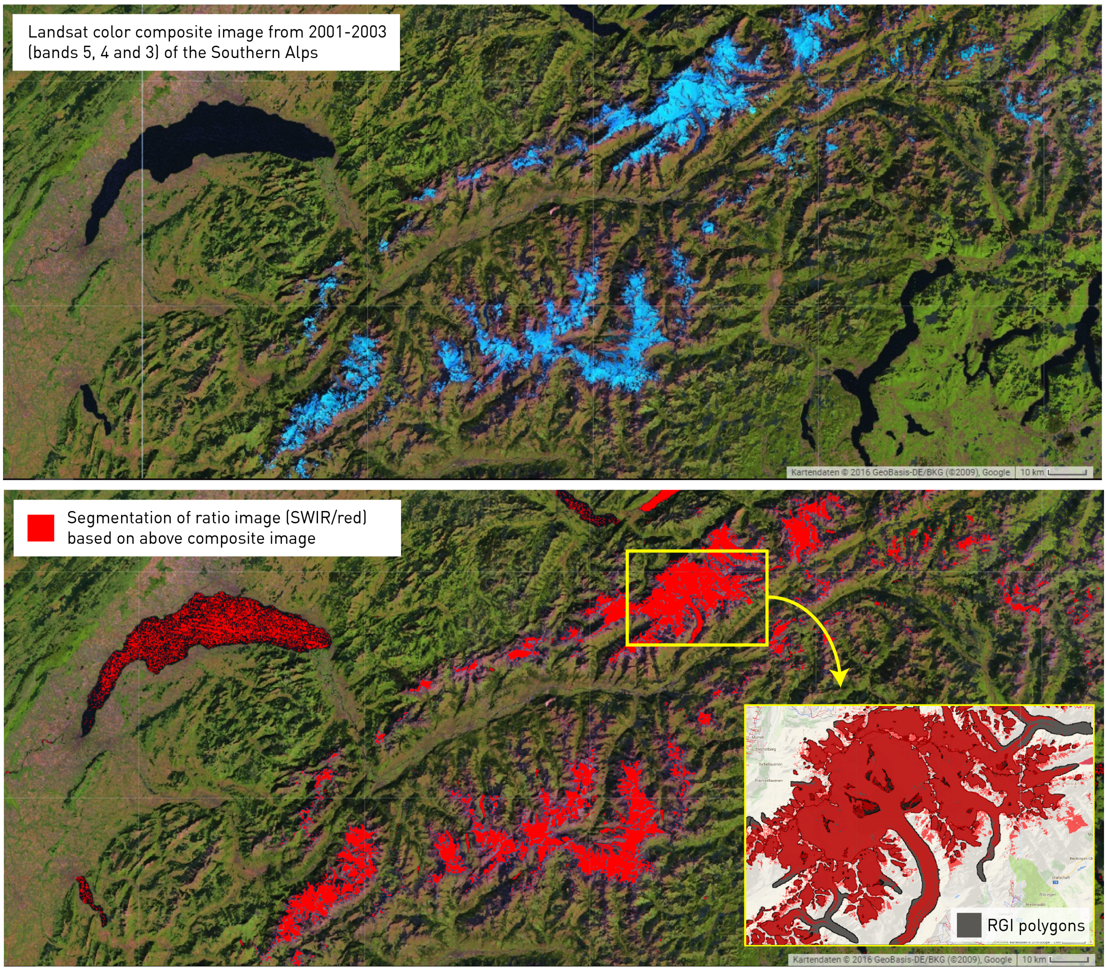

Landsat color composite image from 2001-2003 (bands 5, 4 and 3) of the Southern Alps
Kartendaten © 2016 GeoBasis-DE/BKG (©2009), Google 10 km
🟥 Segmentation of ratio image (SWIR/red) based on above composite image
RGI polygons
Kartendaten © 2016 GeoBasis-DE/BKG (©2009), Google 10 km

Figure 4.5: Example of the segmentation of a ratio image in the Southern Alps based on raw digital numbers (DN). A threshold of 2.5 has been applied to separate non-glacierized (below) and glacierized (above) pixels. Pixels above that threshold lying within the respective RGI polygons are included in the calculation of the glacierized area.

In Figure 4.5 the segmentation of a ratio image based on the short wave infrared (SWIR) and the red band of a Landsat image composite from 2001 to 2003 is exemplified. In addition to the actual glacierized area, lakes and rivers are usually classified as *glacierized* by mistake as their spectral behaviour is similar to the one of ice and snow. In this case this does not pose a problem as the ratio is intersected with the imported RGI polygons before to calculate the area within each polygon. Hence, water bodies generally do not affect the result but require special consideration in case of glaciers terminating in lakes.

In Table 4.1 one can compare the area calculated within GEE with the area which is contained as attribute for each imported RGI polygon. The area calculation is based on an image composite created from images from 1999 to 2003. The specific year, in which these area measurements have been taken, lies within a wide time range (Central Europe data covers the 1980s until today). However, around 95% of the data has been gathered around the early 2000s. The values of the last column (*delta [%]*) indicate that deviations are in a reasonable range. However, in case of very small glaciers these relative differences rise.

Comparison of small and big glaciers

Table 4.2 shows the means of the relative differences of various subgroups sorted by glacier size. The figures indicate, that the larger the area of the glaciers get, the lower is their measure of relative difference (see Table 4.1). An interpretation could be, that the area calculation of small glaciers is likely to be overestimated. However, this might only be a relative effect as the area grows squared, whereas the perimeter grows about linearly with increasing area. Thus, the error of overestimation is relatively smaller for large glaciers.
Figures 4.6 and 4.7 show the progression of the deviation value between the calculated area within GEE and the value attached to the polygon of the RGI inventory for Central Europe and Scandinavia. In both cases a similar effect can be seen. It seems that, most likely, the area of small glaciers is overestimated by the algorithm within GEE. This is probably due to the buffer zones, which are applied in order not to limit the classification process to the outlines of the RGI only. This buffer zone around the imported polygons enlarge the area of small glaciers in a stronger way than the buffering does for large glaciers due to the relative error propagation in regard to perimeter and area.

4.2.2 Algorithm 2.2: Semi-automatic method with additional manual post-processing in conventional GIS

Global analysis

A global application was not yet possible due to limitations in the functionality of GEE for such intense computations. At the time of writing renderings and illustration within the GEE API time out after five minutes. However, re-running the same code can help to overcome this issue as the servers can use cached calculations. Like this, calculations can still be rendered to the API using a few extra loops. Another possibility is to export the results to Google Drive as limitations are set at longer time durations. Hence, export queries hit time out messages only after two hours. Again, re-running the same code will help to overcome this limitation but, nevertheless, it remains unclear at which level the preceding

RGI_Name	RGI_Area	GEE_Area	delta [%]
Aletsch	82.20	84.74	+3.1
Gorner	56.46	59.69	+5.7
Fiescher	31.33	31.97	+2.0
Pasterze	17.77	17.51	-1.4
Rhone	15.83	17.01	+7.5
Argentiere	11.73	12.62	+9.4
Vernagtferner	8.55	9.36	+1.5
Hintereisferner	8.03	8.15	+1.5
Gries	5.29	5.84	+10.4
Kesselwandferner	3.96	4.31	+8.8
Jamtalferner	3.82	4.21	+10.2
Silvretta	2.88	3.10	+7.8
Saint Sorlin	2.87	3.15	+9.9
Careser	2.83	2.73	-3.4
...	...	...	...
Standard Deviation (total):			77.42

Table 4.1: Small selection of the area calculation resulting from Algorithm 2.1 for the RGI region of Central Europe. The middle columns contain the area measurement provided for the RGI inventory (*GEE_Area*). Additionally, it contains the area which was calculated within GEE based on an image composite from the images of the years 1999 to 2003 (*RGI_Area*). The last column indicates the measure of the relative difference (*delta*) between *GEE_Area* and *RGI_Area* in percent.

	mean of delta [%]	number of glaciers
total glaciers	+54.82	3980
glaciers > 0.5 km^2	+11.05	681
glaciers > 1 km^2	+7.30	407
glaciers > 5 km^2	+3.02	78
glaciers > 10 km^2	+0.13	27
glaciers > 15 km^2	-1.94	13
glaciers < 0.5 km^2	+63.65	3298

Table 4.2: Means of the relative difference (*delta*) of various subgroups sorted by glacier size. The larger the area of the glaciers, the lower their measure of relative difference and vice versa.

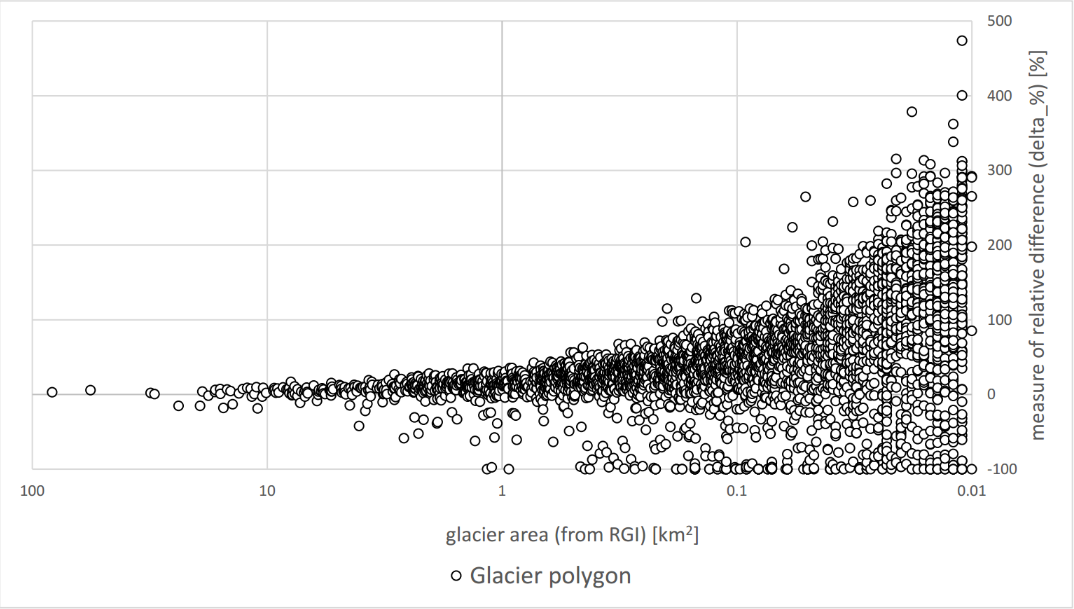


Figure 4.6: For the glaciers of Central Europe: progression of the measure of relative difference (delta_%) between the area calculations from within GEE and the RGI polygon.

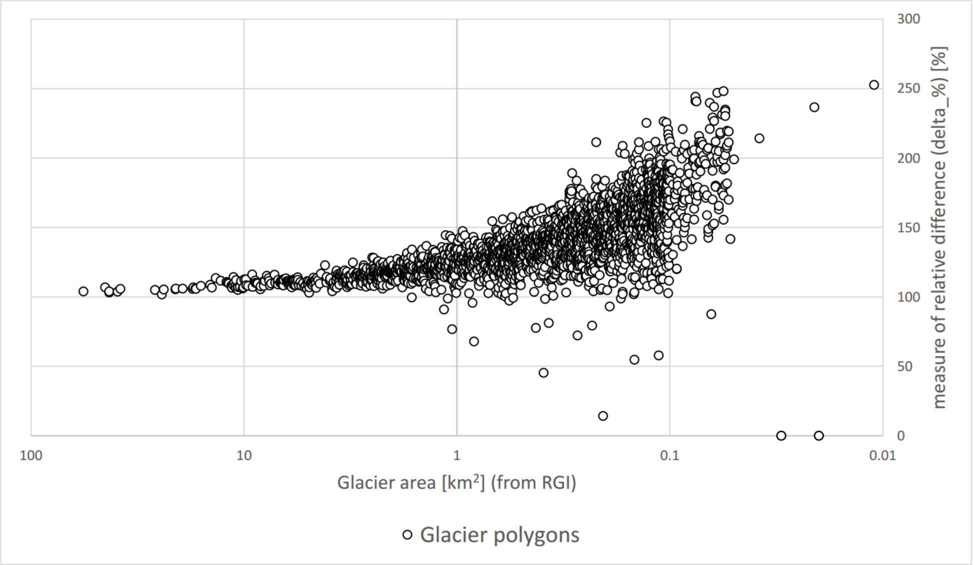


Figure 4.7: For the glaciers of Scandinavia: progression of the measure of relative difference (delta_%) between the area calculations from within GEE and the RGI polygon.

query timed out. Therefore, an application of the stated methods in Chapter 3 on a global level has not yet been done. However, some results carried out on large regional scales are presented in the following subsection.

