

UNIVERSITY OF CALIFORNIA
Santa Barbara

Glacial Response to Climate Change in the Tien Shan Mountain
Range of Central Asia

Submitted in partial fulfillment
of requirements for the degree of
Master of Arts
in Geography

by

Taylor Smith

Committee in charge:
Professor Bodo Bookhagen, Chair
Professor Dar Roberts
Professor Leila Carvalho

September 2014

The thesis of Taylor Smith is approved.

Dar Roberts

Leila Carvalho

Bodo Bookhagen, Committee Chair

July 2014

Acknowledgements

I would like to firstly thank my adviser, Dr. Bodo Bookhagen, for his invaluable help in developing this research and his support over the last two years. I would also like to thank my committee members, Dr. Dar Roberts and Dr. Leila Carvalho, for their valuable comments and feedback on this thesis. I would also like to acknowledge support from the ERI Natural Hazards Fellowship and NSF grant AGS-1116105.

Abstract

Glacial Response to Climate Change in the Tien Shan Mountain Range of Central Asia

by

Taylor Smith

High Mountain Asia (HMA) has been strongly impacted by climate change, and will continue to be impacted by diverse climate stressors in the coming years. A combination of changing weather patterns and retreating glaciers has begun to impact water resources in the region; the rate, extent, and mechanisms behind these changes are poorly quantified due to a lack of data and process understanding. As rivers sourced in HMA feed more than a billion people in the region, increased understanding of climate impacts on regional hydrology is essential for effective water management. Glaciers play a diverse role in hydrology throughout the range, with some catchments having little to no reliance on glacial meltwater, and some catchments receiving almost all of their water from glaciers. This study aims to decipher the impact of climate change on glaciers in the Tien Shan mountain range, a large and understudied region that extends roughly 2,800 km northeast from the Pamir Knot in Tajikistan through China, Kazakhstan, Kyrgyzstan, and Mongolia towards the Altai Mountains.

Previous studies in the region lack the data to decipher gradients in glacial retreat across the range. To address this data deficit, the author designed and implemented a glacial mapping algorithm which delineates both clean glacial ice – methods which

are well documented – and glacial debris tongues, which often require extensive manual digitization. This research is the most significant part of the thesis, and improves upon methods developed to automatically delineate glacial areas using spectral, topographic, velocity, and spatial relationships. The author found that the algorithm misclassifies between 2 and 10% of glacial areas, compared to a ~700 glacier control dataset. These results show significant improvement over previously used methods, as well as large-scale glacier databases such as the Randolph Glacier Inventory (RGI) which are commonly used in mountain-belt scale studies of glaciers.

After validating the algorithm against multiple manually digitized control datasets, the author applied it to a study area encompassing six Landsat scene footprints along the strike of the Tien Shan Orogen. A statistically significant, though minor, gradient in glacier retreat was found, where those glaciers in the west of the range have retreated less than those glaciers in the east. This gradient is explained by differences in regional climate, where large winter storms tend to arrive from the west and break apart and weaken before moving towards the eastern edge of the range. This is substantiated by differences in mean annual precipitation, as derived from Tropical Rainfall Measurement Mission (TRMM) 3B43 V7 data, as well as differences in atmospheric lapse rates, where the western region of the range remains cold year-round and the more eastern edges have larger annual temperature fluctuations, as well as a lower overall lapse rate, which implies higher temperatures at higher elevations.

As glacial area extents do not present a complete picture of the impacts of climate change on glaciers, the author undertook a pilot study to examine the mass balances of glaciers throughout the range. Mass balance estimates are a proxy for changes in water

equivalent stored in a glacier, and are thus essential measurements for understanding the hydrology of the range. The author presents preliminary evidence that high elevation areas in the Tien Shan are thickening, even in the context of retreating glacial tongues. This implies an overall gain of ice mass in the region, despite generally retreating glaciers. This trend was linked to recent increases in precipitation in the range, as well as increases in air temperature. There is not yet enough data to unravel changes in glacier mass balance across the range, although the author predicts that the recently coined ‘Karakoram Anomaly’ may in fact need to be renamed to include both the Pamir and the western Tien Shan.

Table of Contents

Acknowledgements	iii
Abstract	iv
Table Of Contents	viii
List of Figures	xi
I. State of Scientific Knowledge on Glaciology and Climate Change in the Tien Shan Range	1
A. Climate Patterns of the Greater HMA Region	1
B. Climate Setting of the Tien Shan Range	6
C. Glacial Studies in the Tien Shan Range	8
II. Improving Automated Glacial Mapping with a Multi-Method Approach: Examples from the Tien Shan	10
A. Introduction	10
B. Study Area and Data Sources	12
Study Area	13
Data Sources	14
C. Methods	17
Data Preparation	17
Lake Delineation	18
Glacier Delineation	19
D. Results	29
Statistical Analysis of Algorithm Errors	29
E. Discussion	33
Retreat Rates of Selected Glaciers	33
Areal Changes in Glaciers Across the Entire Tien Shan	35
Links to Climate Trends	36
Discussion of Algorithm Use Cases	39
F. Conclusions	40
III. Progress Towards Mass Balance Analysis of the Tien Shan	42
A. Introduction	42
B. Data Sources	43
Optical Data	43

Elevation and Topographic Data	43
C. Methods	44
DEM Correction	44
Mass Balance Estimates	49
D. Preliminary Results	49
E. Preliminary Discussion	51
F. Future Work	53
References	55
Appendix A. Python Code - Glacier Mapping	64
Appendix B. Matlab Code - Lake Delineation	78
Appendix C. Matlab Code - Glacier Delineation	80
Appendix D. Python Code - DEM Correction	87
Appendix E. DEM Correction: Step 1 - Coregistration	98
Appendix F. DEM Correction: Step 2 - Tilt Correction	99
Appendix G. DEM Correction: Step 3 - Aspect Correction	103
Appendix H. DEM Correction: Step 4 - Elevation Correction	108
Appendix I. DEM Correction: Step 5 - DEM Comparison	111

List of Figures

Chapter I

1.1	Percentage of summer rainfall (May-Oct), derived from daily TRMM 3B43 V7 data (1998-2014). The figure illustrates the general split between summer and winter precipitation driven by the Indian Summer Monsoon (ISM) and the Winter Westerly Disturbances (WWDs), as well as the location of the Siberian High which blocks continental precipitation into the Tien Shan during the winter season.	2
1.2	Average precipitation in January and July, 1961-1990. Data from Bohner (2006), analyzing NCAR/CDAS GCM data. Figure modified from Sorg et al. (2012).	7
1.3	Glacial area decline between 2000 and 2007 throughout the Tien Shan modified after Narama et al. (2010)	9

Chapter II

2.1	Study Area, showing SRTM V4.1 topography (Farr and Kobrick, 2000), locations of six Landsat scene outlines used in the study, and along-strike profile (red).	12
2.2	Steps of the algorithm as developed in this research, showing initial spectral classification vs final automated classification results on top of a Landsat scene. Further detail is provided in Figures 2.3-2.8.	17
2.3	Spectrally delineated glacier outlines (Red), over Landsat bands B7/B4/B2 loaded as Red/Green/Blue. Low reflectance on B7 (typical of glacial ice) results in glaciers being displayed as cyan.	20
2.4	Manual glacial outlines (Blue) and Algorithm outlines (Orange) before velocity filtering. Landsat Band 8 background.	21
2.5	Manual glacial outlines (Black) over FFT results on Landsat 7 Band 2 image.	23
2.6	Manual glacial outlines (Black) over 3B/3N ratio showing relative surface roughness.	24
2.7	Manual glacial outlines (Blue) and Algorithm outlines (without velocity integrated) (Black). Correlation results in the background.	25

2.8	Manual glacial outlines (Blue) and Algorithm outlines (with velocity and statistical filtering) (Orange). Landsat band 8 in the background.	28
2.9	Bulk elevation distributions of sampled glaciers, with manual delineation (reference dataset) in blue, algorithm-derived delineation in red, spectral delineation in green, and RGI in black.	29
2.10	Percentage offset of area across elevation slices from a manual control dataset, with algorithm-derived delineation in red, spectral delineation in green, and RGI in black.	30
2.11	Elevation distributions of over- and under-classified glacial areas.	31
2.12	(A) Area mismatch vs average elevation of each glacier, (B) Area mismatch vs area of glaciers, (C) Area mismatch vs average slope, (D) Area mismatch vs average aspect.	33
2.13	Retreat for a glacier towards the far east of the study area, with 95% confidence bounds. Retreat: 0.5%.	34
2.14	Retreat for a glacier towards the far west of the study area, with 95% confidence bounds. Retreat: 0.1%.	34
2.15	Retreat rates of index glaciers across the Tien Shan, showing a slight trend from east to west, with 95% confidence bounds.	35
2.16	Swath Profile along the Tien Shan (swath outline presented in Figure 2.1). (A) Topography of the swath profile, and generalized internal vs external watershed delineation. (B) Mean and Maximum elevation, as well as area above 4000m. Glacier centroids and median elevations plotted in Red. (C) SWE, TRMM precipitation data, and snow covered area. Percentage of summer (red) and winter (green) precipitation from TRMM (1998-2013), on a scale of 0-100. (D) Atmospheric Lapse Rates. Average in Black, Nov-Apr in Red, May-Oct in Blue.	37

Chapter III

3.1	Example output of aspect correction script, showing selected height difference vs height difference divided by tangent of slope and aspect vs height difference divided by tangent of slope.	47
3.2	Number of points in 10m elevation bins vs the difference in height between ASTER and SRTM DEMs. Black is uncorrected, Red is co-registered, Blue is tilt-corrected, Pink is aspect corrected, and Green is elevation corrected. Illustrates improved agreement between SRTM and ASTER DEMs over stable terrain through stepwise correction, as larger numbers of pixels are well matched between the SRTM and ASTER dataset (Δh around zero).	48
3.3	Mean bias between SRTM DEM and ASTER DEM.	49

3.4	Height difference between an example ASTER DEM and the SRTM DEM. Showing general spatial patterns in mass change.	50
3.5	Glacier hypsometry, as well as mass balance estimates through time. Shows a consistent trend of mass loss at the tongue and mass gain in the high elevations across all timesteps (Blue - August 2002, Red - June 2003, Green - July 2013)	51
3.6	Precipitation trends (1940-1991) for station data across the Tien Shan. Generally flat trends, with the exception of low elevation areas in the north and west. Modified from Aizen et al. (1997).	52

I. State of Scientific Knowledge on Glaciology and Climate Change in the Tien Shan Range

High Mountain Asia (HMA) – defined loosely as the seven major mountain ranges of Asia: the Himalaya, the Hindu Kush, the Karakorum, the Pamir, the Tien Shan, the Altai, and the Kunlun – faces diverse and extensive risks due to climate change (e.g., Xu et al., 2009; Bookhagen and Burbank, 2010; Jacob et al., 2012; Stocker, 2013). High dependence on seasonal rainfall and glacial runoff for water resources in the region (e.g., Aizen et al., 2007a; Lioubimtseva and Henebry, 2009; Bookhagen and Burbank, 2010; Sorg et al., 2012; Kitoh et al., 2013; Gardelle et al., 2013; Menon et al., 2013; Stocker, 2013), as well as lack of resources in the region to adapt to climate stressors, have created a large and growing at-risk population. Although the entire region remains relatively understudied in comparison to other glaciated regions around the globe, there is a distinct lack of data for the Tien Shan. The purpose of this thesis is to expand upon current knowledge on glaciers throughout the Tien Shan, and their response to regional climate change.

A. Climate Patterns of the Greater HMA Region

The majority of HMA is influenced by a combination of the Indian Summer Monsoon (ISM) and Winter Westerly Disturbances (WWDs) which combine to create precipitation gradients across the continent (Figure 1.1) (Lang and Barros, 2004). The most recent Intergovernmental Panel on Climate Change (IPCC) report notes that HMA is likely to see increasingly extreme and variable weather due to climate changes (Stocker, 2013). Increasing moisture availability, due to both increased surface evaporation and

rising sea surface temperatures, is likely to lead to increases in both mean and extreme precipitation from the ISM, as well as increased spatial variability in extreme events (e.g., May, 2004; Lau et al., 2006; Malik et al., 2011; May, 2011; Ghosh et al., 2012; Turner and Annamalai, 2012; Stocker, 2013; Wang et al., 2014; Singh et al., 2014; Cannon et al., in review).

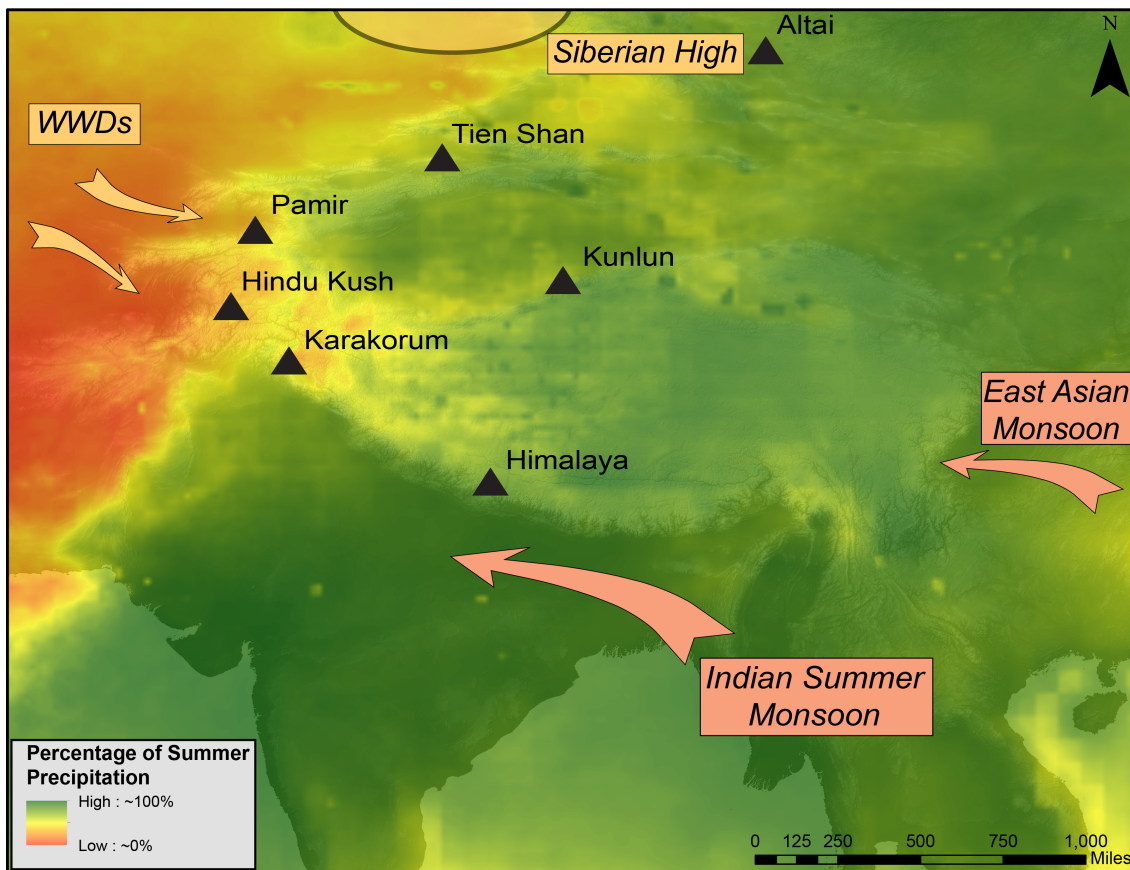


Figure 1.1: Percentage of summer rainfall (May-Oct), derived from daily TRMM 3B43 V7 data (1998-2014). The figure illustrates the general split between summer and winter precipitation driven by the Indian Summer Monsoon (ISM) and the Winter Westerly Disturbances (WWDs), as well as the location of the Siberian High which blocks continental precipitation into the Tien Shan during the winter season.

In the southeastern regions of HMA, the ISM accounts for nearly 80% of yearly precipitation (Lang and Barros, 2004; Bookhagen and Burbank, 2010). The southwest-

erly regions of HMA receive 50% or more of their precipitation from WWDs (Bookhagen and Burbank, 2010; Palazzi et al., 2013; Cannon et al., in review). The more northern ranges (Tien Shan and Altai) are also influenced by climatic teleconnections associated with the North Atlantic Oscillation (NAO), Arctic Oscillation, Eurasian/Polar Pattern, Siberian High, and the El Niño-Southern Oscillation (ENSO), though precipitation is generally low throughout both of these ranges (Cohen and Entekhabi, 1999; Lang and Barros, 2004; Folland et al., 2008; Linderholm et al., 2011; Cannon et al., in review). Several studies have argued that anthropogenic influence can drive shifts in these major circulation systems (e.g., Fu, 2003; Ramanathan et al., 2005; Gautam et al., 2009; May, 2011; Turner and Annamalai, 2012).

Changes in land cover have long been recognized as drivers of climate change (e.g., Charney, 1975; Charney et al., 1977). Charney et al. (1977) concluded that changes in surface albedo could alter weather patterns significantly in both monsoonal and arid regions, based on data collected in North America, Asia, and Africa. More recently, changes in surface roughness, forest cover, evapotranspiration, and seasonal water storage have all been recognized as drivers of climate change (Fu, 2003). Asia has experienced particularly dramatic land cover changes over the past three thousand years, as expanding populations have converted large areas of tropical forest to farmland, and increased the amount of degraded and desertified land (Fu, 2003). Fu's (2003) analysis suggests that anthropogenic influence has weakened the East Asian Summer Monsoon (which provides moisture to the eastern edge of the Himalaya), and enhanced the winter monsoon in southeast Asia; these observations have been substantiated by other studies (e.g., Xue et al., 2004; Yoo et al., 2006; Lee et al., 2011).

The ISM, which provides the majority of the moisture for the southern reaches of HMA, is driven by temperature gradients between the Indian Subcontinent and the Indian Ocean, and has been described in detail in several publications (e.g., Flohn, 1957; Webster, 1983; Fasullo and Webster, 2003; Gadgil, 2003; Barros et al., 2004; Magagi and Barros, 2004; Barlow et al., 2005; Bookhagen et al., 2005; Boos and Kuang, 2010; Bookhagen and Burbank, 2010). Recent work has challenged the temperature gradient between land and ocean as a driving factor and has identified the steep orographic gradient between the subcontinent and the Tibetan Plateau as responsible for creating these strong temperature gradients, as it insulates the warm and moist subcontinent from the colder and drier regions of HMA (Boos and Kuang, 2010). However, these structural controls are unlikely to influence monsoon patterns at a human-relevant timescale, and thus are unlikely to account for observed changes in the ISM.

Several studies have examined changes in the ISM, which has seen a 20% increase in early summer rains since the 1950s (Gautam et al., 2009; Menon et al., 2013; Kitoh et al., 2013). Kitoh et al. (2013) showed that increasing moisture availability has driven an increasingly erratic and powerful summer monsoon, which agrees with the IPCC assessment that monsoon seasons are likely to increase in both severity and duration (Stocker, 2013). The ‘atmospheric brown cloud’ over southeast Asia, as well as other aerosols, have recently been implicated in changes in the ISM (e.g., Ramanathan et al., 2007; Ramanathan and Carmichael, 2008; Gautam et al., 2009; Bonasoni et al., 2010), though long-term warming trends have also been implicated in increasing monsoon strength, due to factors such as volcanic aerosols, precession, obliquity, and eccentricity in earth’s orbit, and changes in glacier size and distribution (e.g., Overpeck et al., 1996; Anderson

et al., 2002).

The mechanism by which aerosols may have modified the monsoon is still debated. Ramanathan et al. (2005) propose that aerosols such as black carbon reduce sea surface temperatures, which in turn reduces available moisture and monsoon strength. Alternatively, it is possible that aerosols act to trap heat over the ocean and increase available moisture (Ramanathan et al., 2005). Lau et al. (2006) proposed that aerosol stacking along the Himalayan front may modify the temperature gradient in the region, leading to earlier summer rainfall. An atmosphere enriched in aerosols provides more abundant nuclei for cloud formation, which could help promote early precipitation (Lau et al., 2006). It is not clear yet what role aerosols play in modified monsoon seasons, or whether anthropogenic aerosols such as black carbon or increased dust availability due to desertification play a more important role in the ISM (Gautam et al., 2009). Natural cycles, such as El Niño-Southern Oscillation (ENSO) variations, seasonal snow cover, and seasonal aerosol content, can also drive changes in the monsoon (Bookhagen et al., 2005). These natural cycles have shown a notable upwards trend over the past few decades, resulting in increased rainfall penetration into the drier reaches of HMA (Bookhagen et al., 2005).