Regional analysis

Several segmentations of ratio images have been done in order to create masks of glacierized area within GEE, export the result and display it in QGIS. The masks have been exported for several RGI regions (see Section 2.1.2). However, only Central Europe (Figure 4.8) and Iceland (Figure 4.9) are displayed here. For large RGI regions (such as Greenland, Alaska or the Antarctic Subantarctic) exports timed out even when using pixel resolutions of 60 m and 90 m instead of 30 m of the exported image. Working with even lower resolution wouldn't make sense as data accuracy decreases a lot using lower resolutions.

The results show that the glacier mask based on the respective image composite represents the glacierized area quite well. Especially the data for Central Europe shows good results in relation to the outlines of the RGI. However, in regard to the quality requirements of a glacier inventory the automatic creation of a glacier mask definitively needs some post-processing as it does not yet represent the actual outlines of the glacierized area in a flawless way. As predicted, especially water bodies need to be removed accordingly, such as rivers, lakes and parts of the sea which are falsely classified to represent glacierized surface. The latter is important particularly when dealing with glaciers reaching into water, which is the case for many glaciers in Scandinavia or Greenland (denoted with a) in Figures 4.8 and 4.9). In addition to that, debris covered parts of the glaciers are often not included if the debris cover is thick enough to hide spectral information underneath (denoted with b) in Figures 4.8 and 4.9).

4.3 Algorithm 3: Using the Otsu thresholding function to map snow cover and approximate the accumulation area ratio (AAR)

4.3.1 Use of 1-year image composites

Figure 4.10 presents the results of creating snow- and ice masks using the Otsu thresholding function (Ostu 1979) described in detail in Section 3.4. These masks are based on segmented ratio images from satellite scenes acquired during the late summer period of one specific year. The same multi image selection code as in Algorithm 1.2 has been applied (see Section 3.2).The resulting masks from *Findelen* in year 2000, 2007 and 2010 show good results in regard to distribution of snow- and ice pixels with a low amount of ice patches in the upper parts of the glacier and a low amount of snow pixels in the lower parts respectively. This could suggest a good depiction of accumulation- (snow pixels) and ablation area (ice pixels). Nonetheless, as the years 2005, 2006, 2008 and 2011 clearly show, chances of getting striping

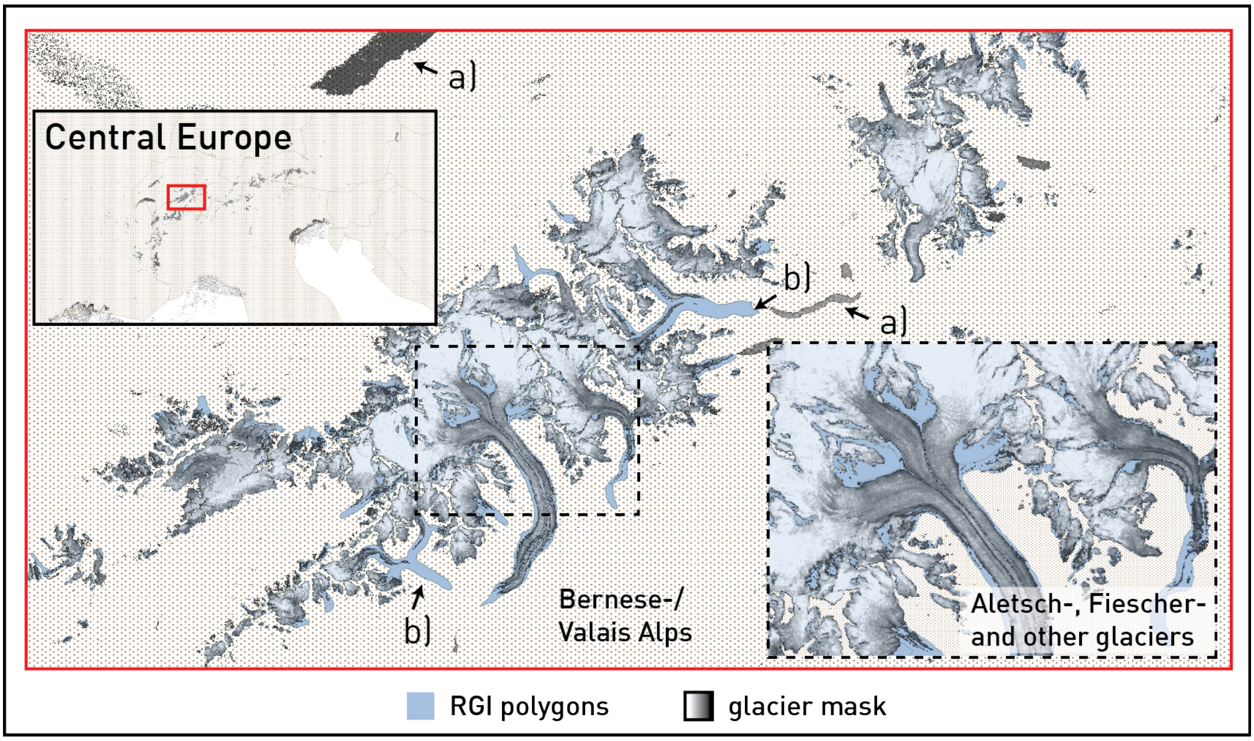


Figure 4.8: Ratio mask created within and exported from GEE of Central Europe with 30m pixel resolution, displayed in QGIS. The RGI polygon dataset is shaded in a light blue whereas the glacier mask is displayed in grey tones. a) Water bodies are falsely classified and often included in the glacier mask. b) Parts which are debris-covered are not detected as the underlying spectral information of the glacierized area is hidden.

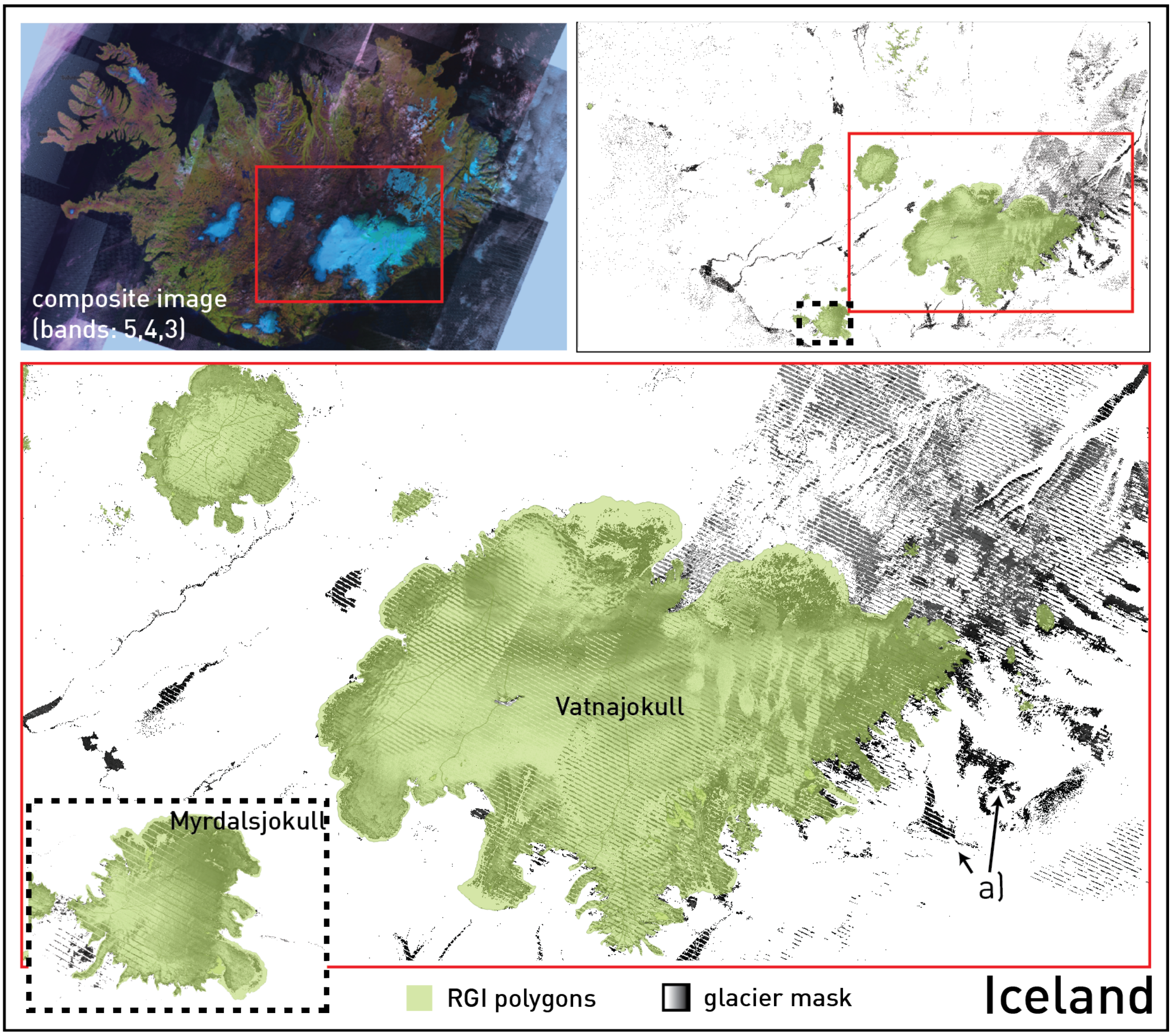


Figure 4.9: Ratio mask created within and exported from GEE of Iceland with 30m pixel resolution, displayed in QGIS. The RGI polygon dataset is shaded in a light green whereas the glacier mask is displayed in grey tones.

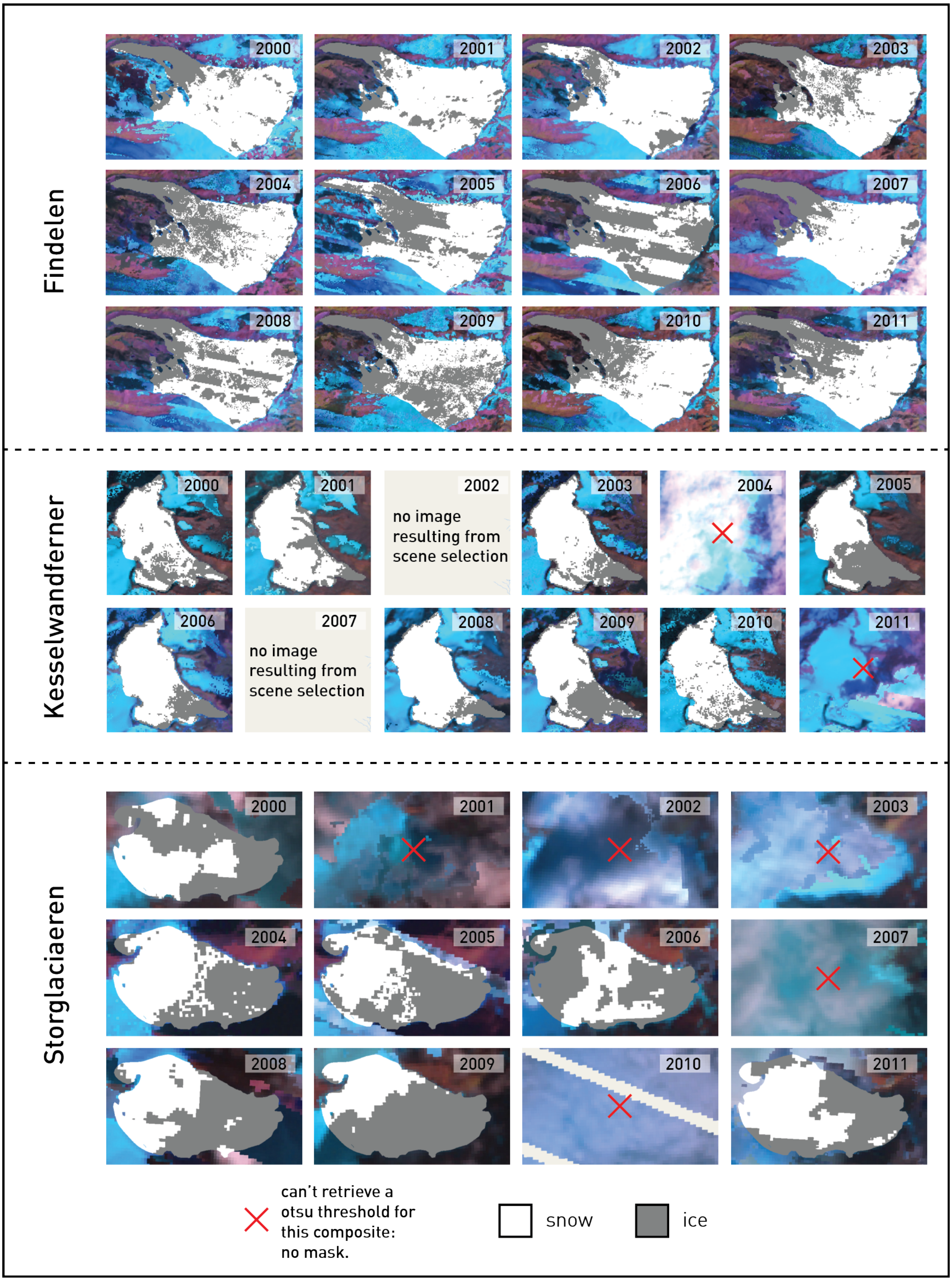


Figure 4.10: Creation of snow- and ice masks for three different glaciers using the Otsu thresholding function with a 30 m resolution. All the snow- and ice masks are based on image composites of a specific year.

pattern within an image (due to the missing scan line corrector of the Landsat 7 satellite) are relatively high. The same results are found in the 2005, 2008 and 2010 composite images for *Storglaciaeren*. However, there are no such striping effects visible in the composite images of *Kesselwandferner*. In some cases this causes the approximated AAR to be too low as the patches in the accumulation area classified as ice turn out to be rather large (see images 2005, 2006, 2008 for *Findelen*). For those composite images flawed by the missing scan line corrector, the method of mapping snow and ice cover is prone to errors and does not represent best practice. A method to compute and retrieve the height of an end-of-summer snowline (Callegari et al. 2016) would probably represent the conditions of a glacier more accurately, as misclassification of a large amount of pixels would not strongly affect the final result. Either way, creating an algorithm adapting to such matters is not a simple task.

Furthermore, there are two causes when no mask can be computed: First, in some years there are no composite images resulting from the scene selection at all (see 2002 and 2007 of *Kesselwandferner*), which can happen if for a specific year no image data could be found at this specific spot or if, more likely, the scenes within the analysed time period were all discarded due to heavy cloud cover. Second, if the image composite quality is low, the image appears blurred at the location of the glacier (mainly due to cloud cover). If that's the case, no optimal threshold can be found by the Otsu method, because no maximum *Between-Class-Variance* could be retrieved for the specific composite image.

Due to the flawed quality in quite a few of the resulting image composites it was decided to extend the time period to two following years to increase the set of images resulting from the respective scene selection. By doing so, the quality of the image composites and consequently also the quality of the snow- and ice masks should be enhanced. Although this is not entirely useful in glaciological terms, because of the fact that the AAR can vary a lot between two years, the results might still be reasonable for a trend analysis.

4.3.2 Use of 2-year image composites

The results of the approximated AAR based on 2-years image composites are shown in Figure 4.11. In contrary to Figure 4.10 the masks are based on a set of images taken within two years as with a greater time span the chance is bigger that there are some suitable images to create a best possible image composite. As predicted, these results do not show any case where no image composite could be created and only one single case where the Otsu method couldn't retrieve a threshold to compute the snow- and ice masks. In addition to that, there are less striping effects of the missing scan line corrector in Landsat 7. For *Findelen* there is visible evidence to that, whereas the composites and resulting masks of *Kesselwandferner* and *Storglaciaeren* from 2002/03 and 2006/07 respectively show a striping pattern. Some masks still show inconsistent classification in comparison to those in Figure 4.10. This is the case when the image composite is based on images which have been taken under conditions unsuitable for glacier mapping. This does not stop the algorithm to create an image composite and results in an unclear classification between ice and snow. Additionally, some composite images might still contain seasonal snow on parts of the ablation area resulting in a overestimation of the AAR. An extreme example would be the masks

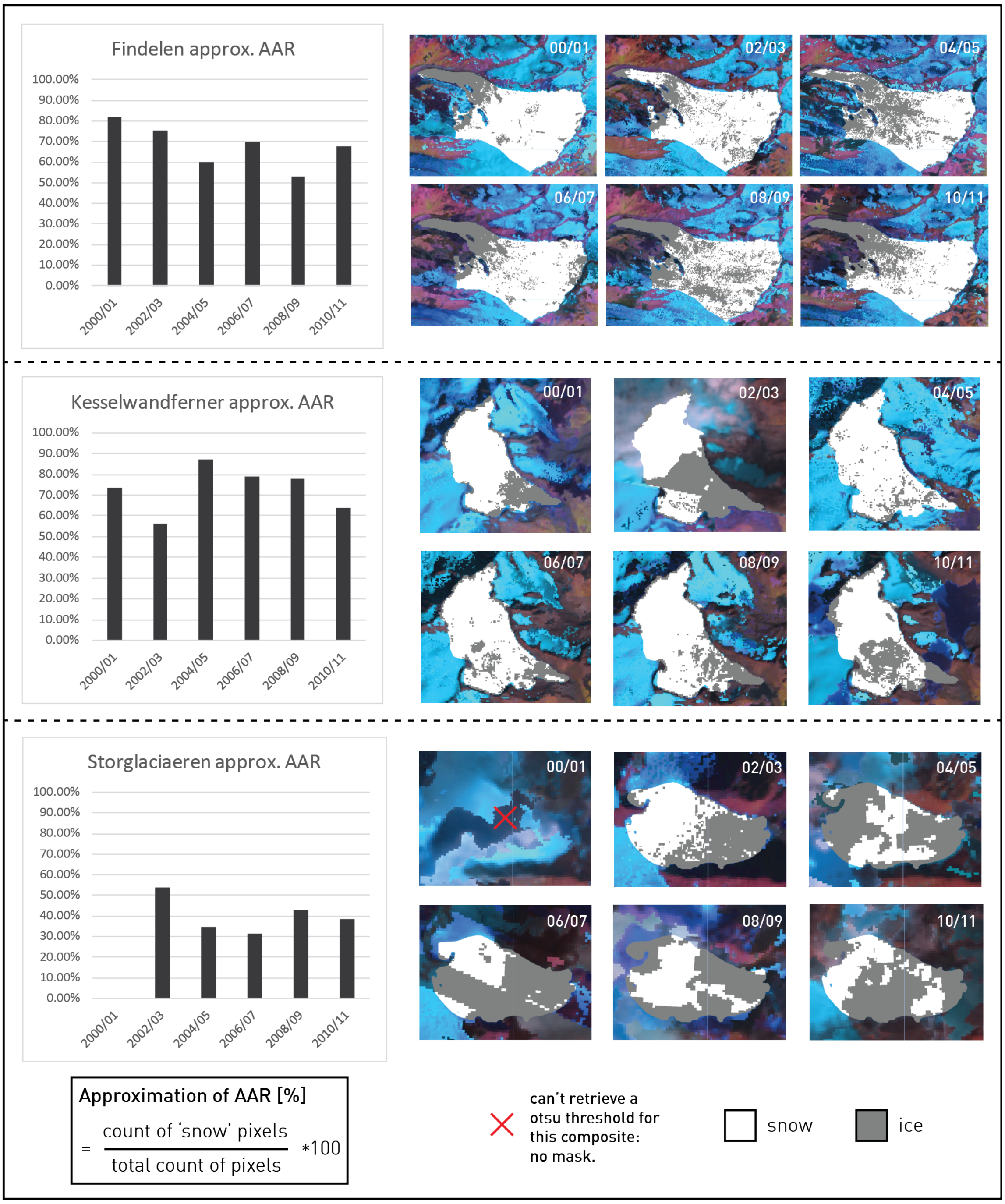


Figure 4.11: Creation of snow- and ice masks of three different glaciers using the Otsu thresholding function with a 30 m resolution. All the snow- and ice masks are based on image composites of two following years. The ratio between the count of snow pixels and the total count of pixels of both classifications represent the approximation of the AAR.

of *Kesselwandferner* in 2004/05 where almost 90% of the area is classified as snow. However, the masks from 2000/2001 for *Kesselwandferner* and *Findelen* show examples of rather consistent classification.

4.3.3 Trend comparison

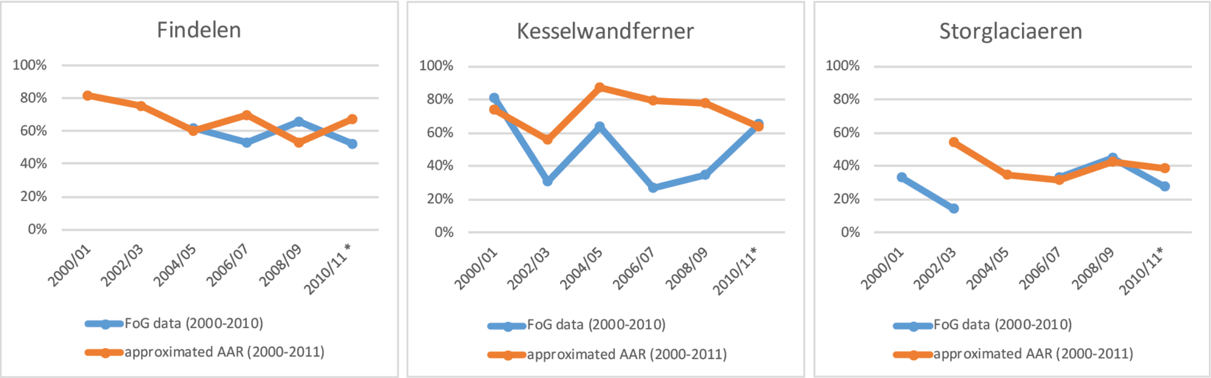
In order to calculate the actual AAR for a glacier one would need to know the exact outlines of the accumulation- and the ablation area. Still, one could say that the approximated AAR computed in Algorithm 3, by mapping snow and ice cover respectively, should follow the same trend as the real AAR. Assuming the data within FoG (WGMS 2008, 2012) are accurate, the trend comparison in Figure 4.12 shows the data in several diagrams for a 12-year period from 2000 to 2011. It has to be noted, that FoG data is not yet published for the year of 2011, which is why 2010 acts as reference value for the period of 2010 to 2011. Unfortunately, there are missing data series for *Findelen* from 2001 to 2004 and for *Storglaciaeren* from 2004 to 2005. Due to the fact that the composite images are based on a 2-year period, the changes do not turn out to be that drastic as the actual AAR measures do. Apart from that, the measures roughly appear to be in the same region. One of the mutual curve progressions is the minimum point of the *Kesselwandferner* in the years of 2002 and 2003 which is likewise represented in the approximated AAR with a local minimum for the same period and a following rise of the AAR value. Another similarity can be detected for *Storglaciaeren* with a slight increasing and a subsequent decreasing AAR in the last three points in time. However, the latter might be accidental when looking at the classification results of Figure 4.12 as the ice- and snow masks respectively are not delineated properly with pixel clusters from both masks over the entire area of the glacier.

4.4 Separation of clean- and debris covered ice

Crucial element of a completely automated Algorithm 4 is again the preceding image selection. It needs to perform well in order to create best possible image composites on which the classification of clean- and debris-covered ice is based on. In this particular example (see Figure 4.13) the selection is additionally based on a time range where images are known to be suitable for glacier mapping. For example, during the hot summer period in 2003 the Swiss Alps were highly exposed with the least possible seasonal snow- and cloud cover in years.

Conditions for the region in Central- and West Asia however, are not that good throughout the entire area but show similar results. The Karakoram mountains are well-known for their large areas of debris-covered glaciers (Herreid et al. 2015), which is why the yellow areas are distributed more extensively than in the Swiss Alps. In addition to that, in some regions the stripe pattern is even visible in the debris mask which is due to the failure of the scan line corrector for all ETM+ scenes after 2003. This can lead to patchy classifications for debris-covered glacier ice and clean ice.

In order to create a neat debris mask it was important to exclude the rock outcrops (visible exposure of bedrock, sometimes occurring within the perimeter of a glacier) before importing



Years	FoG AAR (2000-2010)	approximated AAR (2000-2011)
2000/01		82%
2002/03		76%
2004/05	62%	60%
2006/07	53%	70%
2008/09	66%	53%
2010/11*	52%	68%

Years	FoG AAR (2000-2010)	approximated AAR (2000-2011)
2000/01	81%	74%
2002/03	31%	56%
2004/05	64%	87%
2006/07	27%	79%
2008/09	35%	78%
2010/11*	65%	64%

Years	FoG AAR (2000-2010)	approximated AAR (2000-2011)
2000/01	34%	
2002/03	15%	54%
2004/05		35%
2006/07	34%	31%
2008/09	45%	43%
2010/11*	28%	39%

Figure 4.12: Comparison of the AAR values based on the data series from *Fluctuations of Glaciers* (FoG) and the approximation of AAR by mapping snow- and ice cover within GEE. *Note: As 2011 FoG data is not yet published the reference value is of 2010.

the RGI polygons into GEE. The conversion tool *Shpescape* sometimes did consider the rock outcrops to be part of the glacierized area although they are signified separately within the RGI dataset. Eventually, the rock outcrops should not be included in the debris mask, which should exclusively represent the deris-covered part of the glacierized area. Including these outcrops would result in an overestimation of the debris-covered parts.