The northwestern reaches of the Himalaya, and the more central ranges of the Hindu Kush, Karakorum, Pamir, and Tien Shan are strongly influenced by WWDs (Lioubimtseva et al., 2005; Palazzi et al., 2013). These disturbances are driven by baroclinic systems that arise from the Mediterranean and Arabian Seas, and which occasionally extend into central Asia (Barlow et al., 2005). These storms tend to carry far less moisture, and deposit snow rather than rain (Hewitt, 2005, 2011, 2014). Large-scale changes

in the Central Asian landscape, such as the desiccation of the Aral Sea, have increased aerial dust, and could potentially alter major weather patterns. Unfortunately, following the collapse of the Soviet Union, many monitoring stations ceased to be maintained, and thus long-term data is sparse in the region (Lioubimtseva et al., 2005).

B. Climate Setting of the Tien Shan Range

The Tien Shan range extends nearly 2,800 km northeast from the Pamir Knot towards the Altai mountains. Its climate is primarily controlled by WWDs, which are most active between January and March (Lang and Barros, 2004), as well as the Siberian High, which serves as a blocking mechanism for winter storms (Narama et al., 2010). Both the Pamirs and the Tien Shan receive moisture from extra-tropical cyclones brought into the region by westerly troughs. Moisture transport is generally from the southwest, from the Mediterranean and Arabian Seas. Precipitable water is generally low in all cases, especially in comparison to moisture availability in other glaciated regions of Asia, such as the southern Himalayan front. Furthermore, precipitation is not evenly distributed, as the western edges of the range tend to have more winter precipitation, while the eastern and central ranges tend to see more summer precipitation (Narama et al., 2010). Moisture is also topographically trapped, with the outer reaches receiving relatively more moisture than the inner ranges, which border the Taklamakan Desert in northwest China (Sorg et al., 2012). This difference can be seen in Figure 1.2.

In general, storms hit the Pamirs and break into northern and southern portions: the southern portions head along the front of the Tibetan Plateau towards the Hindu Kush and Karakorum, and the northern portions continue onwards towards the Tien Shan, where they are generally trapped along the orography of the range (Cannon et al.,

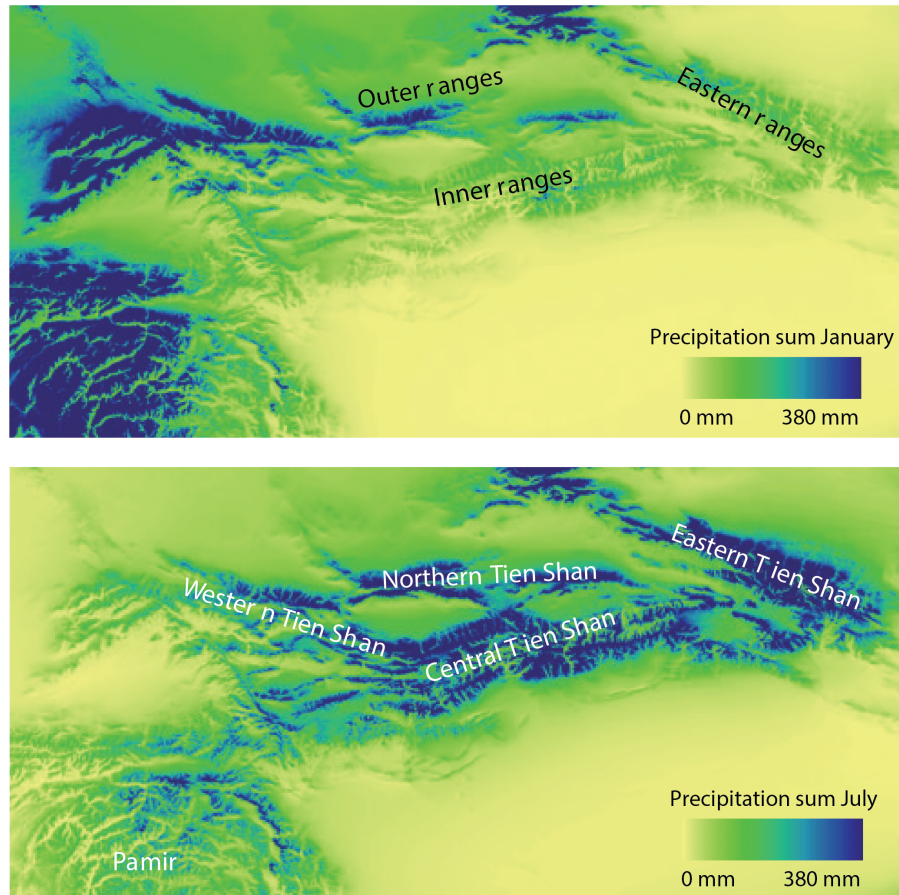


Figure 1.2: Average precipitation in January and July, 1961-1990. Data from Bohner (2006), analyzing NCAR/CDAS GCM data. Figure modified from Sorg et al. (2012).

in review). As they continue northwards they generally weaken, dropping less precipitation as they move from west to east. This can also be seen in the number of events at the 95th percentile, where the Pamirs received 87 extreme events and the Tien Shan received 25 (1998-2013) (Cannon et al., in review). Storms in the Tien Shan are often remnants of larger storms, which broke apart on the Pamirs, and generally carry less precipitation.

During the summer months, both the Pamirs and the Tien Shan receive moisture from the northwest, due to the weakening of the Siberian High which allows more continental moisture to flow into the region. As moisture is generally received from

the northwest (perpendicular to the strike of the orogen), there is less of a precipitation gradient from west to east. Low elevation areas generally do not receive summer snow, although some high elevation areas and glaciers receive moisture during summer storms. The summer is also the period of highest precipitation from the Indian Summer Monsoon (ISM), which provides much of the moisture for the more southern Himalaya Range. However, the Tien Shan receives little to no precipitation from the ISM, as it is topographically separated, and influenced by wind patterns derived from continental Europe (Mölg et al., 2012).

C. Glacial Studies in the Tien Shan Range

Changing climate has had an impact upon glaciers in the Tien Shan. Sorg et al. (2012) noted that the outer ranges have lost more glacial area than the inner ranges, though recent estimates using Gravity Recovery and Climate Experiment (GRACE) data estimate glacial mass loss at only -5 ± 6 gigatons per year between 2003 and 2010 (Jacob et al., 2012). These data, however, assume that all gravitational differences are due to glaciers and this study does not consider sediment transport, groundwater, and other processes that may account for gravitational differences. Narama et al. (2010) reported between 9 and 19% area loss in the Tien Shan over the period 1970-2010, depending on the sub-region analyzed. Glacial loss accelerated between 2000 and 2007, except in several cases in the southeastern Tien Shan (Figure 1.3) (Narama et al., 2010). Similar retreat amounts were reported for the central Tien Shan, with glaciers retreating on average 14.2% between the 1860s and 2003, with the bulk of the area loss occurring in the period 1977 to 2003 (e.g., Aizen et al., 2006, 2007b). More recent work has noted that though there has generally been area loss throughout the Tien Shan, retreat rates

have been shrinking over the past 15 years (Pieczonka et al., 2013). There have also been isolated cases of glacial surging over the period 2000-2007, with some glaciers growing by as much as 1,400 meters (Narama et al., 2010).

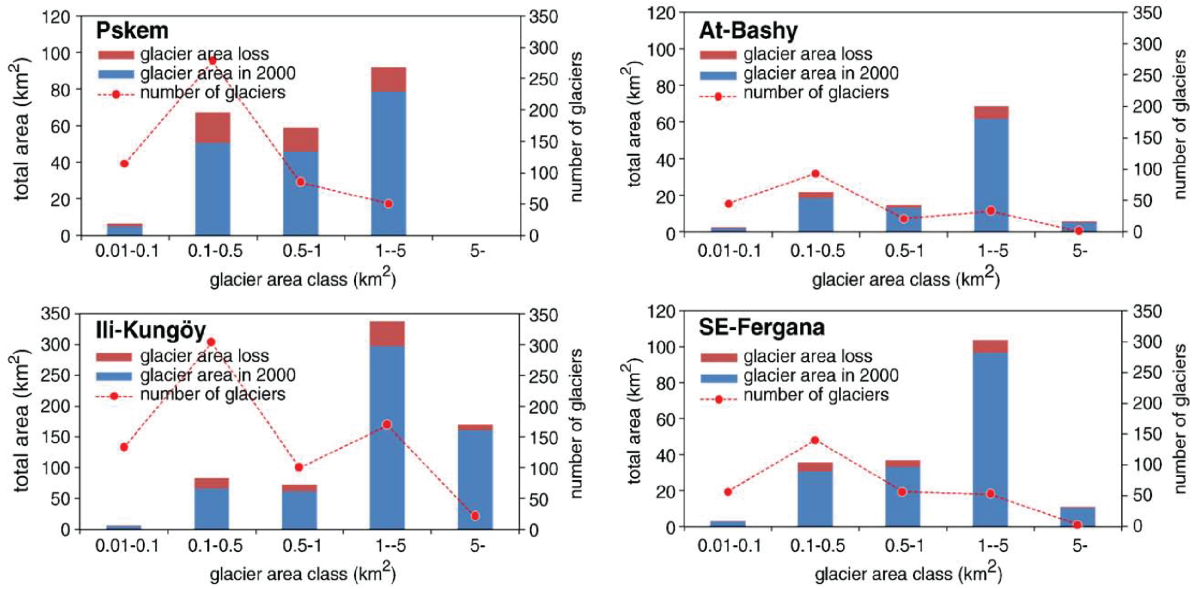


Figure 1.3: Glacial area decline between 2000 and 2007 throughout the Tien Shan modified after Narama et al. (2010)

II. Improving Automated Glacial Mapping with a Multi-Method Approach: Examples from the Tien Shan

A. Introduction

Changes in glacier length have long been considered one of the best indications of climate change (i.e. Oerlemans, 2005), though more recent studies have also assessed glacial volume changes with the advent of new remote sensing techniques (e.g., Berthier et al., 2007; Aizen et al., 2007a; Gardelle et al., 2012, 2013; Bolch et al., 2012; Kääb et al., 2012; Stocker, 2013). This chapter focuses on assessing glacial length fluctuations over a large spatial scale using widely available remotely sensed data. Several attempts have been made to produce accurate and high-resolution glacial outlines, most notably the Global Land Ice Measurements from Space (GLIMS) project (Armstrong et al., 2005; Raup et al., 2007), and the recently produced supplemental GLIMS dataset known as the Randolph Glacial Inventory (RGI) V3.2 (Arendt et al., 2012). A coherent, complete, and accurate global glacial database is important for several reasons, including monitoring global glacial changes driven through climate changes, natural hazard detection and assessment – such as glacial lake outburst floods, and assessment of the role of glaciers in natural and built environments, including glacial contributions to regional water budgets and hydrologic cycles (Racoviteanu et al., 2009). Precision in glacial outlines is of utmost importance for monitoring changes in glaciers, which may only change by 15-30 m/yr (\sim 1-2 pixels of Landsat Enhanced Thematic Mapper (ETM+)/yr). Thus, spatially accurate glacial outlines are imperative for precise glacial change detection (Paul et al., 2009,

2013).

Several methods have been developed to delineate clean glacial ice (i.e. Paul, 2002; Paul et al., 2002; Racoviteanu et al., 2008a,b; Hanshaw and Bookhagen, 2014), relying primarily on spectral data available on satellites such as Landsat and Advanced Spaceborne Thermal Emission and Reflection Radiometer (ASTER). Several authors have also investigated the application of remote sensing datasets to snowcover determination, through both assessments of sub-pixel snow covered area and grain size (i.e. Dozier, 1989; Painter et al., 2003; Dozier and Painter, 2004; Painter et al., 2009). These studies form the basis for analysis of seasonal snowcover throughout the range, an important climatic indicator used in assessing trends in snow covered area across the greater Tien Shan range.

Although significant progress has been made towards automated glacial outline retrieval using satellite imagery, these methods struggle to accurately map debris-covered glaciers, or other glaciers with irregular spectral profiles (Paul et al., 2004; Racoviteanu et al., 2008b; Scherler et al., 2011). Much of this difficulty stems from the similarities in spectral profiles of debris located on top of a glacial tongue and the surrounding hillslopes where the debris is sourced from. The majority of studies examining debris-covered glaciers employ extensive manual digitization in a Geographic Information System (GIS), which is very time consuming, and can introduce significant user-generated errors (Paul et al., 2013; Pfeffer et al., 2014). Building on the multi-spectral, topographic, and spatially weighted methods developed by Paul et al. (2004), the author presents a refined rules-based classification based on spectral, topographic, land-cover, velocity, and spatial relationships between glacial areas and the surrounding environment.

Using a suite of 23 Landsat Thematic Mapper (TM), ETM+ and Optical Land Imager (OLI) images across a spatially and topographically diverse set of study sites comprising six Landsat footprints (Path/Row combinations: 144/30, 147/31, 148/31, 149/31, 151/33, 153/33) along the profile of the Tien Shan Range (Figure 2.1), the author (1) pre-processes the imagery (georeference and co-register), (2) applies the algorithm based on spectral, topographic, and spatial information, (3) analyzes range-wide trends in glacial character, and (4) links spatially heterogeneous glacial retreat to climate factors. We address the hypothesis that spatially heterogeneous glacial retreat throughout the Tien Shan range is driven by precipitation gradients across the range.

B. Study Area and Data Sources

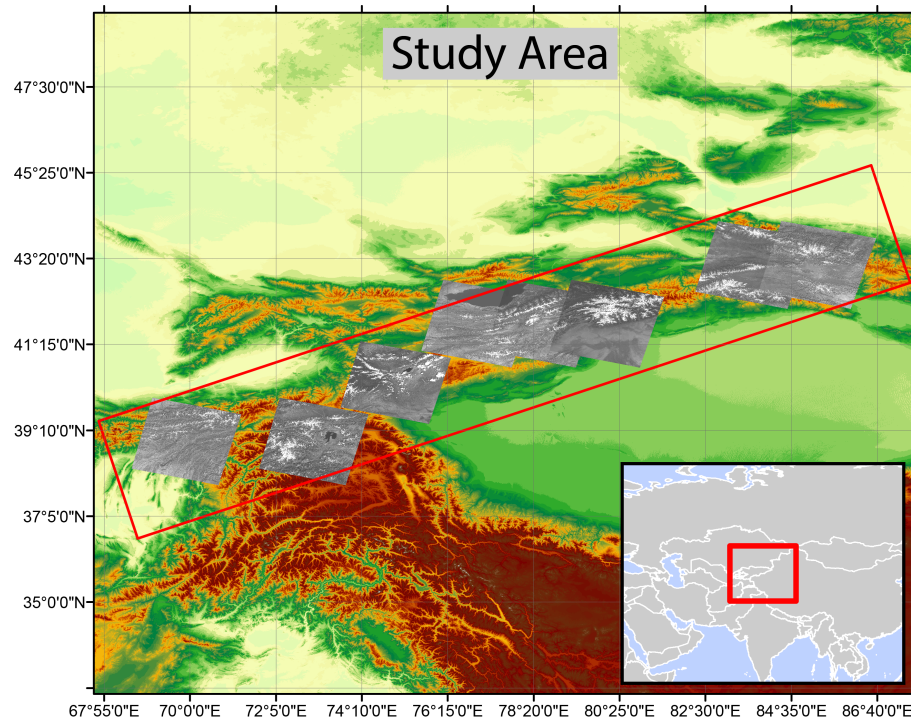


Figure 2.1: Study Area, showing SRTM V4.1 topography (Farr and Kobrick, 2000), locations of six Landsat scene outlines used in the study, and along-strike profile (red).

Study Area

The Tien Shan Range (also known as the Tian Shan) extends almost 2,800 km from Tajikistan northeast into China (Figure 2.1). Its climate is controlled by both the Winter Westerly Disturbances (WWDs) and the Siberian High, which together create strong precipitation gradients throughout the range (Narama et al., 2010; Sorg et al., 2012; Lioubimtseva and Henebry, 2009; Bolch et al., 2011; Cannon et al., in review). The western edges of the range tend to have more winter precipitation in the form of snow, with precipitation concentrated in the spring and summer in the central and eastern reaches of the Tien Shan (Narama et al., 2010).

Sorg et al. (2012) note that due to the interaction of the Siberian High and more western continental atmospheric patterns such as WWDs, the Tien Shan Range has a distinct precipitation gradient with decreasing precipitation from the west to east. There is also a strong gradient between outer and inner reaches of the Tien Shan, as moisture is trapped by the topography of the range (Sorg et al., 2012).

In a smaller scale study, (Narama et al., 2010) found that for the period of 1970-2010, glacial area decreased between nine and 19%, depending on the subregion of the Tien Shan analyzed. Glacial loss was then seen to accelerate between 2000 and 2007, except in a few cases in the southeastern reaches of the Tien Shan (Narama et al., 2010). This agrees generally with work by Narama et al. (2006), who noted that total glaciated area in the northern and central Tien Shan had shrunk by 14.2% between the 1860s and 2003, with the bulk of shrinking occurring between 1977 and 2003. Several larger scale studies (e.g., Kääb et al., 2012; Bolch et al., 2012; Stocker, 2013) have looked at

retreat over large scales, but lack sufficient spatial detail to fully constrain the rates and mechanisms of glacial retreat in the region.

Data Sources

The author's glacial mapping algorithm is based on several datasets. The Landsat 5 (TM), 7 (ETM+), and 8 (OLI) platforms were chosen as the primary spectral data sources, as they provide spatially and temporally extensive coverage of the Tien Shan (Table 1). ASTER can also be used as a source of spectral information, but the author chose to focus on the larger footprint and longer timeseries available through the Landsat archive. In addition to spectral data, the SRTM DEM (~90m) was leveraged to provide elevation, slope, and hillshade information (Farr and Kobrick, 2000). The SRTM data and its derivatives were downsampled to 30m to match the resolution of the Landsat images using bilinear resampling. The USGS Hydrosheds river network (15° resolution) was also used as an input dataset (Lehner et al., 2008).

Several climatic datasets were analyzed to provide context to the analysis of glacial changes in the Tien Shan. Daily gridded Advanced Microwave Scanning Radiometer (AMSR-E) data (0.25° x 0.25° resolution) was used to estimate the snow-water equivalent (SWE) stored throughout the study area on a yearly basis (2002-2011) (Tedesco et al., 2004). The AMSR-E satellite measures microwave brightness temperatures at different frequencies to analyze snow depth, although the algorithm assumes a homogeneous snowpack with constant grain size (Tedesco and Narvekar, 2010). SWE measurements are further complicated by changing snow densities, which are not well represented in the AMSR-E dataset, which converts estimated snow depth to a water equivalent. Despite these caveats, the AMSR-E SWE measurements remain the best estimates of

Table 1: Data table listing Landsat scene IDs used in this study. Organized by WRS2 Path/Row combinations. Starred scene IDs denote ‘Master’ images to which others were rectified.

	144/030	147/031	148/031
Number of Images	4	4	4
Date Range of Images	2002-2013	2000-2013	2002-2013
LT5 IDs	LT51440302006212IKR00 LT51440302009220KHC01	LT51470312011231KHC01	LT51480312007259IKR00 LT51480312011254IKR01
LE7 IDs	LE71440302002257SGS00*	LE71470312000257SGS00* LE71470312002278SGS00	LE71480312002205SGS00*
LC8 IDs	LC81440302013295LGN00	LC81470312013268LGN00	LC81480312013211LGN00
Projection	WGS 1984 45N	WGS 1984 44N	WGS 1984 44N
Comments	Eastern Edge of Study Area	Vicinity of Inylchek Glacier	
	149/031	151/033	153/033
Number of Images	3	5	3
Date Range of Images	1999-2013	1998-2013	1998-2013
LT5 IDs	LT51490312007250IKR00	LT51510331998271XXX01 LT51510332009253KHC00	LT51530331998269AAA01
LE7 IDs	LE71490311999252SGS00*	LE71510332000237SGS00* LE71510332001271SGS00	LE71530332002272SGS00*
LC8 IDs	LC81490312013282LGN00	LC81510332013280LGN00	LC81530332013278LGN00
Projection	WGS 1984 43N	WGS 1984 43N	WGS 1984 42N
Comments			Towards Pamir Knot

the water stored in snowpack throughout much of the world, and especially in remote and understudied areas such as the Tien Shan.

Tropical Rainfall Measurement Mission (TRMM) data (product 3B43 V7) with a spatial resolution of $0.25^\circ \times 0.25^\circ$, aggregated into monthly and yearly means (1998-2014), were used to analyze precipitation trends across the range (Huffman et al., 2010; Bookhagen, 2010; Bookhagen and Burbank, 2010; Duan and Bastiaanssen, 2013). The precipitation radar (PR) onboard TRMM primarily senses rainfall, as liquid water is easier to detect by radar, though the TRMM 3B43 V7 dataset used in this study also contains information from several other datasets (e.g., AMSR-E precipitation estimates, infrared (IR) imagery from several satellites including Geosynchronous Operational Environmental Satellites (GOES), Multifunctional Transport Satellite (MTSat), and Me-

teorological Satellite (Meteosat), Advanced Microwave Sounding Unit (AMSU-B), and Global Precipitation Climatology Centre (GPCC) ground-based rainfall measurements), which are merged during processing to improve precipitation estimates (Huffman and Bolvin, 2007).