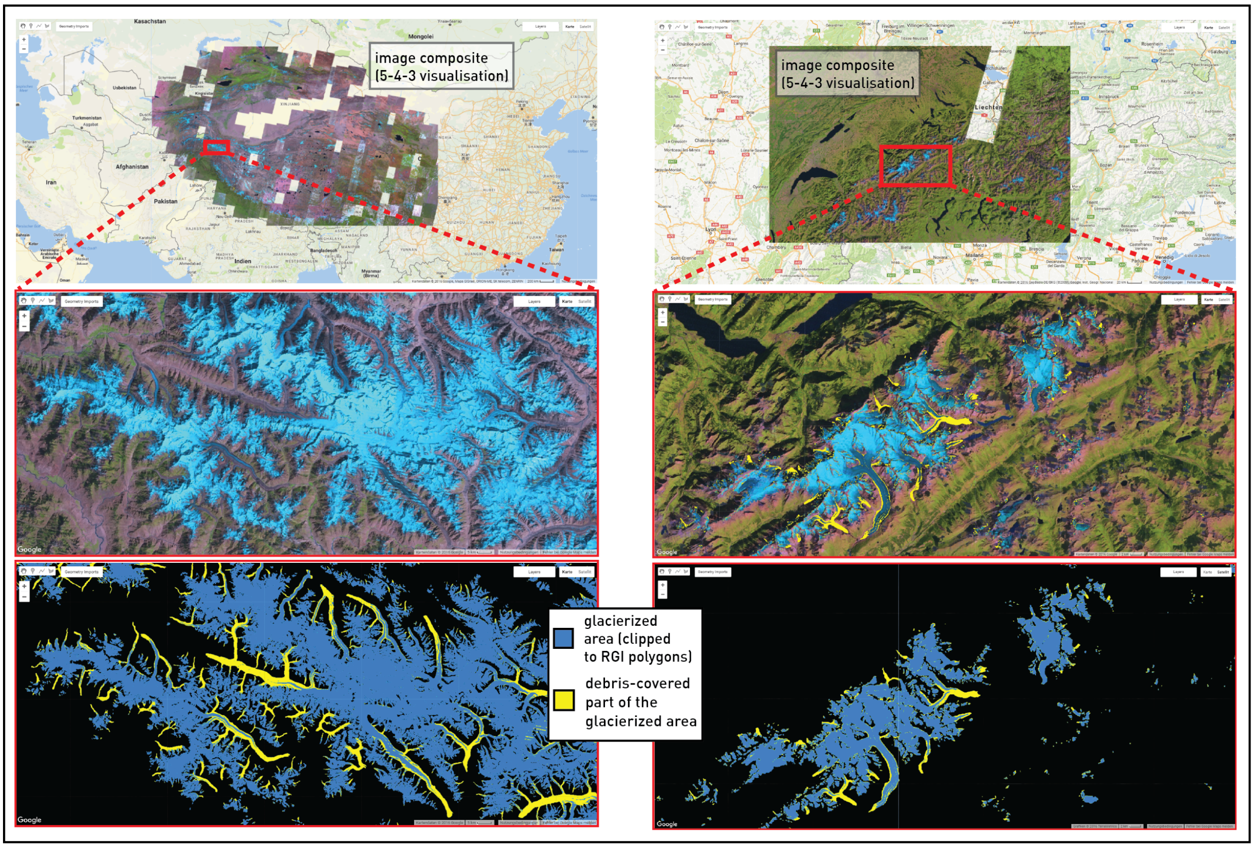


Figure 4.13: Separation of clean- and debris-covered ice based on an image composite of Landsat 5 + 7 images for the years 2002 to 2003 (left, Central- and West Asia) and 2003 (right, Swiss Alps) respectively. Top: complete image composite. Middle: image section of area of analysis. Bottom: corresponding debris-covered part of glacierized area.

5 Discussion

5.1 General suitability of GEE for glacier mapping

General evaluation of functionalities within GEE

To create automatic processes within GEE it is important to understand which processes are time consuming and which are not. By the time of writing it is not possible to monitor ongoing computation processes within GEE although it would simplify the process of building algorithms and keeping them efficient. Despite this missing functionality it is possible to get approximate measures for the computational effort of the individual processes by manually computing them within GEE. The following list displays the relative computation effort of functions used in this thesis (the slowest first):

- Scene selection process
  - minimizing seasonal snow (*snowcount* & subsetting)
  - minimizing cloud cover (apply cloud-score)
- Calculating with raster- and vector data simultaneously (e.g. intersections)
  - buffering of RGI polygons (effort increases with more vertices per polygon)
  - clipping images to polygons
  - area calculation (pixel count within polygons)
- Creation of image composites
- Mathematical operations with images (e.g. creation of segmented ratio images)

In this thesis, clearly the scene selection process takes the most effort to finish as it can include a large amount of datasets in the beginning and filtering the collection of images by rating every image with a certain suitability (which includes the examination of plenty of individual pixels depending on the chosen scale) is a slow process. Compared to that, the creation of a `qualityImage` and the ratio image is much faster. However, whenever a buffer is created or an image needs to be clipped to a complex polygon for further processing, computation time increases significantly. This is not very surprising as the combination of raster and vector data usually calls for complex computations. Of course, the measures stated above depend on the size of the region of analysis, how many images have to be taken into account, the range of the time period or the complexity of the polygons to name just a few important variables for a given setting.

5.2 Algorithm 1: Scene selection

The automatic image selection algorithms (Algorithm 1.1 & 1.2) show satisfying results in relation to the manual image selection. Most of the manually selected images to be found in the automatic selection result with some exceptions if cloud and snow cover meet certain conditions within a processed image. The different approaches (see i), ii) and iii) in Figures 3.1 & 3.2) are used to determine the perimeter on which computations of the minimization of cloud cover and seasonal snow are based on. Each of them leads to different results in regard to the respective image selection and the needed computation time. Moreover, the choice between the two approaches for the classification and count of snow cover (see j and jj) allows to additionally adapt to the given set of conditions.

Approaches to define the perimeter of computations

Limiting the computations to the area of a pre-defined NDSI mask (i) proves to be a fast approach with the flaw of sometimes underestimating the amount of seasonal snow in a respective image due to high cloud cover around the actual glacierized area. If the ultimate aim is to automatically retrieve the best possible image for glacier mapping this will not be the best possible approach to use. However, if the aim is to make a pre-selection of suitable images of a wide region to create an image composite out of it, this approach can be very swift and useful.
Limiting the computations to the area within imported RGI polygons (ii) performs better regarding the resulting sorting of suitable images for glacier mapping. However, it is important to note that - compared to the other approaches - it takes extremely long to perform the calculations within the complex perimeter of the RGI polygons. Therefore, the applicability of this approach is rather low. Nonetheless, a pre-processing of this vector dataset or even its conversion to a raster dataset could show promising results in regard to processing time.
At last, by limiting the computations to the area within a manually drawn polygon around the assumed glacierized area (iii) results have proven to be very sound without needing too much processing time. However, the application of this approach needs some extra work in advance to delineate the perimeter of the test region and some experience with GEE, in particular when various delineations need to be merged and the size of the area of interest is large.

Approaches to classify and count seasonal snow cover

In relation to the minimization of seasonal snow and the required functionality to classify and count snow pixels, the pros and cons of two approaches (see j) & jj)in figures 3.1 & 3.2) have to be balanced: whereas (j) counts all the pixels classified as snow and ice by creating an NDSI mask, (jj) the application of the Otsu method (Ostu 1979) to separate snow from ice allows to specifically count the pixels classified as snow and ignore the ones classified as ice. Therefore, compared to the count of pixels classified by NDSI mask it gives a different result and should be the approach to go for when only snow cover should be mapped. Nevertheless, to be able to use the advantage of the Otsu method the outlines of a glacier need to be available as a classification of snow and ice is not meaningful if too many pixels of other terrain are included in the process. Eventually, using the Otsu method to perform a minimization of seasonal snow does not lead to significantly better results than doing it with the much simpler and faster creation of an NDSI mask. With the usage limits in mind, the Otsu method is not applicable for large datasets unless the spatial or temporal resolution or the size of region to assess is reduced. Additionally, regarding the method of automated scene selection presented before, the different *snowcounts* do not necessarily need to represent the truth but rather display and stay in the same relation to each other. In fact, the consistency and the low error rate of the used approach is more important. Hence, in terms of applicability the approach of creating an NDSI mask to delineate the perimeter of computations (j) represents best practice.

In respect of a global assessment, the results of the image selection algorithms suggest that, wherever applicable, the approach based on a manually drawn geometry (iii) works best. It additionally includes a flexibility to adapt to ones initial research setting as a manually drawn geometry can range from a rudimentary sketched polygon to more complex delineation if favoured and possible at all in relation to the size of the area of analysis and the computation time within GEE that comes with it. However, if the area of interest is rather small and the amount of satellite images computations is rather low (e.g. 2'000 km^2 for a period of 4 months within 3 following years), using the Otsu method in combination with the original RGI polygons is applicable without any problems.

Overall, the automatic scene selection algorithm performs well in relation to the fact that it automatically and successfully filters those scenes which are not suitable for glacier mapping. In regard to the functionalities of Algorithm 2, 3 and 4 this is very useful as they all inherently base on image composites where it is crucial to have a whole collection of images to chose from. However, the automatic scene selection algorithm is not yet sophisticated enough to evaluate the true goodness of a respective scene. To a greater degree it is only capable of interrelating various scenes and filter a collection of the most suitable ones.

5.3 Algorithm 2: Area calculation and glacier extent

Effects of prior scene selection

As Algorithm 2.1 and 2.2 include a prior scene selection they heavily depend on a successful outcome of it. Normally it is possible to include a sufficient amount of images to create an image composite. Nonetheless, this does not mean that it is based on high quality images. A successful outcome of the prior scene selection thus is given if it not only contains the least worst but the best possible images. Therefore, any knowledge needs to be used in the preparation of a query about the period in which a specific region might have been recorded under suitable conditions in terms of glacier mapping (for example, the knowledge of the near perfect conditions in the Swiss Alps in the late summer period in 2003). A guide on these period can be derived from the mass balance time series available at WGMS WGMS (2008). Years with very negative balances point to very good glacier mapping conditions.

Debris cover and shadows

The fact of not being able to correctly classify debris covered parts of glaciers only just made it possible to create a mask for a clean-ice glacier polygon. But nevertheless, missing this functionality definitely is one of the major downsides to manual glacier mapping and would pose a valuable feature to implement in the future as soon as a suitable method can be applied in an automated fashion. Based on this lack, area calculation suffers in regard to accuracy as debris will continuously be detected as *other terrain* and thus not be included in the classification of glacierized area.

In addition to that, the effects of shadows occurring due to the topography of a region or cloud shadows have not been taken into account here. This is also very relevant as glacierized area mostly appears in mountainous regions including steep slopes typically resulting in shady areas and frequent (also orographic) clouds. In contrast to the lack of detecting debris-covered glaciers, occurrence of shadow can vary strongly and also depends on the time of the year. Later in the year, shadows increase but seasonal snow cover decreases. The optimal time is thus a compromise between these two effects.

Effects of buffer zones around imported RGI polygons

In the results of Algorithm 2.1 a buffer zone of 100 m is applied consider possible area changes (in comparison with the respective RGI polygons) of advancing glaciers. While a buffer zone of several dozen metres might be applicable for most of the glaciers worldwide in this context, surging glaciers might exceed this distance within a single year will still not be represented correctly by this method. It is also important to note, that even with a buffer zone of only 100 m the area will be overestimated due to snow covered areas on the other side of a mountain's ridge. This can be corrected by the application of drainage divides which would specify a fixed glacier boundary at the mountain ridges. However, this is a post-processing step that has not been implemented in this thesis.

Additionally, the tendency of overestimating a glacier's area increases for small glaciers as the extra area of analysis added by the creation of a buffer zone is much larger for smaller glaciers in relative terms as it is for larger glaciers. Most likely, this might explain the results shown in figures 4.6 and 4.7 respectively. Considering that the vast majority of glaciers is below a few km^2 in total area, this overestimation of the area of small glaciers would result in large overestimations for a regional scale estimation of the glacierized area.