Two Moderate Resolution Imaging Spectroradiometer (MODIS) products MOD10C1 (snow covered area, SCA) and MOD11C1 (land surface temperature, LST) were used to analyze yearly average snowcover and atmospheric lapse rates respectively (Hall and Salomonson, 2006; Wan, 2008). MOD10C1 data between 2001 and 2014 were aggregated to seasonal timesteps at $0.05^\circ \times 0.05^\circ$ spatial resolution. MOD11C1 data was aggregated seasonally to calculate atmospheric lapse rates at $0.05^\circ \times 0.05^\circ$ spatial resolution using SRTM elevation data. Lapse rates were calculated using night time temperatures, which have been shown to be more consistent in alpine regions (Colombi et al., 2007). A recent study validated MODIS LST data in the region by comparing ground station data for western Tibet with generated LST products, finding generally good agreement across both day and night (Wang et al., 2007).

The RGI V3.2 dataset is comprised of $\sim 198,000$ glaciers with an estimated extent of $726,800 \pm 34,000 \text{ km}^2$ (Pfeffer et al., 2014). This dataset was derived from previous glacier inventories, as well as satellite imagery over the period of 1999-2010, and provides complete global coverage, excepting ice sheets. This dataset was intended for wide-scale analysis, and has been successfully used to improve estimates of global glacial changes, and their contribution to sea-level rise. In this paper, the dataset is used as a comparison dataset against which to analyze algorithm-derived glacial outlines.

C. Methods

Steps Used to Classify Glacial Outlines

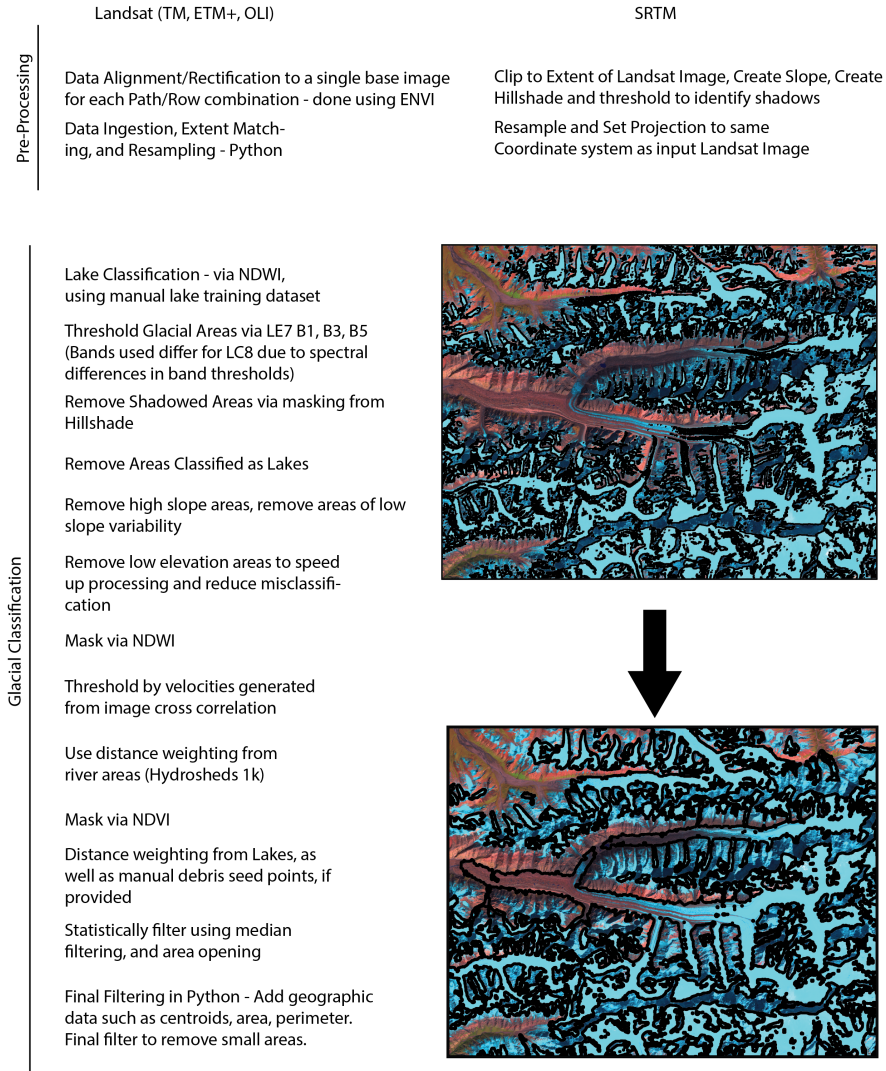


Figure 2.2: Steps of the algorithm as developed in this research, showing initial spectral classification vs final automated classification results on top of a Landsat scene. Further detail is provided in Figures 2.3-2.8.

Data Preparation

For accurate glacial delineation, this research used only those Landsat images which were free of new snow, and had less than 10% cloud cover. The presence of fresh snow in

images tends to overclassify glacial areas and classify non-permanent snow as glaciers. Additionally, cloud cover tends to confuse the algorithm, and render glacial outlines that overlap with cloud cover unreliable (Paul et al., 2004; Hanshaw and Bookhagen, 2014).

After selecting a series of Landsat images, the author co-registered each image to a ‘master’ image specific to each Landsat Row/Path combination, using the Automated Registration and Orthorectification Package (AROP) (Gao et al., 2009). Master images are denoted in Table 1 with asterisk. This ensures that glacier outlines are properly matched, and that any timeseries of glacial change is consistent in space. Once the data were georeferenced and registered, a series of scripts written in Python and Matlab perform the glacial classification (flow diagram shown in Figure 2.2, source code available at www.geog.ucsb.edu/~bodo/Smith_Glacier_Scripts.zip). The Python script prepares several datasets for processing in Matlab by resampling and reprojecting each dataset to the same spatial extent, resolution (30m to match the Landsat data), and projection. Once the hydrologically corrected DEM has been resampled, slope and hillshade images are derived for use in the algorithm. The Hydrosheds 15s river network is buffered by 200m to properly capture stream areas, and then converted to a raster matrix so that it can be ingested along with the other datasets. This dataset is also clipped and projected using the same methods as the SRTM and its derivative datasets. Although the current algorithm is based on proprietary software, the author will continue to update the code with the goal of using only open source tools and libraries in the future.

Lake Delineation

Glacial debris tongues tend to host supra-glacial lakes, particularly during the summer months when snow cover is lowest (Quincey et al., 2007; Gardelle et al., 2011). These

lakes can be used as seed points for distance-weighting mechanisms to more accurately delineate glacial debris tongues. Lakes are delineated using the Normalized Difference Water Index (NDWI) (Gao, 1996), after which misclassified areas are removed by masking out high slopes and shadowed areas (Huggel et al., 2002; Worni et al., 2013; Nie et al., 2013; Hanshaw and Bookhagen, 2014). To increase the modularity of the algorithm, the author did not rely on a fixed NDWI value across all datasets, but instead used a manually generated set of index lakes that exhibit the spectral properties desired. This small manual step greatly increases the number of correctly classified lakes by providing scene specific NDWI thresholds for each processed image. The author notes that future improvements will use radar data to delineate and decipher lake extents.

Glacier Delineation

Spectral Delineation

Calculations are performed on rasterized versions of each input dataset in Matlab. The first step in the classification process leverages Landsat bands 1, 3, and 5. For Landsat 8 OLI images, a slightly different set of bands is used to conform to the modified spectral range of sensors on Landsat 8. For simplicity, bands referenced in this publication refer to Landsat 7 ETM+ spectral ranges. The ratio of TM3/TM5, with additional spectral information from TM1 has long been established as an effective means of delineating glacier areas (e.g., Hall et al., 1987; Hanshaw and Bookhagen, 2014), but is not effective in delineating debris-covered glacial areas (Figure 2.3). Shadowed areas derived from a SRTM-generated hillshade and lake areas derived in the previous lake processing step are then removed from this initial spectral classification. The end result of this step is the spectrally-derived glacier outlines, which are later integrated back into the workflow

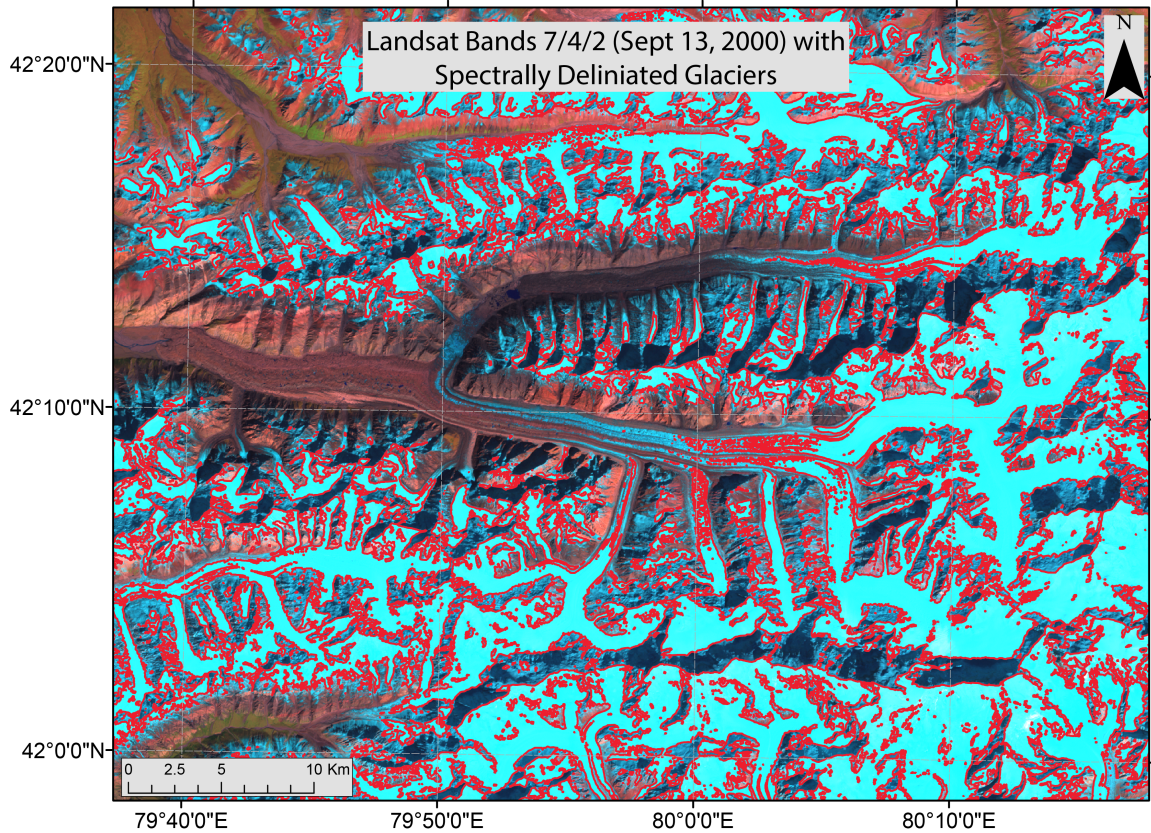


Figure 2.3: Spectrally delineated glacier outlines (Red), over Landsat bands B7/B4/B2 loaded as Red/Green/Blue. Low reflectance on B7 (typical of glacial ice) results in glaciers being displayed as cyan.

before statistical filtering. The results of this step are also saved for comparisons between spectrally delineated glacier outlines and the end result glacier outlines.

Topographic Filtering

Building on the work of Paul et al. (2004), low slope areas (between 1 and 24°) are isolated as areas where debris-covered glaciers are likely to exist. As glacial surfaces tend to be rougher than the surrounding areas, a standard deviation filter (3x3 kernel size) is also applied to the slope and used to mask out areas of low slope variability. Low elevation areas are then masked out to decrease processing time. This step is performed

independent of the previous, spectrally delineated, glacier outlines. In essence, this step identifies areas where there is the potential for a debris covered glacier to exist. Additional filtering is then performed on this subset to identify debris-covered areas (Figure 2.4).

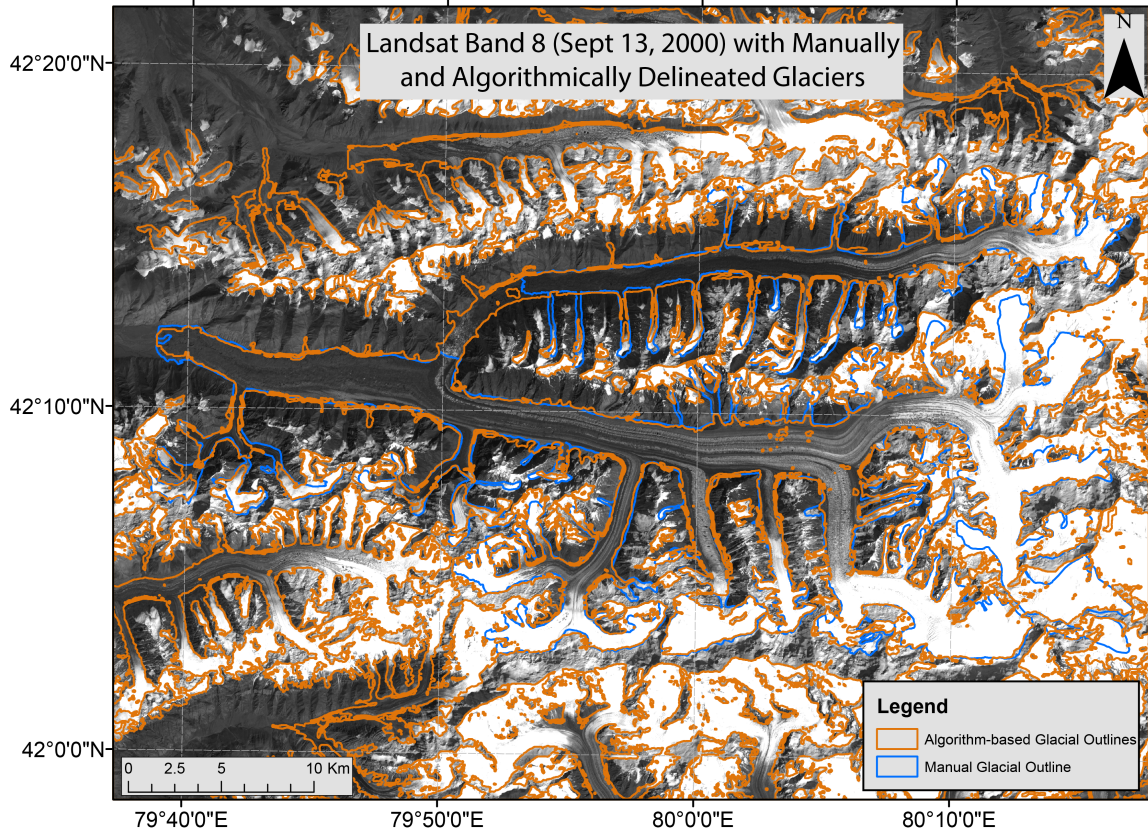


Figure 2.4: Manual glacial outlines (Blue) and Algorithm outlines (Orange) before velocity filtering. Landsat Band 8 background.

The Correlation Image Analysis Software (CIAS) (Kääb, 2002) tool, which uses a method of statistical image cross-correlation, is used to derive glacial velocities from Landsat Band 8 panchromatic images. This method functions by tracking individual pixels across space and time, and provides a velocity surface at the same resolution as the input datasets (15m). The velocity surface is then upsampled using bilinear resampling

to provide a consistent velocity estimate across the entire Landsat scene. Due to the differing timesteps between Landsat images, all velocities used in the glacier algorithm are standardized by the length of time between scene captures. Thus, the author was able to obtain velocity measures in m/yr for all of the target Landsat Path/Row combinations. It is important to note that cloud-free and snow-free images are essential for this step, as the presence of snow or cloud cover can disrupt the correlation process, resulting in anomalous velocity measurements.

The author only used one multi-year velocity measurement for each path/row combination to derive general areas of movement/stability for glacial classification, as using stepped velocity measurements over smaller time increments did not show a noticeable improvement in glacial classification. Furthermore, a method of frequential cross-correlation using the co-registration of optically sensed images and correlation (COSI-Corr) tool (Leprince et al., 2007) was tested and did not show any appreciable improvement in velocity measurements (Heid and Kääb, 2012). Future iterations of the algorithm aim to include provisions for stepped velocity measurements, possibly using the Image Correlation (IMCORR) package (Scambos et al., 1992).

Two additional topographic indices were tested here, though neither provided significant improvement to the algorithm. The author attempted to derive frequential information from several Landsat and ASTER bands, with limited success. Some glaciers do exhibit a unique frequency signature when analyzed using spatial Fast Fourier Transforms (FFTs), though these were not consistent across multiple debris-covered glaciers with differing surface characteristics (Figure 2.5). Additionally, the FFT approach was tested against a principal component analysis (PCA) image derived from Landsat, with-

out significant improvement to the algorithm.

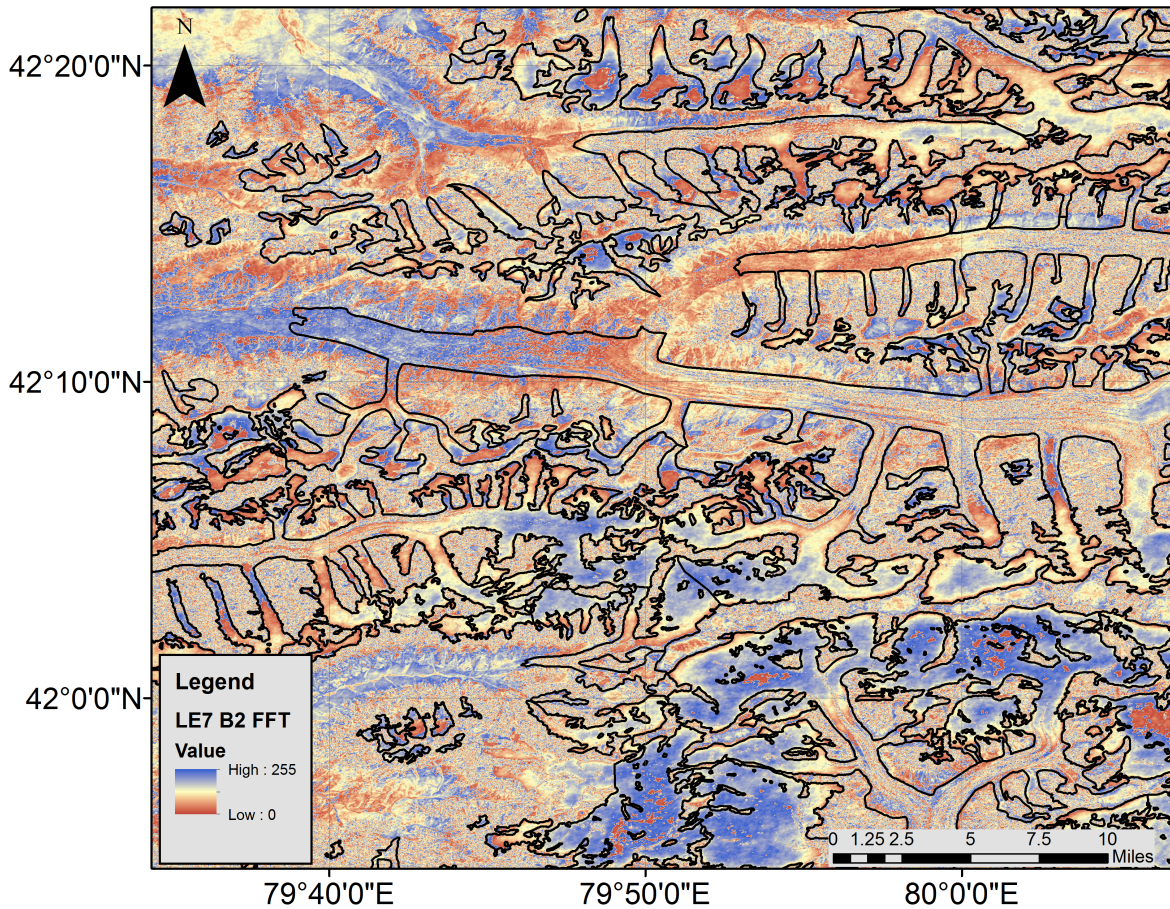


Figure 2.5: Manual glacial outlines (Black) over FFT results on Landsat 7 Band 2 image.

The author also attempted to integrate surface roughness measurements using the ASTER satellite, which contains both a forward looking (3N - nadir) and backwards looking (3B - backwards) imager, primarily intended for the generation of stereoscopic DEMs. However, the difference in imaging angle also provides the opportunity to examine surface roughness by looking at changes in shadowed areas (Mushkin et al., 2006; Mushkin and Gillespie, 2011). The author found that there are slight surface roughness differences between glaciated and non-glaciated areas on some debris tongues, but that these differences are not significant enough to use as a thresholding metric (Figure 2.6).

Furthermore, the nature of the steep topography limits the efficacy of this method, as valleys which parallel the satellite flight path and those which lie perpendicular to the flight path show different results. Shadows tend to move less in parallel valleys, and ridgelines tend to enhance shadow differences in perpendicular valleys. Thus, the algorithm relies on the velocity and slope thresholds to characterize the topography of the glacial areas.