It is thus important to weigh if a buffer should be applied or not. One could say, that for a best possible estimate of the global (or even regional) extent of glacierized area the creation of a buffer zone should be spared. Moreover, the computation time decreases by not applying any buffer zones to the respective geometries. However, by doing so, one takes the risk to ignore any growth in glacierized area in comparison to the condition of the used glacier geometries although this might be a minor problem these days.

Effects of ETM+ scenes with missing scan line corrector

Due to the scan line corrector failure of Landsat 7 all ETM+ scenes acquired after May 2003 include missing data which display in line patterns of no data perpendicular to the flight direction of the satellite. Pixels missing in the corresponding scenes were not included in the computation. This has two implications for the different applications used here: First, in terms of automatic scene selection, Landsat 7 scenes with missing data most probably feature less pixels classified as *snow*. This results in a lower *snowcount* and thus these scenes are generally preferred by the automatic scene selection in comparison with for example Landsat 5 scenes. Second, it happens that when the image composite is created some pixels are missing at specific locations. This happens if at specific locations either the scene selection process does not pass any pixel as none of them is fulfilling the set of conditions or if there is no data to pass due to the failure of the scan line corrector. However, the latter can be avoided if the time range includes a reasonable amount of images.

5.4 Algorithm 3: Using the Otsu thresholding function to map snow cover and approximate the accumulation area ratio (AAR)

5.4.1 Image quality dependency

The differentiation between snow- and cloud cover using the implementation of the Otsu method in GEE performs well and results are reasonable. Although it has proven to be a time-consuming computation it can be applied without limitations for single glaciers or even smaller regions. When the mapped snow cover is used as an AAR approximation the Otsu method is only informative if a respective scene or image composite comes with least possible seasonal snow and without cloud cover or missing data (e.g. failure of scan line corrector in Landsat 7). As soon as an image has issues (clouds, gaps, shadows), the resulting AAR approximation gets imprecise. It is thus highly dependent on the foregoing scene selection, regardless of whether the selection follows automatically or knowledge based.
Moreover, from a methodical and glaciological point of view, 2-year image composites are not the best option to retrieve meaningful end of summer snow cover data. Nevertheless, for a trend analysis they proved to be less prone to errors than 1-year image composites as chances improve to compute fairly delimited ice- and snow masks if the number of images available for a region increases.

5.4.2 Applicability of trend analysis for specific regions

As a result of the observed dependencies, it is difficult to perform a trend comparison. Nevertheless, Algorithm 3 seems to technically do its job by providing accurate snow- and ice masks based on and strongly dependent on the input material it is given. It is thus clear that a complete accordance of the approximated AAR values and the measurements of the AAR published in the Fluctuations of Glaciers (FoG) series is given. Given the short computation time, Algorithm 3 can provide a first glimpse at a specific glacier or area in terms of AAR calculation but results remain an approximation. The AAR determination is thus a tool which should be further refined and offers the possibility to compute a fair estimate of values over large regions. The quality of the approximation should be evaluated visually as unreliable masks can quickly be recognised as such. Further improvements can be expected by using reflectance values that are corrected for topographic effects (variable local incidence angle of the sun) instead of the raw digital numbers.

5.5 Algorithm 4: Separation of clean- and debris-covered ice

The quality of the results depends on the previous scene selection and undoubtedly, the quality of the debris-masks varies due to changing conditions within an image. However, the separation of clean- and debris-covered ice of a glacier does perform well. With Algorithm 4 it is thus possible to provide debris masks by intersecting the glacier outlines retrieved from GEE with the outlines from the respective glacier inventory. The algorithm has proven to compute stable results even over a large region within a few minutes. In this particular application GEE is a valuable tool to map glacierized areas at large spatial scales.

The fact that debris cover highly affects the response of glaciers to climate change (Scherler et al. 2011), illustrates an additional application of this algorithm as it offers the opportunity to regularly map and measure the debris-covered parts of a specific glacierized region. Thus, the algorithm could help to detect a trend in debris cover within that region and hence support calculations of future water availability (Immerzeel et al. 2010). Debris cover also plays a relevant role for ablation processes by decreasing the melting rates of the ice beneath (Kayastha et al. 2000) it .

In addition to that, the algorithm could also serve to provide data for observation purposes and indicate major changes in debris cover such as from rock fall events.

5.6 Outlook and perspectives

Shadows related to topographic conditions and clouds have not been taken into account in this thesis. This could be implemented in a similar way as suggested by Selkowitz & Forster (2016) where the most deeply shadowed pixels were masked when surface reflectance in specific bands were below a certain threshold. Although this method does not distinguish between cloud- and topographic shadow, it could be a first step into including shadow into the glacier mapping process as it represents a simple and effective way of handling shadows and minimizing their effect on the resulting mapping process. Another approach to specifically deal with cloud shadows to reduce misclassification is demonstrated by a method called Temporal Dark Outlier Mask (TDOM) presented by a group from the United States Department of Agriculture (Housman et al. 2015). This algorithm identifies pixels which appear dark in infrared bands only for a specific point in time but not always appear dark when examining the same location within a series of images. Some dark pixels are then recognised as statistical outliers and thus most likely to be affected by cloud shadow.

In regard to the automatic scene selection the intersection between raster and vector data resulted in high computation efforts. A promising approach for the definition of the perimeter of computation would be to priorly smoothen the vector polygons, convert them into a raster format and vary the spatial resolution of the resulting rasterized polygon dataset. In combination with the shortly available functionality of GEE to track the different computation processes the applicability of the automated scene selection method could certainly be improved. In particular the reduced computation time would allow to increase the area covered.

6 Conclusion

The main objective of implementing a tool within GEE was to automate the image selection process and to retrieve the best suitable images for glacier mapping. This goal has been pursued by taking the most important measures for scene selection into account. The created algorithm is able to quickly filter a collection of thousands of images by a specific time period and to recognize and output those images with a minimum of cloud- and seasonal snow cover. Various automated approaches have been compared in terms of the quality of the resulting scene selection and the duration of the computation, whereas they differ in their process of designating the perimeter where cloud-score computation and seasonal snow calculation are applied. Results show that it is best to keep the computations as simple as possible by either using a manually outlined (approach iii) perimeter or use an NDSI mask (approach i) to define the perimeter of computations. Based on these findings, however, further improving this algorithm may result in more precise scene selections within even shorter computation times. As this is the base for most of the other applications described here it will be crucial to refine this algorithm in the future.

Despite the dependency on the prior scene selection, it was possible to create glacier outlines of good quality over a large spatial scale. Not least because the temporal horizon for creating glacier outlines can be set over a few years which leads to a larger collection of suitable images on which the final image composite is based on. Nevertheless, future applications should additionally deal with shadows within the individual scenes and the detection of debris-covered areas as they both illustrate sources of error and reduce the quality of the scene selection process and all subsequent computations. Additionally, the consequences of possibly used buffer zones should be addressed by an implementation of automatically derived drainage divides.

The implementation of an automated thresholding algorithm using the Otsu method (Ostu 1979) within GEE to classify snow and ice surfaces proved to provide good results even under the changing conditions of the scenes. However, achieving a good approximation of the AAR for a single glacier is highly dependent on the quality of the available images acquired near the end of the ablation period. An extension of the time period to two following years resulted in less computation errors and a more distinct classification of ablation and accumulation area but neglects the fact that AAR can vary a lot between two years. Wherever knowledge about the quality of a set of scenes for a specific region is available, it should be used to interpret or even improve the results.

The objective to complement already available glacier inventories with a mask of debris-covered regions was achieved by creating an image composite for the entire regions of West-Central-Asia and the Swiss Alps respectively. Even though the classification of *clean* ice and *debris-covered* ice is dependent on the prior scene selection and the resulting image composite, this application could provide promising results for a better understanding of ablation processes and their implications within a specific region.

Although the presented findings need further improvements, they display an important step towards automation processes in glacier monitoring and illustrate the capabilities of GEE as a computation environment.

Bibliography

Albert, T. H. (2002), 'Evaluation of Remote Sensing Techniques for Ice-Area Classification Applied to the Tropical Quelccaya Ice Cap, Peru', *Polar Geography* **26**(3), 210–226.

Barandun, M., Huss, M., Sold, L., Farinotti, D., Azisov, E., Salzmann, N., Usubaliev, R., Merkushkin, A. & Hoelzle, M. (2015), 'Re-analysis of seasonal mass balance at Abramov glacier 1968–2014', *Journal of Glaciology* **61**(230), 1103–1117.

Callegari, M., Carturan, L., Marin, C., Notarnicola, C., Rastner, P., Seppi, R. & Zucca, F. (2016), 'A Pol-SAR Analysis for Alpine Glacier Classification and Snowline Altitude Retrieval', *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing* **9**(7), 3106–3121.

Carturan, L., Filippi, R., Seppi, R., Gabrielli, P., Notarnicola, C., Bertoldi, L., Paul, F., Rastner, P., Cazorzi, F., Dinale, R. & Dalla Fontana, G. (2013), 'Area and volume loss of the glaciers in the Ortles-Cevedale group (Eastern Italian Alps): Controls and imbalance of the remaining glaciers', *The Cryosphere* **7**(5), 1339–1359.

Choi, H. & Bindschadler, R. (2004), 'Cloud detection in Landsat imagery of ice sheets using shadow matching technique and automatic normalized difference snow index threshold value decision', *Remote Sensing of Environment* **91**(2), 237–242.

De Angelis, H. (2014), 'Hypsometry and sensitivity of the mass balance to changes in equilibrium-line altitude: The case of the Southern Patagonia Icefield', *Journal of Glaciology* **60**(219), 14–28.

Dyurgerov, M., Meier, M. F. & Bahr, D. B. (2009), 'A new index of glacier area change: A tool for glacier monitoring', *Journal of Glaciology* **55**(192), 710–716.

Fischer, M., Huss, M., Barboux, C. & Hoelzle, M. (2014), 'The New Swiss Glacier Inventory SGI2010: Relevance of Using High-Resolution Source Data in Areas Dominated by Very Small Glaciers', *Arctic, Antarctic, and Alpine Research* **46**(4), 933–945.

Gardent, M., Rabatel, A., Dedieu, J.-P. & Deline, P. (2014), 'Multitemporal glacier inventory of the French Alps from the late 1960s to the late 2000s', *Global and Planetary Change* **120**, 24–37.

Google Earth Engine Team (n.d.), 'Google Earth Engine: A planetary-scale geo-spatial analysis platform'.
URL: *https://earthengine.google.com*

Greensted, A. (2010), 'Otsu Thresholding'.
URL: *http://www.labbookpages.co.uk/software/imgProc/otsuThreshold.html*

Herreid, S., Pellicciotti, F., Ayala, A., Chesnokova, A., Kienholz, C., Shea, J. & Shrestha, A. (2015), 'Satellite observations show no net change in the percentage of supraglacial debris-covered area in northern Pakistan from 1977 to 2014', *Journal of Glaciology* **61**(227), 524–536.

Housman, I., Tanpipat, V., Biswas, T., Clark, A., Stephen, P., Maus, P. & Megown (2015), Monitoring forest change in southeast Asia: case studies for USAID Lowering Emissions in Asia's Forests, PhD thesis, U.S. Department of Agriculture, Salt Lake City.

Immerzeel, W. W., van Beek, L. P. H. & Bierkens, M. F. P. (2010), 'Climate change will affect the Asian water towers', *Science (New York, N.Y.)* **328**(5984), 1382–1385.

IPCC, ed. (2013), *Climate Change 2013 The Physical Science Basis: Working Group I Contribution to the Fifth Assessment Report of the Intergovernmental Panel on Climate Change*, Cambridge University Press, Cambridge, United Kingdom and New York, NY, USA.

Irish, R. R. (2000), Landsat 7 automatic cloud cover assessment, *in* S. S. Shen & M. R. Descour, eds, 'AeroSense 2000', SPIE Proceedings, SPIE, p. 348.

Kargel, J. S., Leonard, G. J., Bishop, M. P., Kääb, A. & Raup, B. H., eds (2014), *Global Land Ice Measurements from Space*, Springer Praxis Books, Springer, Dordrecht.
URL: *http://gbv.eblib.com/patron/FullRecord.aspx?p=1106462*

Kaser, G., Cogley, J. G., Dyurgerov, M. B., Meier, M. F. & Ohmura, A. (2006), 'Mass balance of glaciers and ice caps: Consensus estimates for 1961–2004', *Geophysical Research Letters* **33**(19).
URL: *http://onlinelibrary.wiley.com/doi/10.1029/2006GL027511/full*

Kayastha, R. B., Takeuchi, Y., Ageta, Y. & Nakawo, M. (2000), 'Practical prediction of ice melting beneath various thickness of debris cover on Khumbu Glacier, Nepal, using a positive degree-day factor', *Hydrological Sciences Journal* pp. 71–81.