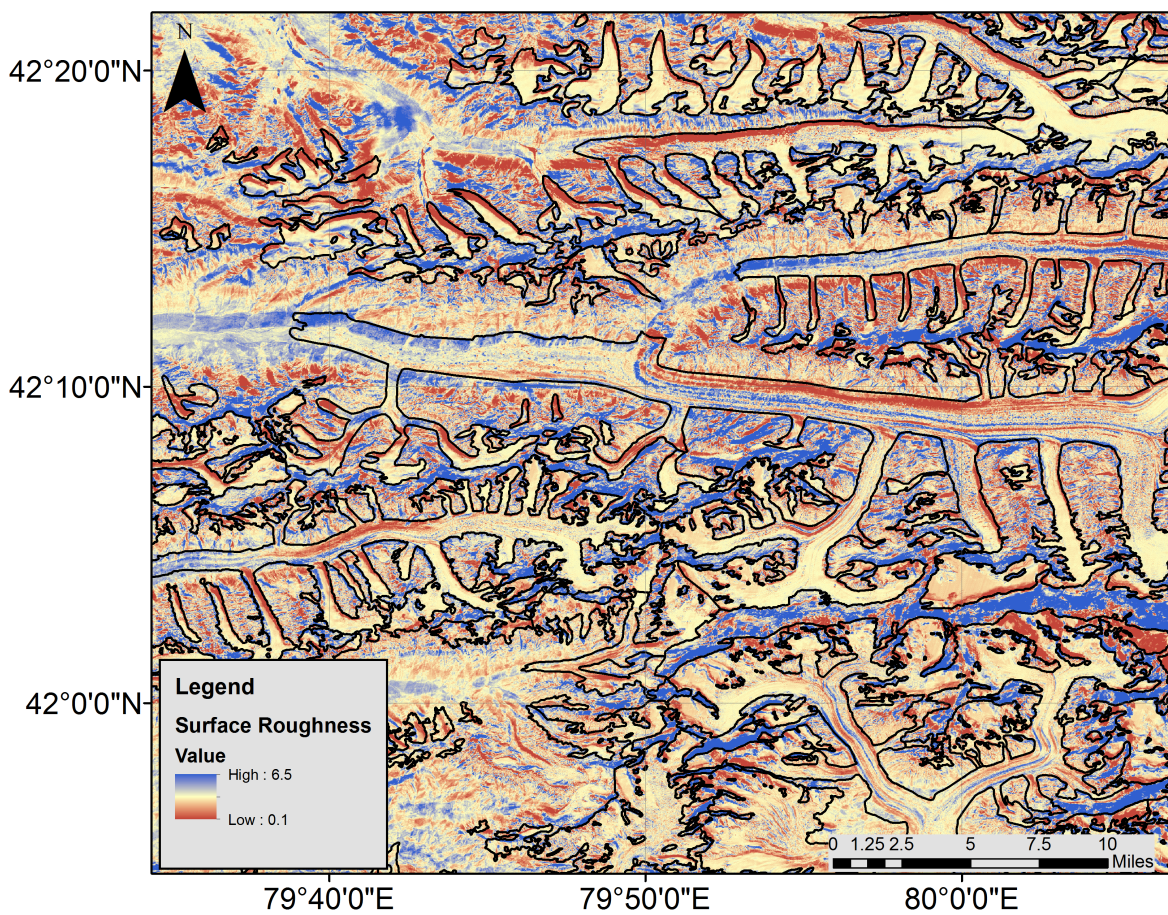


Figure 2.6: Manual glacial outlines (Black) over 3B/3N ratio showing relative surface roughness.

The velocity step is most important for removing hard-to-classify pixels along the edges of glaciers, and wet sands in riverbeds. These regions are often spectrally

indistinguishable from debris tongues, but have very different velocity profiles. It is important to note, however, that this step also removes some glacial area, as not all parts of a glacier are moving at the same speed (Figure 2.7). This can result in small holes in the delineated glaciers, which are addressed by statistical filtering.

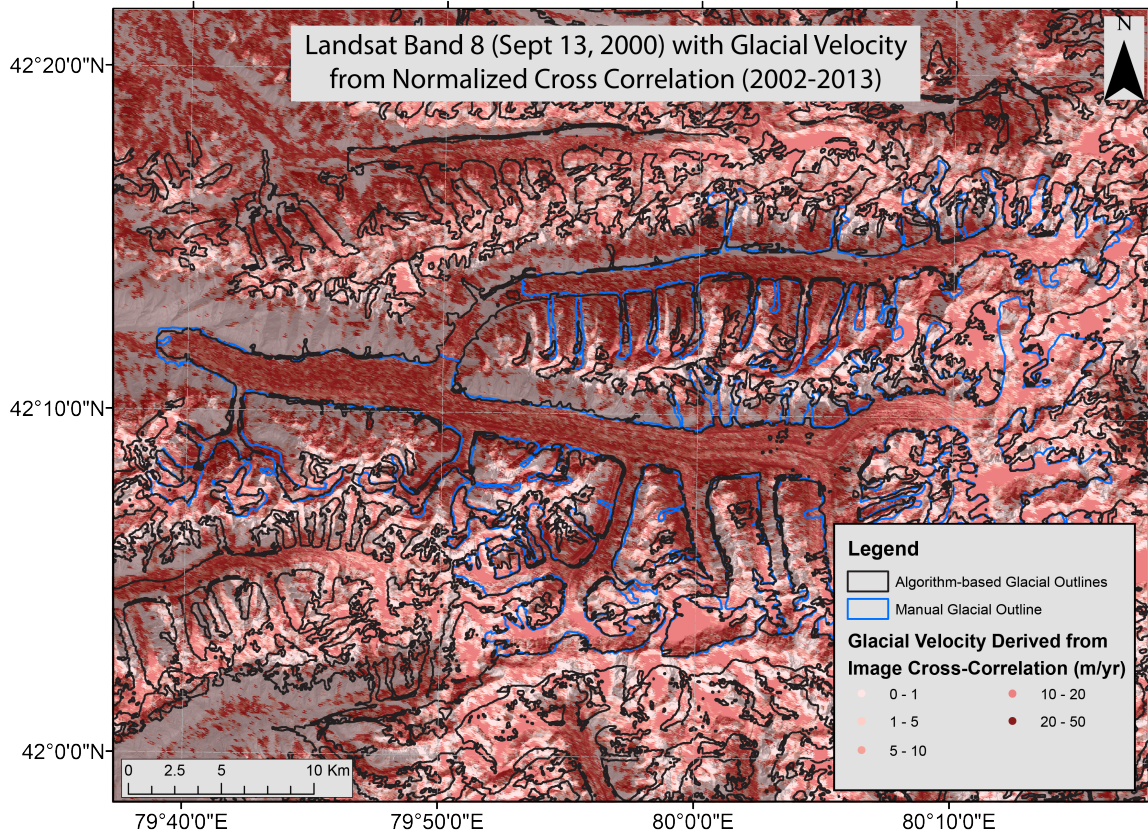


Figure 2.7: Manual glacial outlines (Blue) and Algorithm outlines (without velocity integrated) (Black). Correlation results in the background.

Spatial Weighting

After topographic and velocity filtering, a set of spatially weighted filters was constructed. The first filtering step uses the Hydrosheds river network to remove classified areas which are distant from the center of a given glacial valley. As glaciers occur along the flowlines of rivers, and the Hydrosheds river network generally delineates flowlines

all the way to the peaks of mountains, the river network provides a perfect set of seed points with which to remove misclassified pixels. A second distance weighting is then performed using both the supra-glacial lakes detected during the lake delineation step, and the spectrally delineated glaciers. As debris tongues must occur in proximity to one or more of: (1) glacier areas, (2) the centerlines of valleys, or (3) supra-glacial lakes, this step is effective at removing overclassified areas. At this step, it is also possible to add manual seed points, which may be necessary for some longer debris tongues. The author notes that these are optional, and the majority of glaciers do not need the addition of manual seed points. However, for certain glaciers which do not have many lakes, or do not have lakes large enough to be delineated by Landsat, adding a few manual control points greatly increases the efficacy of the algorithm. In future iterations of the algorithm, high-resolution radar datasets will be used for delineating water bodies on snow and ice to provide additional seed points.

The spatial weighting step is essential for removing pixels spatially distant from any glacier or lake area. In many cases, large numbers of river pixels, and in some cases dry sand pixels, have similar spectral and topographic profiles to glaciers. This step effectively removes the majority of pixels outside the general glaciated area(s) of a Landsat scene.

Masking and Filtering

Once the spatial weighting steps are completed, the glacial outlines are generally accurate. Additional filtering and smoothing steps help to remove some misclassified pixels due to errors in the SRTM, high velocities in non-glacier areas, and areas with similar spectral profiles to glaciers. A NDWI mask is applied to remove dry areas, as

glaciers uniformly exhibit a moisture signal that is detectable with NDWI. A Normalized Difference Vegetation Index (NDVI) mask is also applied, as glaciers in the study region rarely have noticeable vegetation. The author notes that all of the threshold parameters are set on the basis of Landsat scene Path/Row combinations to account for the slightly different topographic, velocity, and landcover settings of spatially diverse Landsat scenes. In general, one set of parameters is sufficient to characterize glaciers within multiple Landsat scenes with the same spatial extent (Path/Row combination) throughout time. However, scenes with extensive cloud cover or snow cover sometimes need separate thresholds to account for the differences in spectral signatures between cloud-free and clouded images.

A set of median filters are then applied, as well as a set of statistical filters designed to remove isolated pixels (3x3 median filter, applied twice, as well as 5x5 median filter, applied after image opening), bridge gaps between isolated glacial areas (image bridging), and fill holes in large contiguous areas (image opening, applied twice). After these filters have been applied, the resulting glacial outlines are post-processed in Python, as described in the next section.

This step is essential for filling holes and reconnecting separated glacier areas. As the filtering methods used are based on a fixed set of threshold values, there are often glacier pixels which are removed. For example, some pixels along the edge of a debris tongue may be moving slower than the provided velocity threshold, and are thus removed. This problem is somewhat, but not completely, mitigated by the statistical filtering (Figure 2.8).

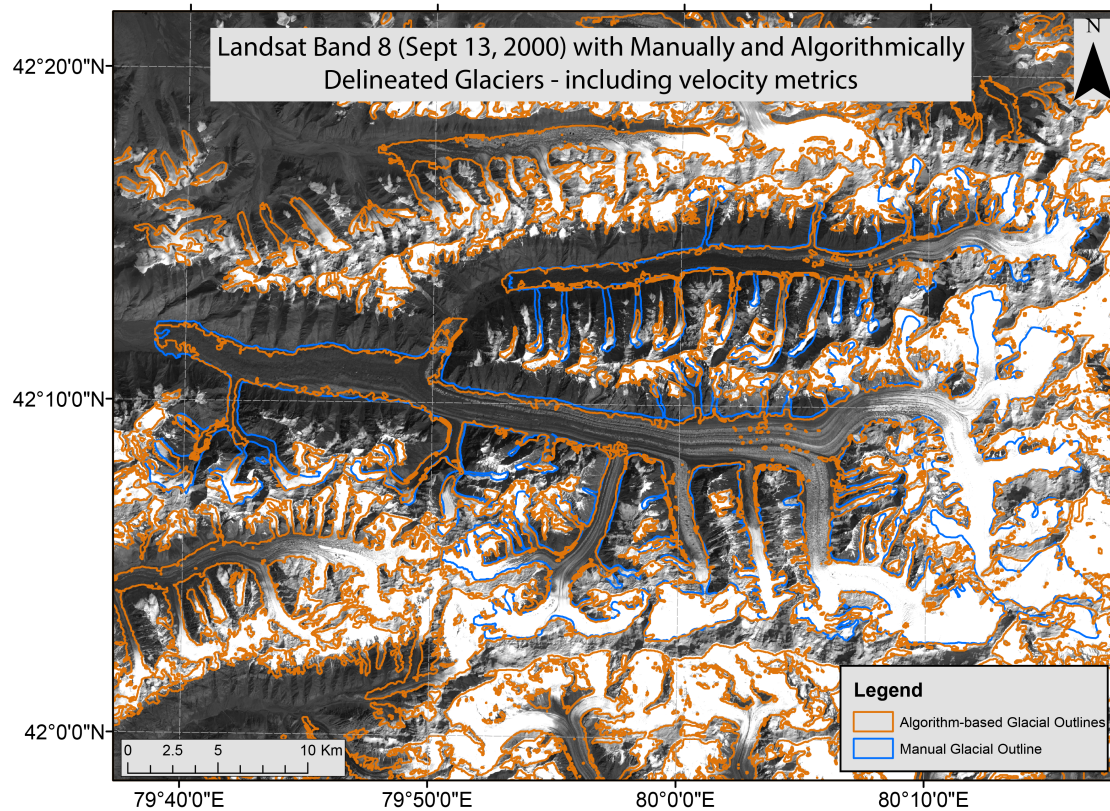


Figure 2.8: Manual glacial outlines (Blue) and Algorithm outlines (with velocity and statistical filtering) (Orange). Landsat band 8 in the background.

Post-Processing

Using the glacial outlines provided by the algorithm, additional metadata is added to each glacier, such as area, perimeter, and polygon centroid, as well as Landsat date and time. Using the area and perimeter information, a final threshold is used to remove very small glacial areas (~ 10 - 20 px or less). The outlines with metadata are then exported to shapefile format, though any geographic vector format could be substituted. For more specific information on the algorithm, please refer to the commented source code in the appendix (appendices A - C).

D. Results

Statistical Analysis of Algorithm Errors

Two manual control datasets encompassing ~ 700 glaciers ($\sim 380 \text{ km}^2$) each were created to test the efficacy of the glacial mapping algorithm. A subset of 138 glaciers of varying size and topographic setting was chosen for more detailed analysis. The unedited algorithm-generated glacial outlines were compared against three separate datasets: (1) the RGI V3.2 (Arendt et al., 2012), which is an optimized subset of the GLIMS database; (2) Spectral outlines, which only classify the glacial areas via commonly used spectral subsetting (using TM1, TM3, and TM5); and (3) a manual control dataset, which was hand-digitized by the author from Landsat imagery. Figure 2.9 shows the bulk elevation distributions across 138 glaciers for each dataset in 10m elevation bins.

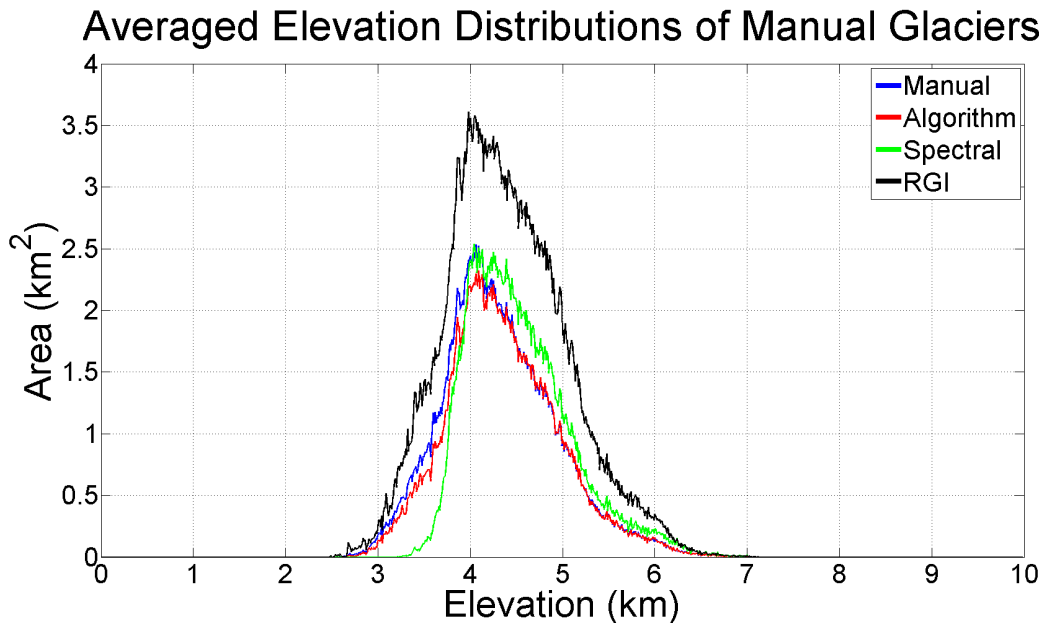


Figure 2.9: Bulk elevation distributions of sampled glaciers, with manual delineation (reference dataset) in blue, algorithm-derived delineation in red, spectral delineation in green, and RGI in black.

From these distributions, it is apparent that the RGI universally overclassifies glacial areas, and that using only a spectral classification tends to overclassify high elevation areas and underclassify low elevation areas. It is important to again note that the RGI is intended for wide-area studies, and thus individual glacial outlines are not guaranteed to be accurate. There is some apparent bias in the author’s algorithm towards low elevation areas, which represent the debris-covered portions of glaciers and are the most difficult to classify. To illustrate the difference between the manual control dataset and each of the comparison datasets, the elevation distributions were differenced in Figure 2.10.

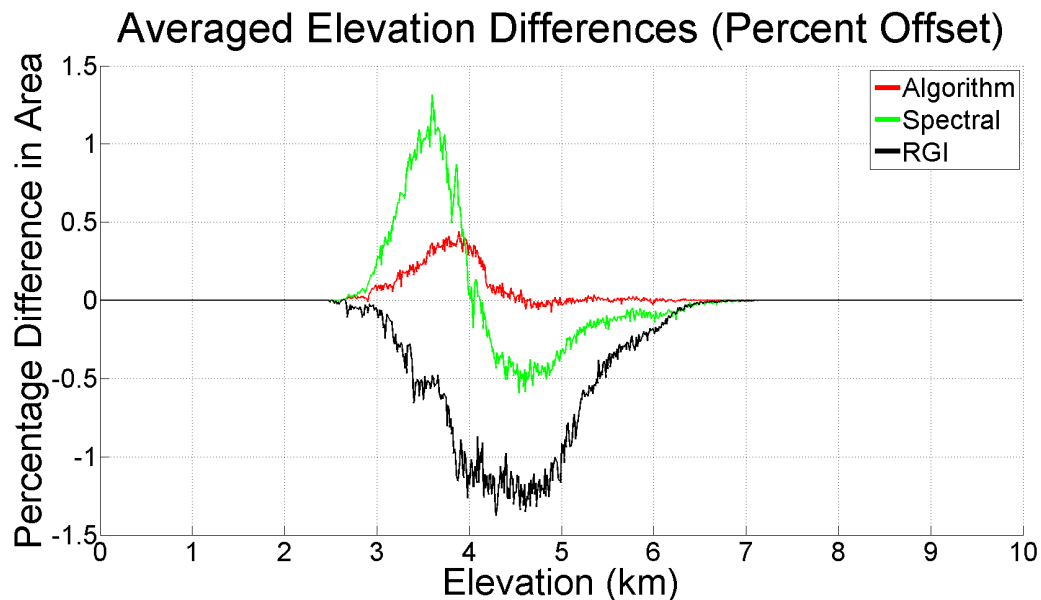


Figure 2.10: Percentage offset of area across elevation slices from a manual control dataset, with algorithm-derived delineation in red, spectral delineation in green, and RGI in black.

In general, the same patterns appear in Figure 2.10 as in Figure 2.9: RGI overclassifies across each elevation slice, the spectral-only method overclassifies in the high elevations and underclassifies in the low elevations, and the algorithm-derived dataset

underclassifies low elevation areas. However, at any given elevation slice, the algorithm-derived data is off by less than 0.5% from the area of the manual control dataset, and is generally well matched at elevations above 4000m. The underclassification at the 3500-4000m class can be attributed to hard-to-classify side glaciers, or glaciers which feed into larger glaciers from connected side valleys. Many of these glaciers are located in shaded or otherwise topographically distinct valleys, which can result in underclassification of connected glacial areas, though an analysis of shadowed area mismatches between both manual and algorithm datasets indicates that shadowed areas are responsible for only very small misclassifications ($\sim 4 \times 10^{-8} \text{ km}^2$, $\sim 5\text{-}10\text{px}$ per elevation slice). Over the span of a single Landsat scene, or even over most individual watersheds within the control Landsat scene, the elevation distributions of the manual and algorithm datasets are virtually indistinguishable (ks-test at the 99% confidence interval passes).

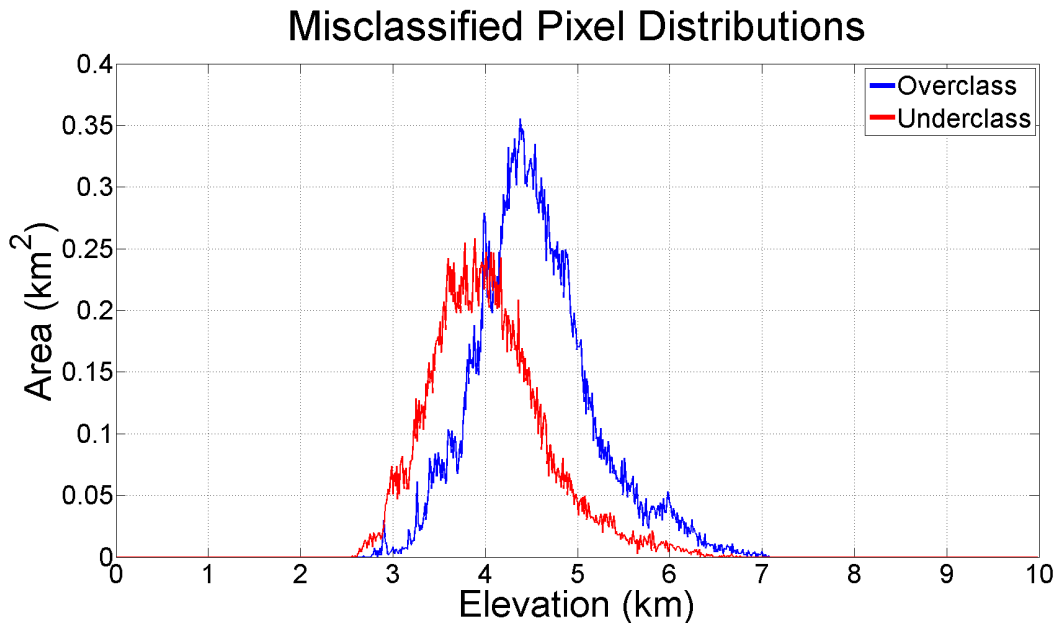


Figure 2.11: Elevation distributions of over- and under-classified glacial areas.

In order to examine persistent bias throughout the algorithm classification, under-

and over-classified areas were examined, as well as areas which may have been unduly affected by shadows. To determine areas of overclassification (underclassification), the manually (algorithm) generated dataset was subtracted from the algorithm (manual) dataset, leaving only pixels which are overclassified (underclassified). Figure 2.11 shows the elevation distributions of under and over classified areas. The algorithm tends to consistently overclassify areas above 4500m, and underclassify areas under 4500m, though much of this misclassification occurs at similar elevation slices, and thus in bulk the mismatch between the manual and algorithm derived datasets is minimal. The total misclassification for the two control datasets are 2% and 10% respectively, which represents a significant improvement from a pure spectral delineation approach, or the data provided by the RGI.

Finally, a set of topographic indices was examined to determine if there was an misclassification bias towards a certain elevation, size, aspect, or slope setting (Figure 2.12). The results of this analysis are that none of the topographic indices show a strong trend across the analyzed glaciers, though there is a slight bias towards underclassifying lower elevation areas vs higher elevation areas, and underclassifying lower slope areas. These are known errors associated with the difficulty of classifying glacial tongues, particularly those with heavy debris cover. However, on average, the mismatch between the algorithm and manual control datasets is minimal regardless of topographic setting (between 2-10%). It is important to note that the algorithm datasets used in the statistical analysis were not manually edited, and are the direct output of the author's algorithm.

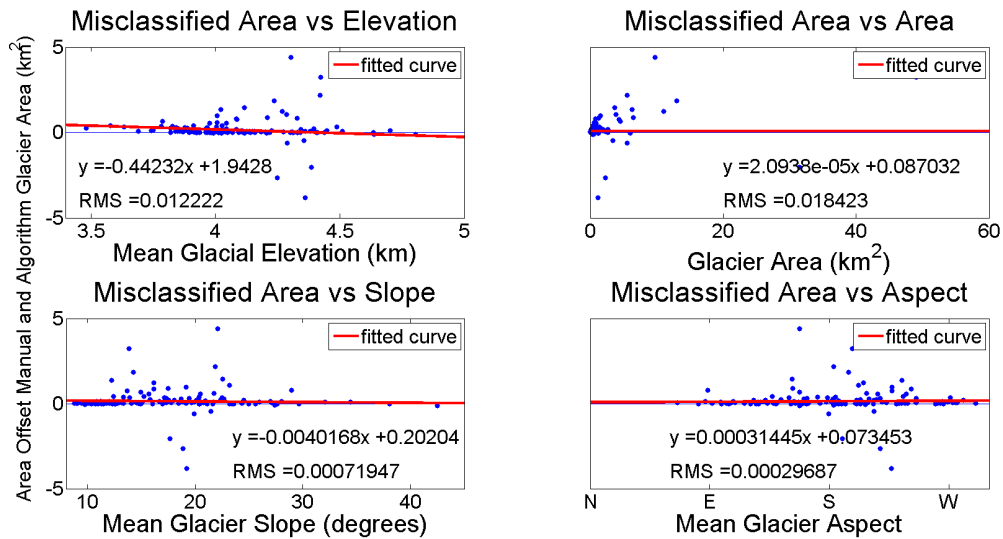


Figure 2.12: (A) Area mismatch vs average elevation of each glacier, (B) Area mismatch vs area of glaciers, (C) Area mismatch vs average slope, (D) Area mismatch vs average aspect.

E. Discussion

Using the newly developed glacier delineation algorithm, the author analyzed retreat rates of glaciers across the greater Tien Shan Range (Figure 2.1).

Retreat Rates of Selected Glaciers

Two example glaciers were chosen to display specific retreat rates (Figures 2.13 and 2.14). In general, glaciers have retreated over the period 1998-2013. However, including additional data sources such as Corona, Landsat Multispectral Scanner (MSS), ASTER, and non-cloud free Landsat TM, ETM+ and OLI images was outside of the scope of this study. Thus, area measurements are limited to the manually selected Landsat scenes.

The author would like to note that the primary use of the algorithm is not in assessing trends within single individual glaciers, but in assessing trends at a larger spatial scale. Due to the difficulties inherent in dividing one glacier area from another,

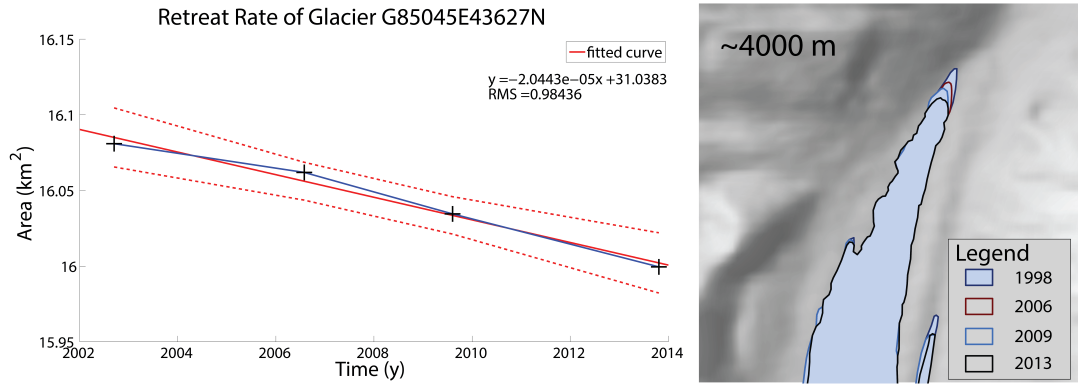


Figure 2.13: Retreat for a glacier towards the far east of the study area, with 95% confidence bounds. Retreat: 0.5%.

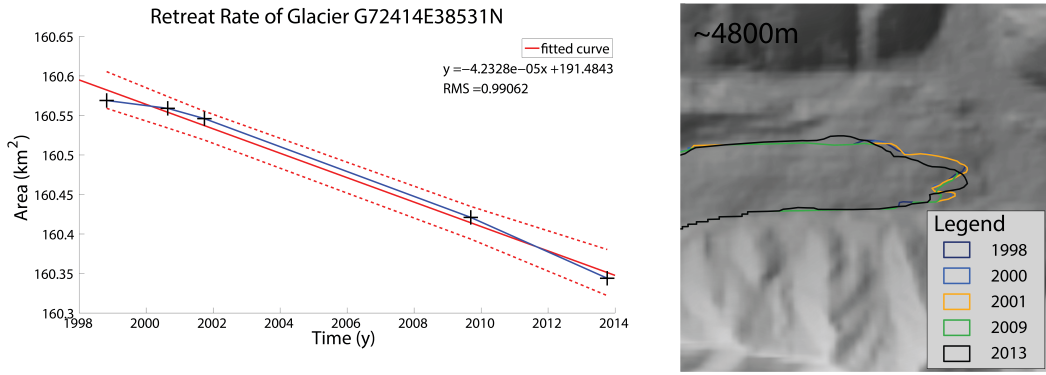


Figure 2.14: Retreat for a glacier towards the far west of the study area, with 95% confidence bounds. Retreat: 0.1%.

particularly at high elevation areas where snowcover often connects multiple glaciers, retreat rates on a glacier-by-glacier basis are associated with larger uncertainties. Furthermore, over- and under-classified areas which tend to even out over a large spatial scale can present problems at the scale of individual glaciers in a particularly distinct topographic setting. An effort was made to choose glaciers which are somewhat isolated to demonstrate retreat rates at a small scale, using both the algorithm and some manual corrections at the tongue of the glacier. The algorithm is more useful when glacier area changes are assessed at a watershed, drainage basin, or range-wide scale.

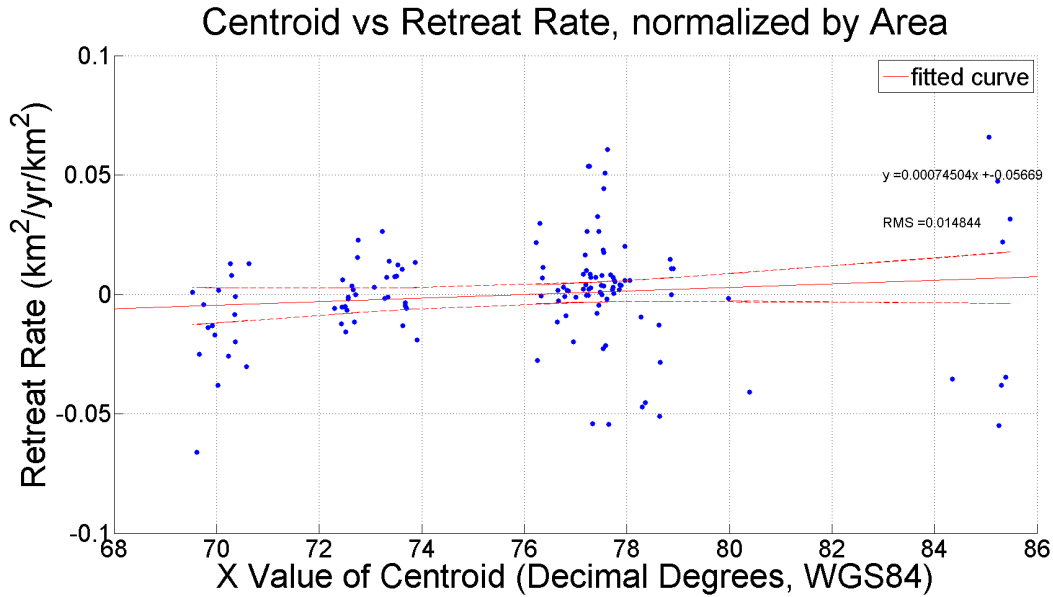


Figure 2.15: Retreat rates of index glaciers across the Tien Shan, showing a slight trend from east to west, with 95% confidence bounds.

Areal Changes in Glaciers Across the Entire Tien Shan

Figure 2.15 illustrates the relationship between spatial location, from east to west, and retreat rates across the entirety of the Tien Shan Range. Though noise in the data is evident, all of the data were generated using the same methodology, and are thus comparable (a t-test performed on the data shows that the trend is significant at the 99 percent confidence level). The one caveat to this is that the author has noted that not all Landsat scenes perform equally; scenes with excess snow, cloud cover, shadows, or other strong spectral signatures may misclassify some glacial areas. An effort was made to choose only the best suited images for this study; as can be seen from the statistical analysis presented above, the glacier outlines are generally accurate, particularly when considered in larger quantities.

Regional bulk retreat rates do not necessarily represent an accurate accounting of

glacial health – for example, high elevation glaciers will generally retreat more slowly than low elevation glaciers. This research has chosen to normalize the retreat rate data by the area of each glacier (Hanshaw and Bookhagen, 2014). When the data are normalized by other indices, such as median slope and elevation, the relationship remains consistent, in that glaciers further to the east in the Tien Shan tend to have retreated further than glaciers to the west, over the period ~1998 - 2013. This agrees well with work by Gardelle et al. (2013), who noted that the Pamir range has experienced slightly positive mass balances over the period 2000-2011. Although this study is focused on areal changes across the Tien Shan, as opposed to the mass balances examined by Gardelle et al. (2013), the author proposes a similar mechanism for the variability in retreat rates: spatial gradients in precipitation. To understand climatic forcing across the range, the author has analyzed several climatic and topographic variables to determine the factors responsible for the variation in retreat rates across the Tien Shan.

Links to Climate Trends

To analyze the impacts of climatic gradients across the study area on glacial behavior, the author created a swath profile along the length of the greater Tien Shan Range.

Topography Across the Tien Shan

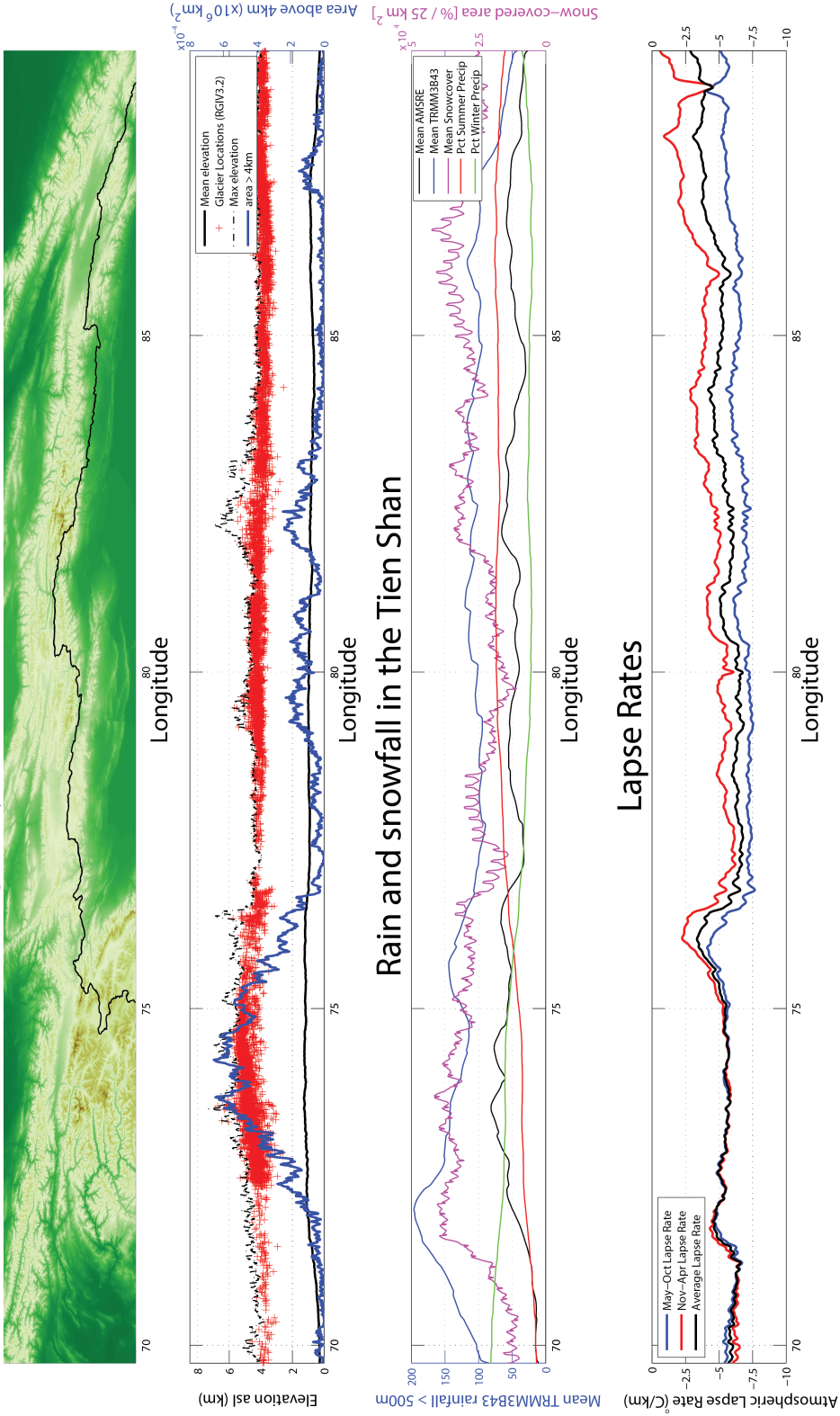


Figure 2.16: Swath Profile along the Tien Shan (swath outline presented in Figure 2.1). (A) Topography of the swath profile, and generalized internal vs external watershed delineation. (B) Mean and Maximum elevation, as well as area above 4000m. Glacier centroids and median elevations plotted in Red. (C) SWE, TRMM precipitation data, and snow covered area. Percentage of summer (red) and winter (green) precipitation from TRMM (1998-2013), on a scale of 0-100. (D) Atmospheric Lapse Rates. Average in Black, Nov-Apr in Red, May-Oct in Blue.

Figures 2.16A and 2.16B show the general topography across the range from the Pamir Knot towards the eastern end of the Tien Shan. The highest elevation areas host the most glaciers, most notably the Pamirs on the west of the swath profile ($\sim 73^{\circ}\text{E}$) and the area surrounding the Inylchek glacier around 80°E . Both SWE and precipitation in general are fairly low throughout the range. Figure 2.16C shows precipitation and snow cover throughout the study area, as well as seasonal precipitation percentages. Winter (November-April) precipitation is dominant up until the eastern edge of the Pamirs, where summer (May-October) precipitation becomes dominant. As glacial response to summer and winter precipitation regimes is different (e.g., Narama et al., 2010), the split between the winter and summer growth regions could also explain some of the east-west variability in glacier retreat.

The atmospheric lapse rates (Figure 2.16D) illustrate generally cold temperatures year round in the Pamirs, and generally higher summer temperatures than winter temperatures throughout the remainder of the Tien Shan. Lapse rates are defined here as the change in atmospheric temperature (in this case, land surface temperature, MODIS product MOD11C) over change in elevation, which gives a temperature gradient across the mountain range. The strongly negative lapse rates in the west indicate colder mountain peaks, which have a strong correlation with larger glacial areas. As lapse rates become less strongly negative towards the east, peaks become slightly warmer, and less glaciated overall. The glaciers to the east are also more likely to be impacted by warming temperatures, as they are situated in generally warmer areas to begin with.

There is a distinct precipitation gradient between the external watersheds and the internal watershed, which drains towards the Taklamakan Desert (generalized into

internal and external components as a black line in Figure 2.16A). The inner basin receives generally less rainfall, and has less snow covered area. Lapse rates are similar in the two regions, primarily due to generally cold temperatures year-round. Unfortunately, TRMM data did not allow the author to effectively measure snowfall, which is a key component in glacier area changes, and there is insufficient data to draw conclusions on whether or not the drainage divide influences glacier retreat rates. It is unlikely that the north-south divide is a strong controlling factor in retreat rates in the region, and that the east-west gradient is a far stronger control.

Discussion of Algorithm Use Cases

Lack of data density and temporal range limits the efficacy of individual glacial analysis. The algorithm presented in this paper was not designed with individual glacial studies in mind, and in many cases, such as in mass balance studies, more accurate manual glacier outlines are necessary. The glacier outlines provided by the algorithm are, however, a useful first pass analysis of glacial area. It is often quicker to digitize only misclassified areas, as opposed to digitizing entire glacier areas by hand (Paul et al., 2013). Paul et al. (2013) also note that for clean ice, automatically derived glacial outlines tend to be more accurate, and it is only in the more difficult debris-covered and shadowed areas that manual digitization becomes preferable.

The algorithm moves a step further and attempts to classify glacier areas as accurately as possible, including debris-covered areas. As can be seen in Figures 2.9 and 2.10, the algorithm performs very well over a large dataset (~700 glaciers). At the scale of watersheds, satellite image footprints, or entire mountain ranges, this is an effective methodology which can be used to generate wide-area glacial statistics, as can be seen

in Figure 2.15. As the method can be easily modified to fit the topographic and glacial setting of any region, it is a powerful tool for analyzing glacier changes over large scales over the period of Landsat TM, ETM+ and OLI coverage. Lastly, as the algorithm performs well on debris-covered areas, it could be used in the place of simple spectral ratios, as it will reduce the time required for manual digitization and correction, as compared to spectral-only methods.

F. Conclusions

This study presents an enhanced glacial classification methodology based on the spectral, topographic, and spatial characteristics of glaciers. The author has additionally applied the algorithm to analyze the character of glaciers across the Tien Shan Range from 1998-2014.