Livni, J. (2016), 'Shape Escape'.
URL: *http://shpescape.com/*

Ostu, N. (1979), 'A Threshold Selection Method from Gray-Level Histograms', *IEEE Transactions on Systems, Man, and Cybernetics* **9**(1), 62–66.

Pal, N. R. & Pal, S. K. (1993), 'A review on image segmentation techniques', *Pattern Recognition* **26**(9), 1277–1294.

Paul, F. (2000), 'Evaluation of different methods for glacier mapping using Landsat TM', *Proceedings of EARSeL-SIG-Workshop Land Ice and Snow* **1**(17), 239–245.

Paul, F. (2004), 'The New Swiss Glacier Inventory 2000: Application of Remote Sensing and GIS', *Schriftenreihe Physische Geographie Glaziologie und Geomorphodynamik* pp. 1–194.

Paul, F., Barry, R. G., Cogley, J. G., Frey, H., Haeberli, W., Ohmura, A., Ommanney, C., Raup, B., Rivera, A. & Zemp, M. (2009), 'Recommendations for the compilation of glacier inventory data from digital sources', *Annals of Glaciology* **50**(53), 119–126.

Paul, F., Huggel, C. & Kääb, A. (2004), 'Combining satellite multispectral image data and a digital elevation model for mapping debris-covered glaciers', *Remote Sensing of Environment* **89**(4), 510–518.

Paul, F. & Kääb, A. (2005), 'Perspectives on the production of a glacier inventory from multispectral satellite data in Arctic Canada: Cumberland Peninsula, Baffin Island', *Annals of Glaciology* **42**(1), 59–66.

Paul, F., Winsvold, S., Kääb, A., Nagler, T. & Schwaizer, G. (2016), 'Glacier Remote Sensing Using Sentinel-2. Part II: Mapping Glacier Extents and Surface Facies, and Comparison to Landsat 8', *Remote Sensing* **8**(7), 575.

Pfeffer, W. T., Arendt, A. A., Bliss, A., Bolch, T., Cogley, J. G., Gardner, A. S., Hagen, J.-O., Hock, R., Kaser, G., Kienholz, C., Miles, E. S., Moholdt, G., Mölg, N., Paul, F., Radić, V., Rastner, P., Raup, B. H., Rich, J. & Sharp, M. J. (2014), 'The Randolph Glacier Inventory: A globally complete inventory of glaciers', *Journal of Glaciology* **60**(221), 537–552.

Racoviteanu, A. E., Paul, F., Raup, B., Khalsa, S. J. S. & Armstrong, R. (2009), 'Challenges and recommendations in mapping of glacier parameters from space: Results of the 2008 Global Land Ice Measurements from Space (GLIMS) workshop, Boulder, Colorado, USA', *Annals of Glaciology* **50**(53), 53–69.

Randolph Glacier Inventory (2015), 'Randolph Glacier Inventory 5.0'.
URL: *http://www.glims.org/RGI/*

Robson, B., Hölbling, D., Nuth, C., Stozzi, T. & Dahl, S. (2016), 'Decadal Scale Changes in Glacier Area in the Hohe Tauern National Park (Austria) Determined by Object-Based Image Analysis', *Remote Sensing* **8**(1), 67.

Rott, H. & Markl, G. (1989), 'Improved snow and glacier monitoring by the Landsat Thematic Mapper', *Proceedings of a workshop on Landsat Thematic Mapper applications* (ESA SP - 1102), 3–12.

Scherler, D., Bookhagen, B. & Strecker, M. R. (2011), 'Spatially variable response of Himalayan glaciers to climate change affected by debris cover', *Nature Geoscience* **4**(3), 156–159.

Selkowitz, D. & Forster, R. (2016), 'An Automated Approach for Mapping Persistent Ice and Snow Cover over High Latitude Regions', *Remote Sensing* **8**(1), 16.

U.S. Geological Survey (2005), 'Global Visualization (GloVis) Viewer'.
URL: *http://glovis.usgs.gov/*

U.S. Geological Survey (2008), 'Imagery for Everyone: Timeline Set to Release Entire USGS Landsat Archive at No Charge'.
URL: *http://landsat.usgs.gov/products_data_at_no_charge.php*

WGMS (1989), 'World glacier inventory: Status 1988. Haeberli, W.; Bösch, H.; Scherler, Dirk; Østrem, C.; Wallén, C. C. (eds.), IAHS (ICSI) / UNEP / UNESCO', *World Glacier Monitoring Service, Zurich, Switzerland* p. 458.

WGMS (2008), 'Fluctuations of Glaciers 2000-2005: Volume IX. Haeberli, W.; Zemp, M.; Kääb, Andreas; Paul, F.; Hoelzle, M. (eds.), ICSU(FAGS)/IUGG(IACS)/UNEP/UNESCO/WMO', *World Glacier Monitoring Service, Zurich, Switzerland* p. 285.

WGMS (2012), 'Fluctuations of Glaciers 2005–2010: Volume X. Zemp, M.; Frey, H.; Gärtner-Roer, I.; Nussbaumer, S. U.; Hoelzle, M.; Paul, F.; Haeberli, W. (eds.), ICSU(FAGS)/IUGG(IACS)/UNEP/UNESCO/WMO', p. 336.
URL: *doi:10.5904/wgms-fog-2012-11*

Winsvold, S. H., Kääb, A. & Nuth, C. (2016), 'Regional Glacier Mapping Using Optical Satellite Data Time Series', *IEEE Journal of selected topics in applied earth observations and remote sensing* **9**(8), 3698–3711.

7 Appendix

7.1 Algorihtm 1: Automated scene selection based on different approaches

```
1  // change following lines for different approaches (i-iii): 25-35, 66-68,
      91, 113
2  //user-defined variables
3  var start = 2010;
4  var end   = 2010;
5  var buffersize = 300; // [m]
6
7  var scale_meanCloud = 30; // [m]
8  var scale_snowcount = 30; // [m]
9  var gte_ndsi = 0.2;
10 var ss_LowThreshold = 0.5;
11 var ss_HighThreshold = 1.15;
12 var cloud_threshold = 15; //[%]
13
14 //___________________________________________________________________
15 // visualisation options
16 var visRGBl7 = {"opacity":1,"bands":["B3","B2","B1"],"gamma":1.3};
17 var visFalse = {"opacity":1,"bands":["B5","B4","B3"],"gamma":1.3};
18 var vis_ratio = {"opacity":0.75,"palette":["0080ff","ff0080"]};
19 var vis_Ice = {"opacity":1,"bands":["B4"],"palette":["00fff4","fdffb6"]};
20
21 //___________________________________________________________________
22 //import the FeatureCollection (imported RGI polygons are saved as
      FeatureCollections)
23 //import-tool: shpescape.com/ft/upload/
24
25 var glacierpolygon = ee.FeatureCollection(
26   //'ft:1s_HD0wV6tLeswo5oGVMQ3mI2UL63cv1lNErc4i1F' // 3glaciers
27   'ft:1hRnmU5H2rgXsywt8m-0TIbmF6I__zl9gqgRsMbaM'  // "bernese alps".
28   //'ft:1DdMSc3xERbmGCfChR3hlkVS-PkLKCVKs2uQw2X7K'  // central Europe
29   //'ft:1oEqGiDwFPyc_t6dG0qdI4942a05k06QBWaAxDNbF' // scandinavia
30
31   //next line is used for approach (iii) and needs a geometry to be drawn
        manually:
32   //geometry
33   )
34   .geometry(); //needed for approach (i) & (ii)
35   //.buffer(buffersize); //optional
36
```

```
Map.addLayer({eeObject: glacierpolygon, name: ’glacierpolygon’});
Map.centerObject(glacierpolygon);
print(glacierpolygon, ’glacierpolygon’)


//____________________________________________________________________
//create function to map over year_coll

var fnChooseCollection = function(start, end){
  var imgColl = ee.ImageCollection(ee.ImageCollection.load(’LANDSAT/LE7_L1T’)
                .merge(ee.ImageCollection.load(’LANDSAT/LT5_L1T’)))
                .filter(ee.Filter.calendarRange(196, 288, ’day_of_year’)) //end of ablation period
                .filter(ee.Filter.calendarRange(ee.Number(start), ee.Number(end), ’year’))
                .filterBounds(glacierpolygon)
  ;
  return imgColl;
};
// call the function to get a subset of the collection.
// this is where the cloud score gets computed on.
var imgColl_filtered = fnChooseCollection(start, end);
print(imgColl_filtered, ’imgColl_filtered by dateRange’);

//______________________________________________________________________
//add simple cloudscore to each image:
//Computes a simple cloud-likelihood score in the range [0,100] using a combination of brightness,
//temperature, and NDSI. It’s intended mainly to compare multiple looks at the same point for
//*relative* cloud likelihood. (https://developers.google.com/earth-engine/landsat#simple-cloud-score)

var fnCloudScore_simple = function(img){
  var ndsi = img.normalizedDifference([’B2’, ’B5’]);
  //next line is for approach (i)
  //var snow_mask = ndsi.updateMask(ndsi.gte(gte_ndsi));
  //next line is for approach (ii) & (iii)
  var snow_mask = img.clip(glacierpolygon);
  // convert to TOA because simpleCloudScore only works with TOA corrected images
  var toTOA = ee.Algorithms.Landsat.TOA(img);
  var toTOA_masked = toTOA.updateMask(snow_mask);
  var img_masked = img.updateMask(snow_mask);
  //cloudscore
  var cloudscore = ee.Algorithms.Landsat.simpleCloudScore(toTOA_masked);
  //add the cloud-score band to the input image
  return img_masked.addBands(cloudscore, [’cloud’]);
  };

//apply the function above to the imageCollection
```

```
var imgColl_cls_scored = imgColl_filtered.map(fnCloudScore_simple);
print(imgColl_cls_scored, 'imgColl_cls_scored');

//________________________________________________________________________
// Function for mean cloud cover

var fnMeanCloud = function(img){
  var reduction = img.reduceRegion({
  reducer: ee.Reducer.mean(),
  //next line is only used for approaches (ii) and (iii). Otherwise: NDSI-mask limits the perimeter
  //geometry: glacierpolygon,
  scale: scale_meanCloud
  , maxPixels: 34e9
  });
  return img.setMulti({cloud_mean: ee.Dictionary(reduction).get('cloud')});
};

//map it over imgColl_cls_scored to keep all the bands necessary for otsu method...
var imgColl_mc = imgColl_cls_scored.select(['B2', 'B5', 'cloud']).map(fnMeanCloud);
 //...and filter it to be less than "cloud_trheshold"
var imgColl_mc_filter = imgColl_mc.filter(ee.Filter.lte('cloud_mean', cloud_threshold))
                        .filter(ee.Filter.lte('CLOUD_COVER', 40));
print(imgColl_mc_filter, 'imgColl filtered by < cloud_threshold');

//_______________________________________________________________________
// Function to apply the snowcount

var fnSnowcount = function(img){
  var ndsi = img.normalizedDifference(['B2', 'B5']);
  var classification_masked = img.updateMask(ndsi.gte(gte_ndsi));
  var count = classification_masked.reduceRegion({reducer: ee.Reducer.count(),
              //next line is only used for approaches (ii) and (iii). Otherwise: NDSI-mask limits the perimeter
              //geometry: glacierpolygon,
              scale: scale_snowcount
              , maxPixels: 34e9
              });
  return classification_masked.setMulti({snowcount: ee.Dictionary(count).get('B2')});
};

//call the snowcount method on imgColl_mc_filter (already applied with cloudscore)
var coll_SC = imgColl_mc_filter.map(fnSnowcount);

//_______________________________________________________________________
```

```
//seasonal snow analysis: add countSnow and path/row-information to each image passed to that function

var fn_PR = function(img){
  var path = ee.String(img.get("WRS_PATH"));
  var row = ee.String(img.get("WRS_ROW"));
  return img.setMulti({pathrow: path.cat("_").cat(row)});
};

//map fn_PR to add path/row-information
var coll_SC_PR = coll_SC.map(fn_PR);

// Get the distinct path/rows as features.
// Note, this leaves the feature with system:index from a random image, but it doesn't matter.
var distinct = ee.FeatureCollection(coll_SC_PR.distinct(["pathrow"]).select(["pathrow"]));

// Join.
var filter = ee.Filter.equals({
  leftField: 'pathrow',
  rightField: 'pathrow'});
var saveall = ee.Join.saveAll("collection").apply(distinct, coll_SC_PR, filter);
//print(saveall, 'saveall');
// Make a collection of collections.
var collcoll = saveall.map(function(result) {
  return ee.ImageCollection.fromImages(result.get("collection"))
      .set("pathrow", result.get("pathrow"));
});

//______________________________________________________________________
//function to add the value of the mean snowcount of an ImageCollection to each individual image in it.

var fnMean = function(imgColl){
  var agg_sum_sc = ee.ImageCollection(imgColl).aggregate_sum('snowcount');
  var imgColl_size = ee.ImageCollection(imgColl).size();
  var mean_sc = ee.Number(agg_sum_sc).divide(imgColl_size);
  var fnSetMean = ee.ImageCollection(imgColl).map(function(img){
    return img.set({mean_sc: mean_sc});
  });
  return fnSetMean
;};

//______________________________________________________________________
// function to add a value which serves as 'checkback' for recognising OUTLIERS

var fnLimit = function(img){
  var count = ee.Number(img.get('snowcount'));
  var mean = ee.Number(img.get('mean_sc'));
  var ss_limit = count.divide(mean); // if this value is bigger than 1.2 --> exclude
  return img.set({ss_limit: ss_limit});
};
//filter the scenes by the upper and lower threshold (ss_High...& ss_Low...)
var bestOf_scenes = collcoll.map(fnMean)
                    .flatten()
                    .map(fnLimit)
                    .filterMetadata('ss_limit', 'not_greater_than', ss_HighThreshold)
                    .filterMetadata('ss_limit', 'not_less_than', ss_LowThreshold);

//______________________________________________________________________
//apply a function which weights the snowcount with the ACCA cloudscore from Landsat.

var accaWeight = bestOf_scenes.map(function(img){
  var weightedSC = ee.Number(ee.Image(img).get('snowcount')).multiply(ee.Number(ee.Image(img).get('CLOUD_COVER')).multiply(0.8).add(1));
  return img.setMulti({snowcount_weighted: weightedSC});
});
print(accaWeight, 'accaweight');

//sort the resulting scene selection
var doublesort = accaWeight.sort('pathrow').sort('snowcount_weighted');
print(doublesort, 'double sorted (pathrow, snowcount)');
Export.table.toDrive({collection: doublesort, folder:'GEE-Exports', fileFormat: 'CSV'});

// taskname to copy for export
var taskname = ee.String('mss_').cat(start.toString()).cat('-').cat(end.toString())
              .cat('_').cat('cl').cat(cloud_threshold.toString())
              .cat('_').cat('ndsi').cat(ee.Number(gte_ndsi).multiply(100).toShort())
              .cat('_').cat('sclMC').cat(scale_meanCloud.toString())
              .cat('_').cat('sclSC').cat(scale_snowcount.toString()).cat('_');
print(taskname);
```