The author's algorithm represents a step forward toward (semi-) automated glacial classification, in that it outperforms spectral-only algorithms in wide use in the glaciology and remote sensing communities. Although it does not completely solve the difficulties associated with debris covered glaciers, it can effectively and rapidly characterize glaciers over a wide area. The first steps of this algorithm, which seek to characterize maximum possible glacier area, are lake delineation, spectral delineation, and topographic filtering. Following these steps, a set of velocity, spatial, and statistical filters are applied to accurately delineate glacier outlines.

Using this newly developed algorithm, the author presents evidence that glacial retreat rates across the Tien Shan show a distinct east to west gradient. Although the difference in retreat rate is slight, it has been shown to be statistically significant, and the author can say with confidence that glaciers closer to the Pamirs (west) are

shrinking less than those closer to the Altai (east). To explain this gradient, the author analyzed climate and topographic data, which reveal a distinct west to east trend in precipitation which matches the general trend in retreat rates. As storms move from the west, they split along the topographic front of the Pamirs, and only the largest storms continue on north into the Tien Shan, where they are topographically trapped, resulting in more precipitation in the west. This split is also evident in precipitation seasonality, where the western regions of the Tien Shan receive more winter precipitation, and less summer precipitation. This implies that the western regions receive both more precipitation overall, and that the majority of this precipitation in glaciated areas falls as snow. As a consequence of increasing temperatures in the region, summer-fed glaciers (towards the east of the study area) will be impacted by longer ablation seasons, shorter accumulation seasons, and shorter periods of high albedo (from fresh snow). All of these factors enhance glacial retreat towards the east of the study area, while having weaker effects on the glaciers towards the west of the study area. In summation, the east-west glacier retreat gradient is driven by a combination of climate and topography.

III. Progress Towards Mass Balance Analysis of the Tien Shan

A. Introduction

Glacier change can be assessed through either areal change (through studies such as the one presented in Chapter 2 of this thesis), or through estimates of mass balance. While areal measurements are useful for tracking changes in glaciers, they are not always a reliable measure of glacier health and are not always accurate enough for studying water resources associated with the cryosphere. However, area measurements are easier to collect over large scales, using satellite imagery, and thus much of the world's glacier inventory is comprised of areal measurements.

Mass balance estimates provide a more holistic view of glacier health by incorporating changes in glacier height, which allows researchers to estimate the amount of water-equivalent stored in a glacier. Throughout the year, glaciers grow and shrink as snowfall accumulates during cold months and ice melts off during warmer months. A glacier's mass balance is a measure of the total mass gained or lost over a fixed period of time. As it is entirely possible for glaciers to thicken while shrinking in area, or thin while growing in area, mass balance estimates are a more accurate and complete measure of glacier health.

The Tien Shan has not received many mass balance estimates, and those estimates were neither spatially extensive nor temporally dense (i.e. Pieczonka et al., 2013). The purpose of this chapter, which is work in progress, is to both increase the density of mass balance estimates and to extend the spatial extent of analyzed glaciers in the Tien

Shan.

B. Data Sources

This study focuses on remotely sensed data and imagery, with additional contributions of hydrologic and in-situ data where available via a long-standing research relationship between the author’s adviser (Dr. Bodo Bookhagen) and several Kyrgyz institutions.

Optical Data

The primary data sources for this investigation are the Landsat and ASTER missions. Landsat’s long time series is essential for assessing areal changes in glaciers over a decadal and annual time scale, and forms the core of the debris-mapping algorithm (presented in Chapter 2 of this thesis). Though additional datasets are required to accurately map debris tongues, mapping would not be possible without the availability of several spectral bands on Landsat.

ASTER stereoscopic imagery at 15m resolution was used to create DEMs of the study region over multiple timesteps. By leveraging a time series of elevation models over a specific glacier area, the author can derive an estimate of volumetric changes. This volumetric data can in turn be used to estimate the amount of water gained or lost over the same time period, and provides key insight into the water budget of a glaciated watershed.

Elevation and Topographic Data

Accurate terrain and elevation data are a prerequisite for assessing volumetric changes in glaciers, and hence any changes in their water storage capacity. The 2000 SRTM DEM serves as a robust baseline elevation measure, as well as a master elevation model with which to correct subsequent DEMs. Several studies have noted that the SRTM

DEM is not always vertically accurate, particularly in mountainous terrain (e.g., Smith and Sandwell, 2003; Berthier et al., 2006; Kääb et al., 2012; Gardelle et al., 2013). This is partially due to over- and under-sampling in steep terrain, and also due to the penetration of radar signals into snow and ice. To control for radar signal penetration, both C and X band SRTM data is used.

Elevation models derived from optical imagery, such as ASTER, often contain sizeable errors, and must be statistically corrected. The author has developed an automated correction technique, modified after Nuth and Kääb (2011), comprised of an image registration step, a tilt correction, an aspect correction, and an elevation correction. Once two DEMs have been accurately co-registered and corrected for vertical agreement, they can provide volumetric change estimates over a defined glacial area.

C. Methods

The first step in estimating glacier mass balance is the derivation of accurate glacial outlines. This was undertaken through use of both the algorithm presented in Chapter 2, and through manual corrections for misclassified areas. Where appropriate, corrections for areal changes are taken into account when estimating mass balances.

DEM Correction

Nuth and Kääb (2011) presented a comprehensive correction methodology involving a tilt correction, aspect correction, elevation correction, and satellite orientation correction. To streamline this algorithm and make it more useful for large datasets, the author has updated and automatized the DEM correction algorithm. Full scripts are provided here as appendices D-I.

Data Preparation

ASTER DEMs are created manually in ENVI by selecting a set of common tie points between 3N (Nadir) and 3B (Backward) images. The SRTM DEM is used as a reference image for accurate geolocation. Tie points are selected based on feature recognizability, lack of snow, and lack of cloudcover. An effort was made to distribute tie points throughout the image, and minimize the RMS error of the tie points (parallax below ~ 1). These steps guarantee that the DEM generated is mostly geographically accurate, although the ASTER DEM is not yet usable for mass balance measurements

Data are prepared by a script which resamples, reprojects, and renames input datasets for easy ingestion by Matlab. The three datasets used in this step are the SRTM DEM, a set of input ASTER DEMs, and a set of corresponding masks. The masks comprise cloud-covered areas, as well as glaciated areas, which are not expected to remain vertically stable over time. Between the 2000 SRTM DEM and subsequent ASTER DEMs, much of the terrain in a given scene is expected to remain stable. Using this stable terrain, a set of corrections can be derived and applied to the entirety of the ASTER DEM, thus removing bias on both stable and unstable terrain. Excluding the unstable terrain improves the statistical correlation over stable terrain by removing areas which are known to be vertically inconsistent between the ‘master’ and ‘slave’ DEMs, and is an essential step in accurately correcting the ASTER DEMs.

Co-registration

The image co-registration step takes the ASTER DEM as a slave input, and the SRTM DEM as a master input. Image registration is performed through an intensity-based

image registration which shifts the slave image in x , y , and z dimensions until the closest match is found. This step is an extension of the algorithm proposed by Nuth and Kääb (2011), and greatly increases the efficacy of the subsequent statistical corrections.

Tilt Correction

The tilt correction as implemented in the correction algorithm chooses low slope areas, as well as areas with a low absolute difference between the SRTM DEM and ASTER DEM, and performs iterative plane fitting to identify and correct relatively too-high and too-low areas. The step is performed multiple times until the absolute difference between the two datasets is minimized. The tilt corrected DEM is then used as the input for an aspect based correction.

Aspect Correction

In complex terrain, such as the study area, slopes of differing aspect will be relatively under- and over-sampled based on their orientation relative to the satellite view angle. To reduce this sampling bias, an aspect correction is applied iteratively to minimize the sampling bias. The aspect correction selects high slope areas (above a given threshold), as well as the difference between the SRTM and ASTER DEM at those points. From these points, a statistically significant sample is taken ($1e5$ pixels), and a sinusoidal curve is fit to the difference vs aspect (Figure 3.1).

Using the curve fitted to the aspect data, a correction was applied to ‘flatten’ the aspect bias until there was no longer a statistically significant correlation between aspect and height bias. Using the aspect corrected DEM, a final elevation correction was applied.

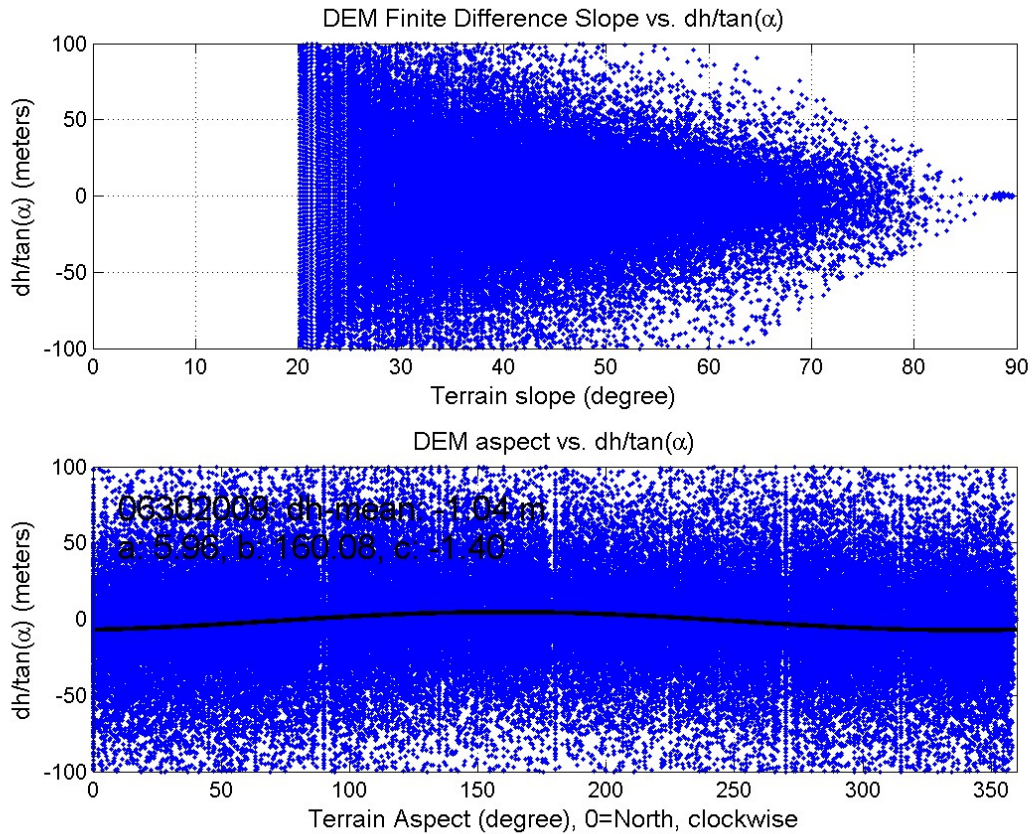


Figure 3.1: Example output of aspect correction script, showing selected height difference vs height difference divided by tangent of slope and aspect vs height difference divided by tangent of slope.

Elevation Correction

Studies have noted a significant bias in ASTER datasets due to uneven spatial distribution of ground control points used in the generation of stereoscopic DEMs (Nuth and Kääb, 2011). This can distort the height scale, and lead to significant offsets, especially at the scales of glacial height changes. To correct for this, a model is fitted to compare the ASTER DEM with the difference between the ASTER DEM and SRTM DEM, to identify any elevation-dependant bias. The resulting correction removes any remaining bias along the z axis. The results of these corrections can be seen in Figure 3.2.

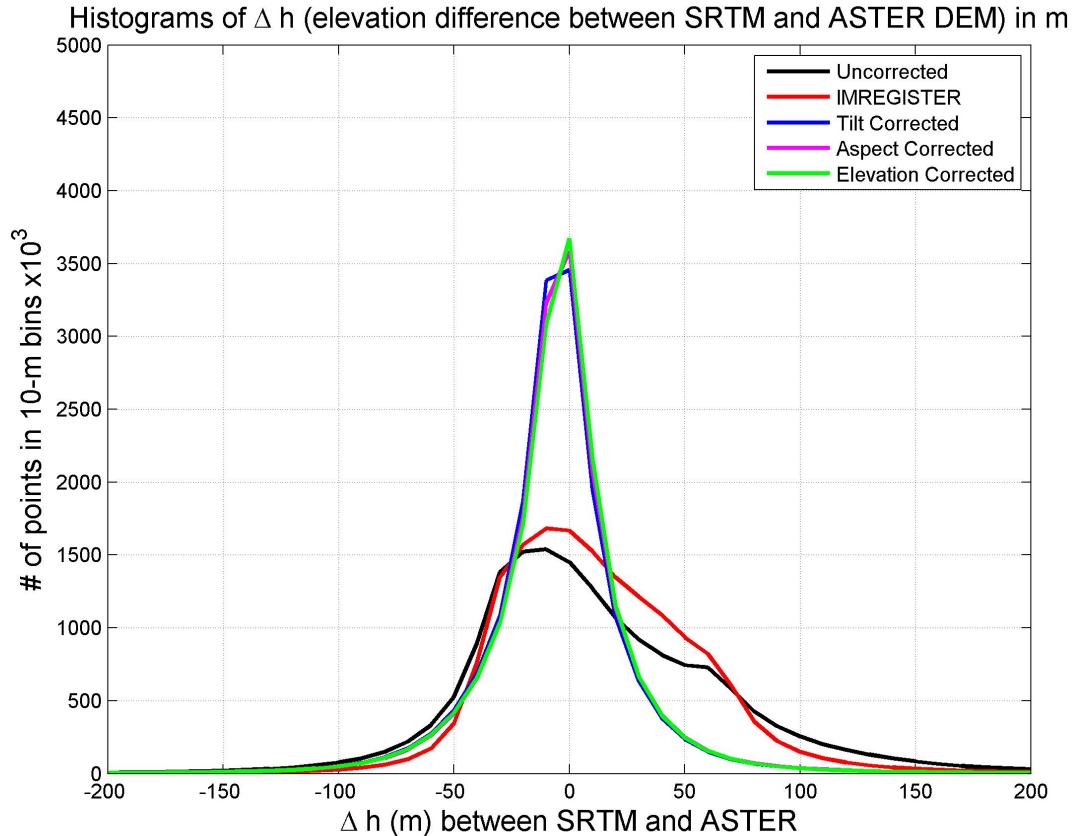


Figure 3.2: Number of points in 10m elevation bins vs the difference in height between ASTER and SRTM DEMs. Black is uncorrected, Red is co-registered, Blue is tilt-corrected, Pink is aspect corrected, and Green is elevation corrected. Illustrates improved agreement between SRTM and ASTER DEMs over stable terrain through stepwise correction, as larger numbers of pixels are well matched between the SRTM and ASTER dataset (Δh around zero).

As can be seen in Figure 3.2, the stepwise corrections increase the agreement between the ‘master’ SRTM DEM and ‘slave’ ASTER DEM. Once the two datasets have been properly co-registered, they can be used for elevation, and thus volume, change estimates. As a final step, the mean bias between the two datasets is calculated, and the bias is added to the ASTER DEM. This ensures that they two datasets are properly matched in the z axis, and that changes in height between datasets can be trusted as being significant (Figure 3.3).

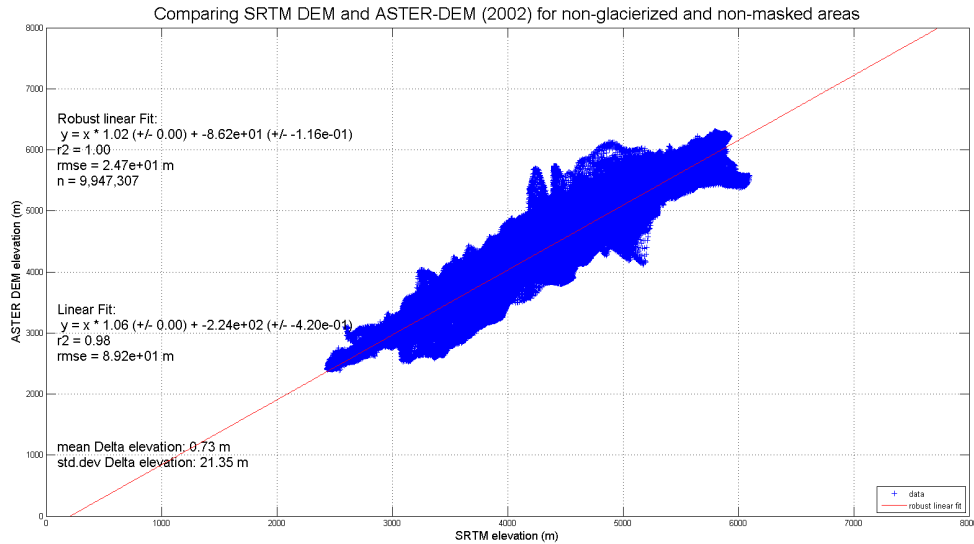


Figure 3.3: Mean bias between SRTM DEM and ASTER DEM.

Mass Balance Estimates

Using the set of corrected ASTER DEMs, as well as the master SRTM DEM, a timeseries of glacier height changes can be calculated. As the research is interested in not only how much the volume of a glacier changes, but where that volume change occurs, rasterized versions of each DEM, bounded by the glacier outlines, are differenced. The results of this preliminary study are presented below.

D. Preliminary Results

To examine the extent of glacial change in the Tien Shan, a pilot study site of one glacier was chosen. This glacier was chosen due to its proximity to relatively large glaciers in the region, and a large number of ASTER scenes which provided a relatively dense volumetric timeseries. A preliminary figure illustrating the spatial gradient across a single glacier can be seen in Figure 3.4.

Figure 3.4 shows general mass gain at the higher areas of the glacier, and mass

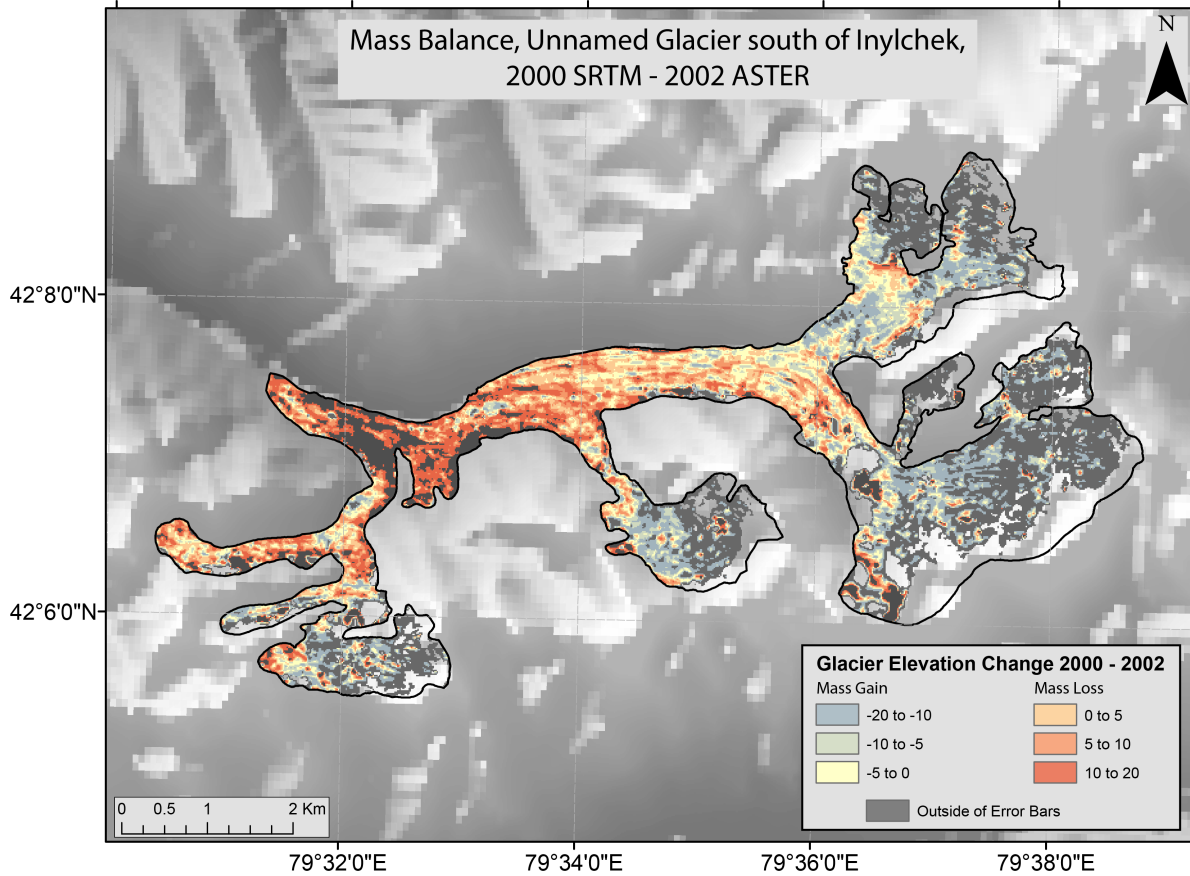


Figure 3.4: Height difference between an example ASTER DEM and the SRTM DEM. Showing general spatial patterns in mass change.

loss towards the glacier tongue. From this perspective, it is possible that the glacier tongue is thinning and retreating while the higher elevation areas of the glacier are actually thickening. Following this result, several more estimates of mass change across glacier elevation distributions were created. Figure 3.5 illustrates the densest timeseries examined, for a medium sized glacier near the Inylchek glacier.