7.2 Algorihtm 2 fragment: Segmentation of ratio image and area calculation

```
//NOTE: Prior to this area calculation algorithm could/should be a
//      automatic or manual scene selection. To keep things faster
//      no such scene selection has been included but can be added.

//SET VARIABLES
var year = 2000;
var meters = 100; //buffer-size
var iceAndSnowThreshold = 2.5;

// 1. load feature(collection) OUTPUT: buffered polygon (feature or featureCollection)
var fnLoadFeature = function(meters){
  var glacierpolygon = ee.FeatureCollection(
    'ft:1s_HD0wV6tLeswo5oGVMQ3mI2UL63cv1lNErc4i1F' // 3glaciers
    //'ft:1hRnmU5H2rgXsywt8m-0TIbmF6I__zl9gqgRsMbaM'  // "bernese alps".
    //'ft:1DdMSc3xERbmGCfChR3hlkVS-PkLKCVKs2uQw2X7K'  // central Europe
    //'ft:1riNNK134G_MAJ19d5D9J0Jd7kfRbj3u0jukE0Vgk' //Trift Gletscher
    //'ft:137OesD3wY9mtBg3xW9X9TuQqt5i2zd74GcQoh04l' // southern andes new
    //'ft:1Qppz_vwNhJmAcJ0SynJaFN23Pc1LylZnnMqbnh6s' // normal new zealand
    //'ft:1C5cK7ZANtNzWkhb6ORxC2Gc003tbijzVGyIt04jK' // iceland
    //'ft:1oEqGiDwFPyc_t6dG0qdI4942a05k06QBWaAxDNbF' // scandinavia
  );
  var bufferIt = function(glacierpolygon){
    return glacierpolygon.buffer(meters);
  };
  var buff_glacierpolygon = glacierpolygon.map(bufferIt);
  Map.addLayer(buff_glacierpolygon);
  Map.centerObject(buff_glacierpolygon, 11);
  return buff_glacierpolygon
  //.first() //outputs only the first polygon
  ;
};
// --------------------------------------------------------------------------
//here could be a whole scene selection process as it is stated in
//Algorithm 1 (automatic scene selection)

//2. function to load an ImageCollection
var fnChooseCollection = function(year){ // L5=LT5_L1T, L7=LE7_L1T, L8=LC8_L1T
  var imageCollection = ee.ImageCollection('LANDSAT/LE7_L1T')
                  .filterDate(ee.String(ee.Number(year).toShort()).cat('-07-15'),
                  ee.String(ee.Number(year).toShort()).cat('-10-15'));
  return imageCollection;
};
// ---------------------------------------------------------------------------

// 3. simpleComposite
var fnSimpleComposite = function(imageCollection){
  var simpleCompositeIMG = ee.Algorithms.Landsat.simpleComposite({
      collection: imageCollection,
      percentile: 50,
      cloudScoreRange: 10,
      maxDepth: 80
      });
  return simpleCompositeIMG;
};
// ---------------------------------------------------------------------------
// 4. ratio with classification of Ice/Snow + masking
var fnRatio = function(simpleCompositeIMG){
  var tmRatioIMG = simpleCompositeIMG.expression(
    'TM3 / TM5', {
      'TM3': simpleCompositeIMG.select('B3'), //landsat 7
      'TM5': simpleCompositeIMG.select('B5') //landsat 7
  });
  var tmRatioClassified_0 = tmRatioIMG.where(tmRatioIMG.lt(iceAndSnowThreshold),
                          0); //...where blablabla...set the output to "number"
  var tmRatioClassified_01 = tmRatioClassified_0.where(tmRatioClassified_0.gte(iceAndSnowThreshold), 1);
  //mask it with itself to get rid of data declared as "0"(zero)
  var tmRatioMask = tmRatioClassified_01.updateMask(tmRatioClassified_01);
  return tmRatioMask;
};

// ---------------------------------------------------------------------------
// 5. calculate area
//with reducer.sum()
var fnAreaCalc_sum = function(tmRatioMaskclip, passedFeature){
    //Generates an image in which the value of each pixel
    //is the area of that pixel in square meters.
    var areaImage = tmRatioMaskclip.multiply(ee.Image.pixelArea());
    var reducedRatioCount = areaImage.reduceRegion({
    reducer: ee.Reducer.sum(),
    geometry: passedFeature.simplify(100).geometry(),
    scale: 30,
    bestEffort: true
    , maxPixels: 34e9
    });
    var calcArea = ee.Number(reducedRatioCount.get('B3')).divide(1e6); // = km2
return calcArea;
};

//////////////////////////////////////////////////////////////////////
//apply the functions 1. - 5.

var loadedFeatureCollection = ee.FeatureCollection(fnLoadFeature(meters));
var imageCollection = fnChooseCollection(year).filterBounds(loadedFeatureCollection.geometry());
print(imageCollection, 'imageCollection')
var simpleComposite = fnSimpleComposite(imageCollection);
```

```
print(simpleComposite)
// ----------------------------------------------------------------------------
// MAIN FUNCTION which sets the respective area property to a glacier
var fnSetArea = function(passedFeature) {
  return passedFeature.setMulti({GEE_Area: fnAreaCalc_sum(fnRatio(simpleComposite), passedFeature)});
};
//apply fnMultiPoly
var series_multifeature = loadedFeatureCollection.map(fnSetArea); //takes loadedFeature one by one
print(’series_multifeature’, series_multifeature);

// ----------------------------------------------------------------------------
// EXPORT
var exportTable = ee.FeatureCollection(series_multifeature); //needs to be a featurecollection to be exported properly
Export.table.toDrive(exportTable);
```