Notably, single-timestep estimates for several other glaciers in the region show a similar pattern. This work has not yet been extended throughout the entirety of the Tien Shan Range, though recent work by Gardelle et al. (2013) has shown a slightly positive mass balance in the Pamir Range, accompanied by a slight decrease in glacier

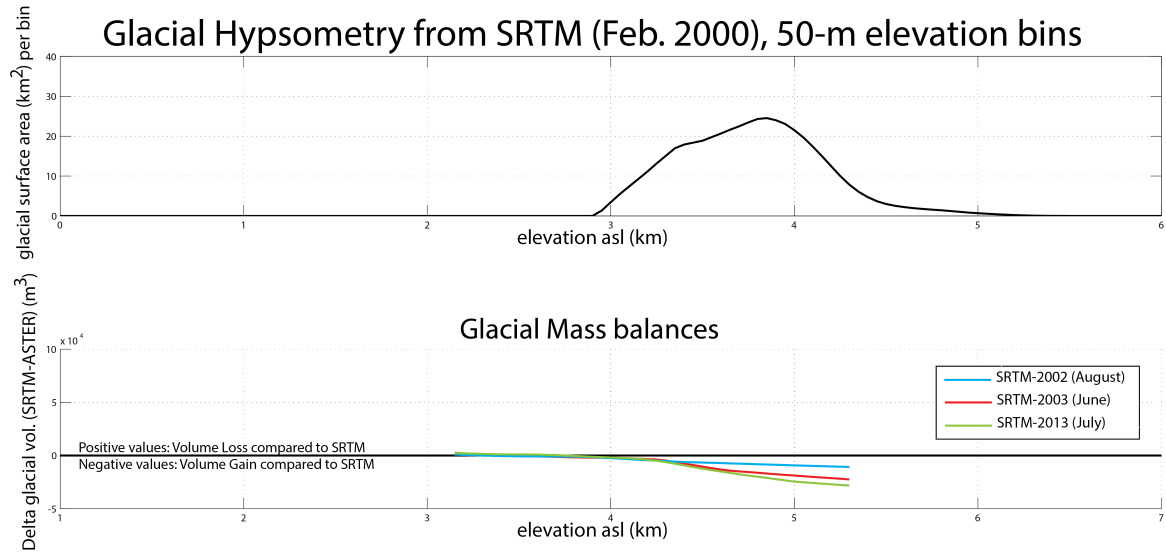


Figure 3.5: Glacier hypsometry, as well as mass balance estimates through time. Shows a consistent trend of mass loss at the tongue and mass gain in the high elevations across all timesteps (Blue - August 2002, Red - June 2003, Green - July 2013)

area ($\sim 1\%$).

E. Preliminary Discussion

As has been discussed above in Chapter 2 of this thesis, there are significant climatic gradients across the greater Tien Shan Range. Though Figure 2.2.16 illustrates the spatial aspects of climate gradients across the Tien Shan, it does not effectively illustrate climate shifts over time. Aizen et al. (1997) analyzed long-term temperature and precipitation trends across the Tien Shan, and showed that across all zones there was statistically significant warming of both summer and annual air temperature (1940-1991). Precipitation trends were generally flat, with the exception of the more western and northern areas of the Tien Shan, though this trend is only exhibited below 2000 meters, as can be seen in Figure 3.6 (Aizen et al., 1997). This is likely at least somewhat influenced by lack of accurate snowfall measurements, particularly at high elevations. However, these trends generally agree that the western areas of the Tien Shan towards the Pamir are

receiving more moisture in general.

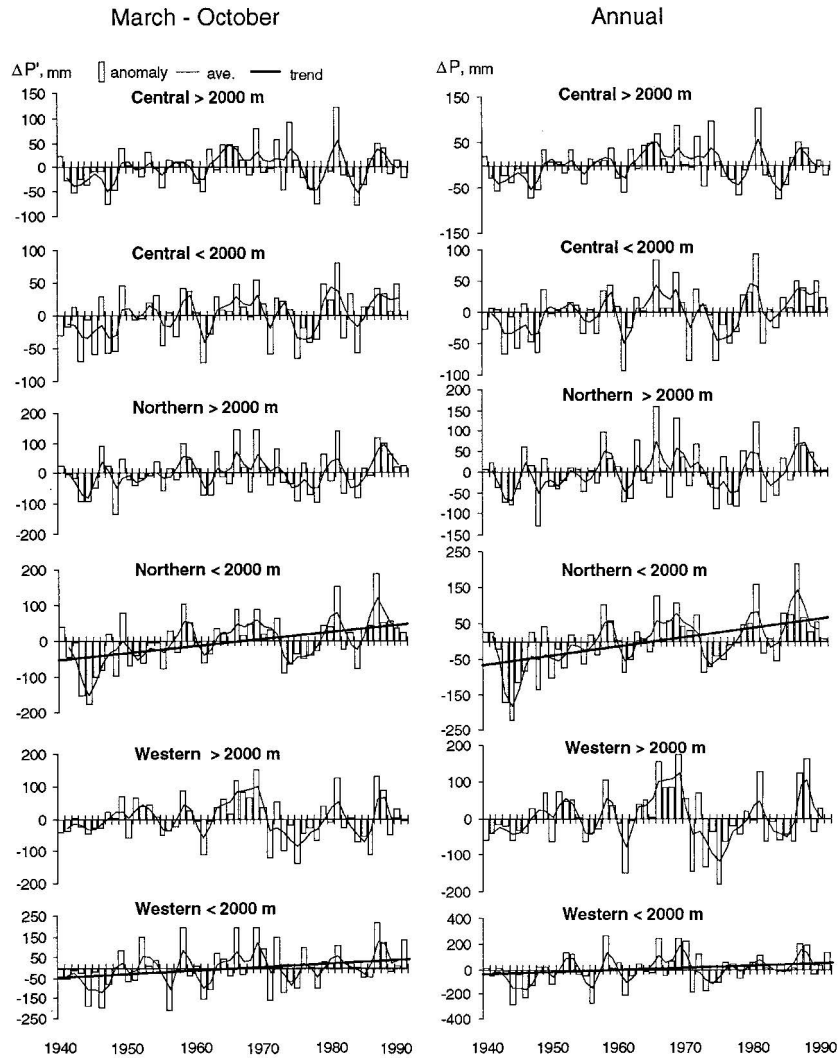


Figure 3.6: Precipitation trends (1940-1991) for station data across the Tien Shan. Generally flat trends, with the exception of low elevation areas in the north and west. Modified from Aizen et al. (1997).

One possibility which would explain both the glacial retreat and glacial mass balance estimates presented in this thesis is that warming throughout the Tien Shan has led to glacier loss at low elevations. This would work to expand the ablation area, particularly in low elevation glaciers, and could account for the observed glacier retreat. Glacier thickening at high elevations is less temperature dependent, because temperatures remain low, but more reliant on precipitation. As precipitation has been shown to

slightly increase in the west, and stay relatively constant in the east, the author would expect slightly positive mass balances in the west and central regions of the Tien Shan, and constant or negative mass balances towards the east. Preliminary work, combined with the work of Gardelle et al. (2013), has shown that there are positive mass balances, at least at high elevations, in the western and central regions of the Tien Shan. Extrapolating from this, it is likely that further analysis of glaciers in the eastern regions of the Tien Shan will result in negative mass balance measurements. This, however, will require further work outside of the scope of this thesis.

F. Future Work

Although this study presents some interesting results, they should be considered preliminary until further analysis can be undertaken. The author aims to incorporate three more data sources in order to further constrain the character of glaciers in the Tien Shan.

First, Ice, Cloud, and Land Elevation Satellite (ICESat) data will be used to constrain accurate point elevations on selected glaciers, dependent on data coverage. The ICESat mission can provide very accurate (cm-to-dm-scale) measures of elevation at point locations (Schutz et al., 2005). These point locations, although spatially diverse and temporally separated, provide valuable controls on ice elevations. Preliminary analysis indicates that there are enough data points throughout the study region for ICESat to provide a valuable control on elevation estimates derived from optical and radar imagery. As the values are spatially limited, they are not appropriate for mass balance estimates. When combined with ASTER DEM data, however, they provide useful high-resolution control points.

Second, Synthetic Aperture Radar Interferometry (InSAR) will be used for three purposes: to (1) accurately identify lakes at high spatial resolution, (2) create interferograms to map glacial area via incoherence, and (3) generate high-resolution DEMs from interferograms with a temporal baseline of less than 2 weeks. Pro- and peri-glacial lakes, which provide a highly visible signal in SAR data, are crucial seed points for the glacier algorithm presented in Chapter 2 of this thesis. As glaciers move much faster than the surrounding terrain, it is possible to remove low-velocity areas and return a map of only ‘incoherent’ areas. This provides similar utility to the image cross correlation presented above, though at a higher spatial resolution. The high spatial resolution will also be leveraged in the mass balance estimates, as low temporal baseline SAR images can be used for DEM generation, which will help enhance the temporal and spatial resolution of the mass balance estimates.

Finally, MODIS surface reflectances will be used to track surface reflectance changes on glaciers. This is useful for identifying changes in snow cover (fresh vs. old snow), and delineating ice and snow. On larger glaciers, it can also be used to estimate equilibrium line altitudes (ELAs), which are a key measure of glacier health. If ELAs are moving upwards, either precipitation is falling or temperatures are rising. When combined with mass balance estimates over the whole glacier, ELAs can help untangle the true drivers of glacial change in the region. As the mechanisms driving glacial change in the Tien Shan are still poorly understood, creating linkages between seasonal snow cover and glacier snowlines, and their resulting effects on glacial extents and mass balances, is essential in understanding climate change impacts in the Tien Shan.

References

- Aizen, V., Aizen, E., Kuzmichonok, V., 2007a. Glaciers and hydrological changes in the tien shan: simulation and prediction. *Environmental Research Letters* 2 (4), 045019.
- Aizen, V. B., Aizen, E. M., Melack, J. M., Dozier, J., 1997. Climatic and hydrologic changes in the tien shan, central asia. *Journal of Climate* 10 (6), 1393–1404.
- Aizen, V. B., Kuzmichenok, V. A., Surazakov, A. B., Aizen, E. M., 2006. Glacier changes in the central and northern tien shan during the last 140 years based on surface and remote-sensing data. *Annals of Glaciology* 43 (1), 202–213.
- Aizen, V. B., Kuzmichenok, V. A., Surazakov, A. B., Aizen, E. M., 2007b. Glacier changes in the tien shan as determined from topographic and remotely sensed data. *Global and Planetary Change* 56 (3), 328–340.
- Anderson, D. M., Overpeck, J. T., Gupta, A. K., 2002. Increase in the asian southwest monsoon during the past four centuries. *Science* 297 (5581), 596–599.
- Arendt, A., Bolch, T., Cogley, J., Gardner, A., Hagen, J., Hock, R., Kaser, G., Pfeffer, W., Moholdt, G., Paul, F., et al., 2012. Randolph glacier inventory [v2. 0]: A dataset of global glacier outlines. global land ice measurements from space, boulder colorado, usa. Digital Media.
- Armstrong, R., Raup, B., Khalsa, S., Barry, R., Kargel, J., Helm, C., Kieffer, H., 2005. Glims glacier database. National Snow and Ice Data Center, Boulder, Colorado, USA.
- Barlow, M., Wheeler, M., Lyon, B., Cullen, H., 2005. Modulation of daily precipitation over southwest asia by the madden–julian oscillation. *Monthly weather review* 133 (12).
- Barros, A. P., Kim, G., Williams, E., Nesbitt, S. W., et al., 2004. Probing orographic controls in the himalayas during the monsoon using satellite imagery. *Natural Hazards and Earth System Science* 4 (1), 29–51.
- Berthier, E., Arnaud, Y., Kumar, R., Ahmad, S., Wagnon, P., Chevallier, P., 2007. Remote sensing estimates of glacier mass balances in the himachal pradesh (western himalaya, india). *Remote Sensing of Environment* 108 (3), 327–338.
- Berthier, E., Arnaud, Y., Vincent, C., Remy, F., 2006. Biases of srtm in high-mountain areas: Implications for the monitoring of glacier volume changes. *Geophysical Research Letters* 33 (8).
- Bohner, J., 2006. General climatic controls and topoclimatic variations in central and high asia. *Boreas* 35 (2), 279–295.
- Bolch, T., Kulkarni, A., Käab, A., Huggel, C., Paul, F., Cogley, J., Frey, H., Kargel, J., Fujita, K., Scheel, M., et al., 2012. The state and fate of himalayan glaciers. *Science* 336 (6079), 310–314.

- Bolch, T., Peters, J., Yegorov, A., Pradhan, B., Buchroithner, M., Blagoveshchensky, V., 2011. Identification of potentially dangerous glacial lakes in the northern tian shan. In: *Terrigenous Mass Movements*. Springer, pp. 369–398.
- Bonasoni, P., Laj, P., Marinoni, A., Sprenger, M., Angelini, F., Arduini, J., Bonafe, U., Calzolari, F., Colombo, T., Decesari, S., et al., 2010. Atmospheric brown clouds in the himalayas: first two years of continuous observations at the nepal-climate observatory at pyramid (5079 m). *Atmospheric Chemistry and Physics Discussions* 10 (2), 4823–4885.
- Bookhagen, B., 2010. Appearance of extreme monsoonal rainfall events and their impact on erosion in the himalaya. *Geomatics, Natural Hazards and Risk* 1 (1), 37–50.
- Bookhagen, B., Burbank, D. W., 2010. Toward a complete himalayan hydrological budget: Spatiotemporal distribution of snowmelt and rainfall and their impact on river discharge. *Journal of Geophysical Research: Earth Surface* (2003–2012) 115 (F3).
- Bookhagen, B., Thiede, R. C., Strecker, M. R., 2005. Abnormal monsoon years and their control on erosion and sediment flux in the high, arid northwest himalaya. *Earth and Planetary Science Letters* 231 (1), 131–146.
- Boos, W. R., Kuang, Z., 2010. Dominant control of the south asian monsoon by orographic insulation versus plateau heating. *Nature* 463 (7278), 218–222.
- Cannon, F., Carvalho, L. M. V., Jones, C., Bookhagen, B., in review. Multi-annual variations in winter westerly disturbance activity affecting the himalaya. *Climate Dynamics*.
- Charney, J., Quirk, W. J., Chow, S.-H., Kornfield, J., 1977. A comparative study of the effects of albedo change on drought in semi-arid regions. *Journal of the Atmospheric Sciences* 34 (9), 1366–1385.
- Charney, J. G., 1975. Dynamics of deserts and drought in the sahel. *Quarterly Journal of the Royal Meteorological Society* 101 (428), 193–202.
- Cohen, J., Entekhabi, D., 1999. Eurasian snow cover variability and northern hemisphere climate predictability. *Geophysical Research Letters* 26 (3), 345–348.
- Colombi, A., De Michele, C., Pepe, M., Rampini, A., 2007. Estimation of daily mean air temperature from modis lst in alpine areas. *EARSeL eProceedings* 6 (1), 38–46.
- Dozier, J., 1989. Spectral signature of alpine snow cover from the landsat thematic mapper. *Remote sensing of Environment* 28, 9–22.
- Dozier, J., Painter, T. H., 2004. Multispectral and hyperspectral remote sensing of alpine snow properties. *Annu. Rev. Earth Planet. Sci.* 32, 465–494.
- Duan, Z., Bastiaanssen, W., 2013. First results from version 7 trmm 3b43 precipitation product in combination with a new downscaling–calibration procedure. *Remote Sensing of Environment* 131, 1–13.

- Farr, T. G., Kobrick, M., 2000. Shuttle radar topography mission produces a wealth of data. *Eos, Transactions American Geophysical Union* 81 (48), 583–585.
- Fasullo, J., Webster, P., 2003. A hydrological definition of indian monsoon onset and withdrawal. *Journal of Climate* 16 (19).
- Flohn, H., 1957. Large-scale aspects of the” summer monsoon” in south and east asia. *J. Meteor. Soc. Japan* 75, 180–186.
- Folland, C. K., Knight, J., Linderholm, H. W., Fereday, D., Ineson, S., Hurrell, J. W., 2008. The summer north atlantic oscillation: Past, present, and future. *Journal of Climate* 22 (5).
- Fu, C., 2003. Potential impacts of human-induced land cover change on east asia monsoon. *Global and Planetary Change* 37 (3), 219–229.
- Gadgil, S., 2003. The indian monsoon and its variability. *Annual Review of Earth and Planetary Sciences* 31 (1), 429–467.
- Gao, B.-C., 1996. NdwI - a normalized difference water index for remote sensing of vegetation liquid water from space. *Remote sensing of environment* 58 (3), 257–266.
- Gao, F., Masek, J., Wolfe, R. E., 2009. Automated registration and orthorectification package for landsat and landsat-like data processing. *Journal of Applied Remote Sensing* 3 (1), 033515–033515.
- Gardelle, J., Arnaud, Y., Berthier, E., 2011. Contrasted evolution of glacial lakes along the hindu kush himalaya mountain range between 1990 and 2009. *Global and Planetary Change* 75 (1), 47–55.
- Gardelle, J., Berthier, E., Arnaud, Y., 2012. Slight mass gain of karakoram glaciers in the early twenty-first century. *Nature geoscience* 5 (5), 322–325.
- Gardelle, J., Berthier, E., Arnaud, Y., Kääb, A., 2013. Region-wide glacier mass balances over the pamir-karakoram-himalaya during 1999–2011. *Cryosphere* 7 (4).
- Gautam, R., Hsu, N., Lau, K.-M., Kafatos, M., 2009. Aerosol and rainfall variability over the indian monsoon region: distributions, trends and coupling. In: *Annales Geophysicae*. Vol. 27. Copernicus GmbH, pp. 3691–3703.
- Ghosh, S., Das, D., Kao, S.-C., Ganguly, A. R., 2012. Lack of uniform trends but increasing spatial variability in observed indian rainfall extremes. *Nature Climate Change* 2 (2), 86–91.
- Hall, D., Ormsby, J., Bindschadler, R., Siddalingaiah, H., 1987. Characterization of snow and ice reflectance zones on glaciers using landsat thematic mapper data. *Ann. Glaciol* 9, 1–5.

- Hall, D. K., G. A. R., Salomonson, V. V., 2006. Modis/terra snow cover daily l3 global 0.05 deg cmg v005, mod10c1. National Snow and Ice Data Center, Boulder, Colorado, USA.
- Hanshaw, M. N., Bookhagen, B., 2014. Glacial areas, lake areas, and snowlines from 1975 to 2012: status of the cordillera vilcanota, including the quelccaya ice cap, northern central andes, peru. *Cryosphere Discussions* 7 (1).
- Heid, T., Kääb, A., 2012. Evaluation of existing image matching methods for deriving glacier surface displacements globally from optical satellite imagery. *Remote Sensing of Environment* 118, 339–355.
- Hewitt, K., 2005. The karakoram anomaly? glacier expansion and the ‘elevation effect,’ karakoram himalaya. *Mountain Research and Development* 25 (4), 332–340.
- Hewitt, K., 2011. Glacier change, concentration, and elevation effects in the karakoram himalaya, upper indus basin. *Mountain Research and Development* 31 (3), 188–200.
- Hewitt, K., 2014. *Glaciers of the Karakoram Himalaya: glacial environments, processes, hazards and resources*. Springer.
- Huffman, G. J., Adler, R. F., Bolvin, D. T., Nelkin, E. J., 2010. The trmm multi-satellite precipitation analysis (tampa): Quasi-global, multiyear, combined-sensor precipitation estimates at fine scales. In: *Journal of Hydrometeorology*. Vol. 8(1). Springer, doi:10.1175/jhm560.1., pp. 38–55.
- Huffman, G. J., Bolvin, D. T., 2007. Trmm and other data precipitation data set documentation. Laboratory for Atmospheres, NASA Goddard Space Flight Center and Science Systems and Applications, Inc.[WWW document] Available at: ftp://meso-a.gsfc.nasa.gov/pub/trmmdocs/3B42_3B43_doc.pdf (accessed 23 March 2012).
- Huggel, C., Kääb, A., Haeberli, W., Teyssere, P., Paul, F., 2002. Remote sensing based assessment of hazards from glacier lake outbursts: a case study in the swiss alps. *Canadian Geotechnical Journal* 39 (2), 316–330.
- Jacob, T., Wahr, J., Pfeffer, W. T., Swenson, S., 2012. Recent contributions of glaciers and ice caps to sea level rise. *Nature* 482 (7386), 514–518.
- Kääb, A., 2002. Monitoring high-mountain terrain deformation from repeated air-and spaceborne optical data: examples using digital aerial imagery and aster data. *ISPRS Journal of Photogrammetry and remote sensing* 57 (1), 39–52.
- Kääb, A., Berthier, E., Nuth, C., Gardelle, J., Arnaud, Y., 2012. Contrasting patterns of early twenty-first-century glacier mass change in the himalayas. *Nature* 488 (7412), 495–498.
- Kehrwald, N. M., Thompson, L. G., Tandong, Y., Mosley-Thompson, E., Schotterer, U., Alifimov, V., Beer, J., Eikenberg, J., Davis, M. E., 2008. Mass loss on himalayan glacier endangers water resources. *Geophysical Research Letters* 35 (22).