7.3 Algorihtm 3: Using the Otsu thresholding function to map snow cover and approximate the accumulation area ratio (AAR)

```
//With the some variations of the following lines the masks can also be mapped
//and visualised for specific years. Follow the link to this algorithm:
//https://code.earthengine.google.com/4774da8d1558c5796c80678c4e9c4c00

var vizParams = {"opacity":1, ’bands’:’swir1,nir,red’, ’gamma’:1.3};
var vizParams2 = {"opacity":1, ’bands’:’red,green,blue’, ’gamma’:1.3};
var vizICE1 = {"opacity":1,"bands":[’red’],"palette":["2fa6ff"]};
var vizICE2 = {"opacity":1,"bands":[’red’],"palette":["FFFF00"]};

//create a sequence of years which should be included in the exported table:
var years = ee.List.sequence(2000, 2002, 2);
print(’years’, years);
///////////////////////////////
var bounds = ee.FeatureCollection(
  //’ft:1e6ybmK_UqYmpgFxSIbOTia-XVuPgu-DQya8YTHip’ //Kesselwandferner
  //’ft:1WJVcFvo4alGl9Enl-9NQ2aOdyvnuwIGTZBDutWo9’ // storglaciaeren
  ’ft:1bqVVU2QG2_76ZYENOPgR11BqMXgI3DkhpR240Uqj’  //findelengletscher
  //’ft:1s_HD0wV6tLeswo5oGVMQ3mI2UL63cv1lNErc4i1F’ // 3glaciers
  //’ft:1hRnmU5H2rgXsywt8m-0TIbmF6I__zl9gqgRsMbaM’  // "bernese alps".
);
Map.centerObject(bounds);
//////////////////////////////
```

```
var         startJulian = 182;
var         endJulian   = 288;
var         increment   = 1;
var     gte_ndsi        = 0.3;
var     scale_snowcount = 30;
var     ss_LowThreshold = 0.4;
var     ss_HighThreshold= 1.2;
var iceAndSnowThreshold = 3;
var scale_otsu          = 30;
var scale_exportimage   = 30;
/////////////////////////////
var sensorBandDictLandsatTOA =ee.Dictionary({
                        L8 : ee.List([1,2,3,4,5,9,6]),
                        L7 : ee.List([0,1,2,3,4,5,7]),
                        L5 : ee.List([0,1,2,3,4,5,6]),
                        L4 : ee.List([0,1,2,3,4,5,6])
  });
// ...change names to:
var STD_NAMES=       ['B1',  'B2',    'B3',  'B4',  'B5',    'B7',    'B6'];
var STD_NAMES2=      ['blue', 'green', 'red', 'nir', 'swir1', 'temp',  'swir2'];
//____________________________________________________________________
//_____________________________________________________________________
//create imageCollection from Top-Of-Atmosphere data
var imgCollTOA = ee.ImageCollection(
  ee.ImageCollection.load('LANDSAT/LE7_L1T_TOA')
  .select(sensorBandDictLandsatTOA.get('L7'), STD_NAMES2)
  .merge(ee.ImageCollection.load('LANDSAT/LT5_L1T_TOA')
  .select(sensorBandDictLandsatTOA.get('L5'), STD_NAMES2))
  );
  //create imageCollection from raw data
var imgCollRAW = ee.ImageCollection(
  ee.ImageCollection.load('LANDSAT/LE7_L1T')
  .select(sensorBandDictLandsatTOA.get('L7'), STD_NAMES2)
  .merge(ee.ImageCollection.load('LANDSAT/LT5_L1T')
  .select(sensorBandDictLandsatTOA.get('L5'), STD_NAMES2))
  )
//////////////////////////////////////////////////////////// MAIN FUNCTION
//------------------------------------------------------------------------
var fnMain= function(startYear){
var imageCollectionTOA= imgCollTOA
  .filter(ee.Filter.calendarRange(startYear, ee.Number(startYear).add(increment), 'year'))
  .filter(ee.Filter.calendarRange(startJulian, endJulian, 'day_of_year'))
  .filterBounds(bounds)
  ;
// Compute a cloud score.  This expects the input image to have the common
// band names: ["red", "blue", etc], so it can work across sensors.
```

```
var cloudScore = function(img) {
  // A helper to apply an expression and linearly rescale the output.
  var rescale = function(img, exp, thresholds) {
    return img.expression(exp, {img: img})
        .subtract(thresholds[0]).divide(thresholds[1] - thresholds[0]);
  };

  // Compute several indicators of cloudyness and take the minimum of them.
  var score = ee.Image(1.0);
  // Clouds are reasonably bright in the blue band.
  score = score.min(rescale(img, ’img.blue’, [0.1, 0.3]));

  // Clouds are reasonably bright in all visible bands.
  score = score.min(rescale(img, ’img.red + img.green + img.blue’, [0.2, 0.8]));

  // Clouds are reasonably bright in all infrared bands.
  score = score.min(
      rescale(img, ’img.nir + img.swir1 + img.swir2’, [0.3, 0.8]));

  // Clouds are reasonably cool in temperature.
  score = score.min(rescale(img, ’img.temp’, [300, 290]));

  // However, clouds are not snow.
  var ndsi = img.normalizedDifference([’green’, ’swir1’]);
  return score.min(rescale(ndsi, ’img’, [0.8, 0.6]));
};

var addCloudband = function(img) {
      // Invert the cloudscore so 1 is least cloudy, and rename the band.
      var toTOA = ee.Algorithms.Landsat.TOA(img);
      var score = cloudScore(toTOA);
      score = ee.Image(1).subtract(score).select([0], [’cloudscore’]);
      return img.addBands(score);
    };
var scoredTOACollection = imageCollectionTOA.map(addCloudband);

//same collection but in RAW format (DN)
var imgCollRAW_filtered = imgCollRAW
  .filter(ee.Filter.calendarRange(startYear, ee.Number(startYear).add(increment), ’year’))
  .filter(ee.Filter.calendarRange(startJulian, endJulian, ’day_of_year’))
  .filterBounds(bounds);
var scoredRAWCollection = imgCollRAW_filtered
  .combine(scoredTOACollection
  .select(’cloudscore’));

// count the snowpixels and add a snowscore + path/row info
var fnSnowcount = function(img){
  var path = ee.String(img.get("WRS_PATH"));
  var row = ee.String(img.get("WRS_ROW"));
  var ndsi = img.normalizedDifference(['green', 'swir1']);
  var classification_masked = img.updateMask(ndsi.gte(gte_ndsi));
  var count = classification_masked.reduceRegion({reducer: ee.Reducer.count(),
              geometry: bounds,
              scale: scale_snowcount,
              maxPixels: 34e9
  });
  return img.setMulti({snowcount: ee.Dictionary(count).get('green')
                                        , pathrow: path.cat("_").cat(row)});
};
// map the snowcount-function over the collection with the cloudscore
var scoredRAWCollection = scoredRAWCollection.map(fnSnowcount);

// Get the distinct path/rows as features.
// Note, this leaves the feature with system:index from a random image,
// but it doesn't matter.
var distinct = ee.FeatureCollection(scoredRAWCollection.distinct(["pathrow"]).select(["pathrow"]));

// Join.
var filter = ee.Filter.equals({
  leftField: 'pathrow',
  rightField: 'pathrow'});
var saveall = ee.Join.saveAll("collection").apply(distinct, scoredRAWCollection, filter);

// Make a collection of collections.
var collcoll = saveall.map(function(result) {
  return ee.ImageCollection.fromImages(result.get("collection"))
      .set("pathrow", result.get("pathrow"));
});

//function to add the value of the mean snowcount of an ImageCollection
//to each individual image in it.
var fnMean = function(imgColl){
  var agg_sum_sc = ee.ImageCollection(imgColl).aggregate_sum('snowcount');
  var imgColl_size = ee.ImageCollection(imgColl).size();
  var mean_sc = ee.Number(agg_sum_sc).divide(imgColl_size);
  var fnSetMean = ee.ImageCollection(imgColl).map(function(img){
    return img.set({mean_sc: mean_sc});
  });
  return fnSetMean
;};
// function to add a value which serves as 'checkback' for recognising OUTLIERS
var fnLimit = function(img){
  var count = ee.Number(img.get('snowcount'));
  var mean = ee.Number(img.get('mean_sc'));
  var ss_limit = count.divide(mean);
  return img.set({ss_limit: ss_limit});
```

```
};

// map these functions above:
var bestOf_scenes = collcoll.map(fnMean)
    .flatten()
    .map(fnLimit)
    .filterMetadata('ss_limit', 'not_greater_than', ss_HighThreshold)
    .filterMetadata('ss_limit', 'not_less_than', ss_LowThreshold);
//print(bestOf_scenes, 'collection after snowcount-filter');

// create a quality-image with the images left over from the previous selection process
var qualityImage2 = ee.ImageCollection.fromImages(bestOf_scenes.toList(1000))
                    .qualityMosaic('cloudscore');

/////////////////    OTSU    /////////////////////
var qualityImageclipped = qualityImage2.clip(bounds);
// Create a combined reducer.
var reducer = ee.Reducer.mean()
    .combine(ee.Reducer.count(), null, true);

// Create an image of threshold constants.
//This is the list where all the possible thresholds are stored.
var thresholds = ee.List.sequence(
  30, //from here...
  180, // to here...
  5); //...with this increment.
var thresholdImage = ee.Image.constant(thresholds);

//////OTSU FUNCTION
var fnOtsu = function(img){

// Compare 'nir' of img against each of the threshold images.
  var bgMask = img.select('nir').lt(thresholdImage);
  var fgMask = img.select('nir').gte(thresholdImage);

  // Compute stats in each half.
  // Magic here: This multiplies 'nir' of img by the mask to replicate B4 of img N times,
  // so we can then mask with multiple masks all at once.
  var bgStats = img.select('nir').multiply(bgMask).updateMask(bgMask)
  .reduceRegion({
    reducer: reducer,
    geometry: bounds,
    scale: 30,
    maxPixels: 34000000000});
  //print(bgStats, 'bgStats');
  var fgStats = img.select('nir').multiply(fgMask).updateMask(fgMask)
  .reduceRegion({
    reducer: reducer,
    geometry: bounds,
    scale: 30,
    maxPixels: 34000000000});

    //check.

  // Loop over thresholds to compute BCV for each.  These aren't images anymore.
  var size = thresholds.size().subtract(1); //

  var bcv = ee.List.sequence(0, size).map(function(n) {
    var n = ee.Number(n).int();
    // Names for the mean and count properties.
    var mean = ee.String('constant_').cat(ee.String(n)).cat(ee.String('_mean'));
    var count = ee.String('constant_').cat(ee.String(n)).cat(ee.String('_count'));

    // Normalize the counts
    var cb = ee.Number(bgStats.get(count))
    var cf = ee.Number(fgStats.get(count))
    var total = cb.add(cf);
    var wb = cb.divide(total);
    var wf = cf.divide(total);

    var mb = ee.Number(bgStats.get(mean)).divide(total);
    var mf = ee.Number(fgStats.get(mean)).divide(total);
    return wb.multiply(wf).multiply(mb.subtract(mf).pow(2)); // !!!!!
  })

  var maxbcv = bcv.sort().get(size); // gets the maximum value of bcv-list
  //create an image with selected band of thresholdImage: constant value!
  var indexofOptimum = thresholdImage.select(bcv.indexOf(maxbcv))
  var castToNumber = indexofOptimum.reduceRegion({
  reducer: ee.Reducer.max(),
  geometry: bounds.geometry(),
  scale: scale_otsu,
  maxPixels: 34e9
  });

  return img.setMulti({otsu_threshold: ee.Number(castToNumber
  .get(ee.String('constant_').cat(ee.String(bcv.indexOf(maxbcv)))))});
};
/////////////////////////////////////////////////////////////
var otsuIMG = ee.Image(fnOtsu(qualityImageclipped));
var nir = otsuIMG.select('nir');
var iceSnow = nir.gt(0)
          .add(nir.gt(ee.Number(otsuIMG.get('otsu_threshold'))));

// Output contains 0s and 1s.  Mask it with itself to get rid of the 0s.
var iceMask = iceSnow.mask(iceSnow)//.clip(bounds);

//add the snowcount to the polygon.
```

```
var count = iceMask
.reduceRegion({
  reducer: ee.Reducer.frequencyHistogram().unweighted(),
  geometry: bounds,
  scale: 30,
  bestEffort: true
  , maxPixels: 34e9
});

var toFC = ee.FeatureCollection(ee.Feature(count.values()));

var dictionary = count.get('nir')
var thisyear = ee.String(ee.Number(startYear).toShort())
  .cat('-').cat((ee.Number(startYear)
  .add(increment).toShort()));
var icecount = ee.Number(ee.Dictionary(dictionary).get('1'));
var snowcount= ee.Number(ee.Dictionary(dictionary).get('2'));
return bounds.setMulti({
        year: thisyear
      , snowcount: snowcount
      , icecount: icecount
      , proxyAAR: snowcount.divide(icecount.add(snowcount))
})
}
///////////////////////////////////////////////////end of MAIN FUNCTION
//map the MAIN FUNCTION over the set of years
var series = ee.FeatureCollection(years.map(fnMain));
print(series, 'series');
Export.table.toDrive({
  collection: series
  //, description:
  , folder: 'Algorithm3'
  //, fileNamePrefix:
  , fileFormat: 'CSV'});
```

7.4 Algorihtm 4: Separation of clean- and debris-covered ice and creation of debris masks

```
//The following link includes the process of automatic scene selection
// and the creation of the debris-mask:
//https://code.earthengine.google.com/d0c3364def6f1952584ff0589afc22ab

//load the respective image composite (saved as an asset to your drive)
//based on this image the debris-mask will be created.
var composite = ee.Image("users/*username*/*nameOfImageComposite");
//example composite
//https://code.earthengine.google.com/?asset=users/noahzeltner/SUI_qualityImage
////////////////////////////////////////////////////////////////////////
var vizParams = {
  bands: ['vis-red', 'vis-green', 'vis-blue'],
  min: 0,
  max: 255,
  gamma: [1.2]
};
var vizICE2 = {"opacity":1,"bands":['vis-blue'],"palette":["FFFF00"]};
Map.addLayer(composite, vizParams, 'quality Image');
var t = 2.5;

var tmRatioIMG = composite.expression(
  'red / swir1', {
    'red': composite.select('vis-blue'),
    'swir1': composite.select('vis-red')
  });
var tmRatioMask = tmRatioIMG.updateMask(tmRatioIMG.gte(t));
var tmRatioMaskbin = tmRatioMask.lt(t).and(tmRatioMask.gte(t));

//create a binary ratio mask (only 0s and 1s)
var ratio = tmRatioMaskbin.where(tmRatioMaskbin.eq(0), 1);

//load the featurecollection for the debris-mask (glacier inventory import)
var fc = ee.FeatureCollection(
  //'ft:1HYxXmQrLBrp4IQ9wzvsX5q4C5D6zQw9N6BSH-BrI' // centralAsia
  //'ft:13GQSeqSXIZHWyjC-CbDcdya0-cyGRaFa7oPcLPZN'  // centralAsia without rock outcrops
  'ft:1Fu9aEQhD3UsqLvikVWfWtyR8VCTynxNRktHIyqG3' // swiss alps
  );
Map.addLayer({eeObject: fc, name: 'featureCollection', shown:false});
//clip ratio to featurecollection
var r_clipped = ratio.clip(fc).unmask(0, false);
//create constant image and clip it to the featurecollection
var constant = ee.Image(1.0);
var c_clipped = constant.clip(fc);

//subtract the clipped ratio image rom the clipped constant image:
//debris should stay "1" whereas rest is 0.
var debris = c_clipped
  .subtract(r_clipped
  );
Map.addLayer({eeObject: debris, name: 'debris', shown: false});
//create the debris-mask
var debrisMask = debris.updateMask(debris);
Map.addLayer({eeObject: debrisMask, visParams: {"palette":["#00FF00"]}, name: 'debris mask', opacity: 1})
```

Acknowledgements

I am thankful to many people who supported me throughout this Master's thesis. First of all, I want to thank my supervisors, Michael Zemp and Frank Paul. Their scientific and motivational support fundamentally contributed to the success of this thesis. A special thanks goes to Frank Paul who provided assistance within the scientific fields crossing the topic of this thesis and with revising it extensively.

I am very grateful to all the Google Earth Engine developers and users throughout the world who provided me with constructive suggestions and solutions in regard to the challenges which emerged during the coding process, most notably Noel Gorelick.

Furthermore, I want to thank all my fellow students and colleagues for their interest in my work and their support and discussions at the coffee table: Joël Durand, Michael Fehlmann, Tobias Riebler and Luca Scherrer. A special thanks goes to Andri Künzli and Philipp Möhr who provided distraction and comfort and helped me to change perspective from time to time.

My deepest gratitude goes to my family who provided me with encouragement and constantly supported me through all the time of my thesis; my parents Rita and Heinz Zeltner and my siblings Nina and Nico Zeltner. Finally, I am deeply thankful to Nadine Ursprung, for her enormous support in all respects and for having been there for me throughout this time.

Personal declaration

I hereby declare that the submitted thesis is the result of my own, independent work. All external sources are explicitly acknowledged in the thesis.

Date: ______________________________ Signature: ______________________________