- Kitoh, A., Endo, H., Krishna Kumar, K., Cavalcanti, I. F., Goswami, P., Zhou, T., 2013. Monsoons in a changing world: a regional perspective in a global context. *Journal of Geophysical Research: Atmospheres* 118 (8), 3053–3065.
- Lang, T. J., Barros, A. P., 2004. Winter storms in the central himalayas. *Journal of the Meteorological Society of Japan* 82 (3), 829–844.
- Lau, K., Kim, M., Kim, K., 2006. Asian summer monsoon anomalies induced by aerosol direct forcing: the role of the tibetan plateau. *Climate Dynamics* 26 (7-8), 855–864.
- Lee, E., Sacks, W. J., Chase, T. N., Foley, J. A., 2011. Simulated impacts of irrigation on the atmospheric circulation over asia. *Journal of Geophysical Research: Atmospheres* (1984–2012) 116 (D8).
- Lehner, B., Verdin, K., Jarvis, A., 2008. New global hydrography derived from spaceborne elevation data. *EOS, Transactions American Geophysical Union* 89 (10), 93–94.
- Leprince, S., Ayoub, F., Klingert, Y., Avouac, J.-P., 2007. Co-registration of optically sensed images and correlation (cosi-corr): An operational methodology for ground deformation measurements. In: *Geoscience and Remote Sensing Symposium, 2007. IGARSS 2007. IEEE International. IEEE*, pp. 1943–1946.
- Linderholm, H. W., Ou, T., Jeong, J.-H., Folland, C. K., Gong, D., Liu, H., Liu, Y., Chen, D., 2011. Interannual teleconnections between the summer north atlantic oscillation and the east asian summer monsoon. *Journal of Geophysical Research: Atmospheres* (1984–2012) 116 (D13).
- Lioubimtseva, E., Cole, R., Adams, J., Kapustin, G., 2005. Impacts of climate and land-cover changes in arid lands of central asia. *Journal of Arid Environments* 62 (2), 285–308.
- Lioubimtseva, E., Henebry, G. M., 2009. Climate and environmental change in arid central asia: Impacts, vulnerability, and adaptations. *Journal of Arid Environments* 73 (11), 963–977.
- Magagi, R., Barros, A. P., 2004. Estimation of latent heating of rainfall during the onset of the indian monsoon using trmm pr and radiosonde data. *Journal of Applied Meteorology* 43 (2).
- Malik, N., Bookhagen, B., Marwan, N., Kurths, J., 2011. Analysis of spatial and temporal extreme monsoonal rainfall over south asia using complex networks. *Climate dynamics* 39 (3-4), 971–987.
- May, W., 2004. Potential future changes in the indian summer monsoon due to greenhouse warming: analysis of mechanisms in a global time-slice experiment. *Climate Dynamics* 22 (4), 389–414.
- May, W., 2011. The sensitivity of the indian summer monsoon to a global warming of 2°C with respect to pre-industrial times. *Climate Dynamics* 37 (9-10), 1843–1868.

- Menon, A., Levermann, A., Schewe, J., 2013. Enhanced future variability during india's rainy season. *Geophysical Research Letters* 40 (12), 3242–3247.
- Mölg, T., Maussion, F., Yang, W., Scherer, D., 2012. The footprint of asian monsoon dynamics in the mass and energy balance of a tibetan glacier. *The Cryosphere Discussions* 6 (4), 3243–3286.
- Mushkin, A., Gillespie, A., 2011. Using aster stereo images to quantify surface roughness. In: *Land Remote Sensing and Global Environmental Change*. Springer, pp. 463–481.
- Mushkin, A., Gillespie, A., Danilina, I., O'Neal, M., Pietro, L., Abbott, E., Balick, L., 2006. Using sub-pixel roughness estimates from aster stereo images to compensate for roughness effects in the thermal infrared. In: *RAQRS II: 2nd International Symposium on Recent Advances in Quantitative Remote Sensing*.
- Narama, C., Kääh, A., Duishonakunov, M., Abdrakhmatov, K., 2010. Spatial variability of recent glacier area changes in the tien shan mountains, central asia, using glaciers (~1970), landsat (~2000), and alos (~2007) satellite data. *Global and Planetary Change* 71 (1), 42–54.
- Narama, C., Shimamura, Y., Nakayama, D., Abdrakhmatov, K., 2006. Recent changes of glacier coverage in the western terskey-alatoo range, kyrgyz republic, using corona and landsat. *Annals of Glaciology* 43 (1), 223–229.
- Nie, Y., Liu, Q., Liu, S., 2013. Glacial lake expansion in the central himalayas by landsat images, 1990–2010. *PloS one* 8 (12), e83973.
- Nuth, C., Kääh, A., 2011. Co-registration and bias corrections of satellite elevation data sets for quantifying glacier thickness change. *The Cryosphere* 5 (1), 271–290.
- Oerlemans, J., 2005. Extracting a climate signal from 169 glacier records. *Science* 308 (5722), 675–677.
- Overpeck, J., Anderson, D., Trumbore, S., Prell, W., 1996. The southwest indian monsoon over the last 18 000 years. *Climate Dynamics* 12 (3), 213–225.
- Painter, T. H., Dozier, J., Roberts, D. A., Davis, R. E., Green, R. O., 2003. Retrieval of subpixel snow-covered area and grain size from imaging spectrometer data. *Remote Sensing of Environment* 85 (1), 64–77.
- Painter, T. H., Rittger, K., McKenzie, C., Slaughter, P., Davis, R. E., Dozier, J., 2009. Retrieval of subpixel snow covered area, grain size, and albedo from modis. *Remote Sensing of Environment* 113 (4), 868–879.
- Palazzi, E., Hardenberg, J., Provenzale, A., 2013. Precipitation in the hindu-kush karakoram himalaya: Observations and future scenarios. *Journal of Geophysical Research: Atmospheres* 118 (1), 85–100.

- Paul, F., 2002. Changes in glacier area in tyrol, austria, between 1969 and 1992 derived from landsat 5 thematic mapper and australian glacier inventory data. *International Journal of Remote Sensing* 23 (4), 787–799.
- Paul, F., Barrand, N., Baumann, S., Berthier, E., Bolch, T., Casey, K., Frey, H., Joshi, S., Konovalov, V., Bris, R. L., et al., 2013. On the accuracy of glacier outlines derived from remote-sensing data. *Annals of Glaciology* 54 (63), 171–182.
- Paul, F., Barry, R., Cogley, J., 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., Käab, 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., Kaab, A., Maisch, M., Kellenberger, T., Haeberli, W., 2002. The new remote-sensing-derived swiss glacier inventory: I. methods. *Annals of Glaciology* 34 (1), 355–361.
- Pfeffer, W. T., Arendt, A., Bliss, A., Bolch, T., Cogley, J., Gardner, A., Hagen, J., Hock, R., Kaser, G., Kienholz, C., Miles, E., Moholdt, G., Mölg, Paul, F., Radić, V., Rastner, P., Raup, B., Rich, J., Sharp, M., Consortium, T. R., 2014. The randolph glacier inventory: a globally complete inventory of glaciers. *Journal of Glaciology* 60 (221), 537–552.
- Pieczonka, T., Bolch, T., Junfeng, W., Shiyin, L., 2013. Heterogeneous mass loss of glaciers in the aksu-tarim catchment (central tien shan) revealed by 1976 kh-9 hexagon and 2009 spot-5 stereo imagery. *Remote Sensing of Environment* 130, 233–244.
- Quincey, D., Richardson, S., Luckman, A., Lucas, R., Reynolds, J., Hambrey, M., Glasser, N., 2007. Early recognition of glacial lake hazards in the himalaya using remote sensing datasets. *Global and Planetary Change* 56 (1), 137–152.
- Racoviteanu, A. E., Arnaud, Y., Williams, M. W., Ordonez, J., 2008a. Decadal changes in glacier parameters in the cordillera blanca, peru, derived from remote sensing. *Journal of Glaciology* 54 (186), 499–510.
- 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.
- Racoviteanu, A. E., Williams, M. W., Barry, R. G., 2008b. Optical remote sensing of glacier characteristics: a review with focus on the himalaya. *Sensors* 8 (5), 3355–3383.
- Ramanathan, V., Carmichael, G., 2008. Global and regional climate changes due to black carbon. *Nature geoscience* 1 (4), 221–227.

- Ramanathan, V., Chung, C., Kim, D., Bettge, T., Buja, L., Kiehl, J., Washington, W., Fu, Q., Sikka, D., Wild, M., 2005. Atmospheric brown clouds: Impacts on south asian climate and hydrological cycle. *Proceedings of the National Academy of Sciences of the United States of America* 102 (15), 5326–5333.
- Ramanathan, V., Ramana, M. V., Roberts, G., Kim, D., Corrigan, C., Chung, C., Winker, D., 2007. Warming trends in asia amplified by brown cloud solar absorption. *Nature* 448 (7153), 575–578.
- Raup, B., Kääb, A., Kargel, J. S., Bishop, M. P., Hamilton, G., Lee, E., Paul, F., Rau, F., Soltesz, D., Khalsa, S. J. S., et al., 2007. Remote sensing and gis technology in the global land ice measurements from space (glims) project. *Computers & Geosciences* 33 (1), 104–125.
- Scambos, T. A., Dutkiewicz, M. J., Wilson, J. C., Bindshadler, R. A., 1992. Application of image cross-correlation to the measurement of glacier velocity using satellite image data. *Remote Sensing of Environment* 42 (3), 177–186.
- 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.
- Schutz, B., Zwally, H., Shuman, C., Hancock, D., DiMarzio, J., 2005. Overview of the icesat mission. *Geophysical Research Letters* 32 (21).
- Singh, D., Tsiang, M., Rajaratnam, B., Diffenbaugh, N. S., 2014. Observed changes in extreme wet and dry spells during the south asian summer monsoon season. *Nature Climate Change*.
- Smith, B., Sandwell, D., 2003. Accuracy and resolution of shuttle radar topography mission data. *Geophysical Research Letters* 30 (9).
- Sorg, A., Bolch, T., Stoffel, M., Solomina, O., Beniston, M., 2012. Climate change impacts on glaciers and runoff in tien shan (central asia). *Nature Climate Change* 2 (10), 725–731.
- Stocker, D. Q., 2013. Climate change 2013: The physical science basis. Working Group I Contribution to the Fifth Assessment Report of the Intergovernmental Panel on Climate Change, Summary for Policymakers, IPCC.
- Tedesco, M., Kelly, R., Foster, J., Chang, A., 2004. Amsr-e/aqua daily l3 global snow water equivalent ease-grids v002. National Snow and Ice Data Center: Boulder, CO.
- Tedesco, M., Narvekar, P. S., 2010. Assessment of the nasa amsr-e swe product. *Selected Topics in Applied Earth Observations and Remote Sensing, IEEE Journal of* 3 (1), 141–159.
- Turner, A. G., Annamalai, H., 2012. Climate change and the south asian summer monsoon. *Nature Climate Change* 2 (8), 587–595.

- Wan, Z., 2008. New refinements and validation of the modis land-surface temperature/emissivity products. *Remote Sensing of Environment* 112 (1), 59–74.
- Wang, K., Wan, Z., Wang, P., Sparrow, M., Liu, J., Haginoya, S., 2007. Evaluation and improvement of the modis land surface temperature/emissivity products using ground-based measurements at a semi-desert site on the western tibetan plateau. *International Journal of Remote Sensing* 28 (11), 2549–2565.
- Wang, S., Zhang, M., Pepin, N., Li, Z., Sun, M., Huang, X., Wang, Q., 2014. Recent changes in freezing level heights in high asia and their impact on glacier changes. *Journal of Geophysical Research: Atmospheres*.
- Webster, P. J., 1983. Mechanisms of monsoon low-frequency variability: Surface hydrological effects. *Journal of the atmospheric Sciences* 40 (9), 2110–2124.
- Worni, R., Huggel, C., Stoffel, M., 2013. Glacial lakes in the indian himalayas - from an area-wide glacial lake inventory to on-site and modeling based risk assessment of critical glacial lakes. *Science of the Total Environment* 468, S71–S84.
- Xu, J., Grumbine, R. E., Shrestha, A., Eriksson, M., Yang, X., Wang, Y., Wilkes, A., 2009. The melting himalayas: cascading effects of climate change on water, biodiversity, and livelihoods. *Conservation Biology* 23 (3), 520–530.
- Xue, Y., Juang, H.-M., Li, W.-P., Prince, S., DeFries, R., Jiao, Y., Vasic, R., 2004. Role of land surface processes in monsoon development: East asia and west africa. *Journal of Geophysical Research: Atmospheres* (1984–2012) 109 (D3).
- Yoo, S.-H., Yang, S., Ho, C.-H., 2006. Variability of the indian ocean sea surface temperature and its impacts on asian-australian monsoon climate. *Journal of Geophysical Research: Atmospheres* (1984–2012) 111 (D3).

Appendix A. Python Code - Glacier Mapping

```
## This script takes Landsat data and performs automated
  classifications of glaciated areas and lake areas.
## Wrapper to two Matlab scripts (Glacier_Extraction,
  Late_Extraction)
## Created by Taylor Smith, June 2014

#Import Modules
print "Loading modules..."
import arcpy, sys, os, string, csv, numpy, glob
import datetime, solar, time, subprocess, math, traceback
#All modules are open source with the exception of ArcPy

arcpy.env.overwriteOutput = True #Set the script to be able
  to overwrite files
arcpy.CheckOutExtension("Spatial") #Access Spatial Analyst
  extension

basepath = "" # This is the base directory path. All other
  file paths are relative to this

def conf_reader(config_file):
    """Reads configuration files and returns a list of
    strings"""

    f = open(config_file, 'r')
    conf_list = []
    for line in f:
        raw = line.split('#')
        data = raw[0]
        new = data.replace('\r\n', '')
        new_2 = new.replace(' ', '')
        conf_list.append(new_2)
    f.close()
    return conf_list

if os.path.exists(basepath + "\\Extract.conf"):
    #Read config file to set up input datasets
    conf = conf_reader(basepath + "\\Extract.conf")
    Asia_SRTM = conf[0] #SRTM (WGS84)
    Asia_Slope = conf[1] #Pre-calculated Slope map from SRTM
    , or NULL to calc during script
    Rivers = conf[2] #Pre-calculated Rivers with 200m buffer
```

```

    (Tif Format)
TraceLakes = conf[3] #Training Lake TIFF FILE. SHP->TIFF
    Conversion is highly error prone
TraceDebris = conf[4] #Training Debris TIFF FILE. SHP->
    TIFF Conversion is highly error prone
Velocitydir = conf[5] # Points to directory of
    Velocities for each zone (choose by path/row, date)
prjdir = conf[6] #Output projection directory

def LandsatMetadataRead(filename):
    """Parses the metadata associated with Landsat images (
        _MTL.txt). Assumes that the metadata files are stored
        in the same directory
    as the image files (default). It then creates slope and
        hillshade images from a clipped DEM (SRTM by Default)
    ."""

    metafile = Path_Raw + filename + '\\\ ' + filename + '_MTL
        .txt'
    Sensor = filename[0:3] #Extract the sensor (LE7,LT4/5,
        LC8)
    f = open(metafile, 'r')
    for line in f: #Parse the XML structure for necessary
        metadata
        if 'DATE_ACQUIRED' in line:
            date = line.split('=')[1]
            datelist = date.split('-')
            year = datelist[0].strip(' ')
            month = datelist[1]
            day1 = datelist[2]
            day = day1.strip('\n')
        elif 'SUN_AZIMUTH' in line:
            Sun_azi = float(line.split('=')[1])
        elif 'SUN_ELEVATION' in line:
            Sun_elev = float(line.split('=')[1])
        elif 'SCENE_CENTER_TIME' in line:
            timetop = line.split('=')[1]
            t = timetop.split(':')
            t3 = t[2].strip('abcdefghijklmnopqrstuvwxy \
                nABCDEFGHIJKLMNOPQRSTUVWXYZ')
            timestamp = t[0] + ':' + t[1] + ':' + t3
        elif 'GRID_CELL_SIZE_REFLECTIVE' in line:
            sizetop = line.split('=')[1]
            size = sizetop.strip('\n')
            G = float(size)

```

```

elif 'UTM_ZONE' in line:
    UTMZONE = str(line.split('=')[1])
    UTMZONE_strip = UTMZONE.strip(' \n')
    #The following line should be modified based on
    #where .prj files are stored relative to data.
    Ex:
    prjfile = basepath + '\\PRJ_Files\\WGS 1984 UTM
    Zone ' + UTMZONE_strip + 'N.prj'
elif 'WRS_PATH' in line:
    PATH = int(line.split('=')[1])
elif 'WRS_ROW' in line:
    ROW = int(line.split('=')[1])
else:
    continue
f.close()
print 'Metadata parsed...'

year_day = year + '_' + month + '_' + day #Construct
timestamp for naming conventions
raster = Path_Raw + filename + '\\ ' + filename + '_B4.
tif' #Use a single band for creation clipping mask
for other datasets. This can be any image.
flag = 'N'
clipshp, arcrast = RastToShp(raster, year_day, flag) #
Return clipping polygon and raster that is used to
snap other datasets to same grid
print 'Extent Created...'
HS_Mask, prjDEM = HSMask(year_day, clipshp, arcrast,
Sun_azimuth, Sun_elevation, Asia_SRTM, prjfile) #Create a
hillshade image to remove shadows from forthcoming
datasets
print 'HS Created'

if Asia_Slope == 'NULL':
    Slope = Path_SRTM + year_day + "_Slope.tif"
    S = arcpy.sa.Slope(prjDEM)
    S.save(Slope)
else:
    Slope == Asia_Slope
print 'Slope done...'
#Choose which velocity profile to use
Velocity = Velocitydir + str(PATH) + '_' + str(ROW) + '
_Normed.tif' #Velocities are named relative to the
Path/Row combination
print 'Velocity chosen...'

```

```

return year_day, HS_Mask, clipshp, timestamp, G, Sensor,
        arcrast, Velocity, Slope, prjfile, prjDEM, PATH, ROW
#Returns data needed for the rest of the script

def RastToShp(rastinfile, year_day, flag):
    """Creates the output extent to be used in the rest of
        the script. Allows all images to be snapped to
        a constant grid, for direct matrix to matrix comparison"""
    ""
    if flag == 'N':
        arcrast = TMP + year_day + "_recast.tif"
        tmptif = arcpy.Raster(rastinfile)
        tmptif.save(arcrast) #This step can be necessary for
            Arc tools to work properly on .tif files
        arcpy.CalculateStatistics_management(arcrast)
    if flag == 'Y':
        arcrast = TMP + year_day + "_recast.tif"
        arcpy.CopyRaster_management(rastinfile, arcrast, '',
            '-9999', '-9999')
    arcpy.env.extent = arcrast
    clipshp_tmp = TMP + year_day + "_clipshp_tmp.shp"
    EXT_data = arcpy.sa.Con(arcrast, 1, 0, "Value" > 0) #
        Choose only areas with data
    tmp_ext = TMP + year_day + "_clipshp_con.tif"
    EXT_data.save(tmp_ext)
    del EXT_data
    arcpy.RasterToPolygon_conversion(tmp_ext, clipshp_tmp, "
        NO_SIMPLIFY", "Value") #Convert to Shpfile
    clipshp = TMP + year_day + "_ext.shp"
    clipshp_tmp2 = TMP + year_day + "_ext_tmp2.shp"
    arcpy.Select_analysis(clipshp_tmp, clipshp_tmp2, "
        GRIDCODE" = 1')
    arcpy.Buffer_analysis(clipshp_tmp2, clipshp, "-2
        Kilometers") #Negative buffer to remove edge effects

    cliprast = TMP + year_day + '_ext.tif'
    arcpy.FeatureToRaster_conversion(clipshp, "GRIDCODE",
        cliprast, 30) #Convert back to TIF file to use as
        output extent

    arcpy.Delete_management(clipshp)
    arcpy.Delete_management(clipshp_tmp)
    arcpy.Delete_management(clipshp_tmp2)
    arcpy.Delete_management(tmp_ext)

```

```

    return cliprast, arcrast

def HSMask(year_day, clipshp, rastband, Sun_azi, Sun_elev,
Asia_DEM, prj):
    """Clips a subset of a larger DEM, and then creates both
        a hillshade and a shadow mask. Returns the shadow
        mask,
        as well as a projected version of the DEM"""
    outDEM = Path_SRTM + year_day + "_DEM.tif"
    HSname = Path_SRTM + year_day + "_DEM_HS.tif"

    arcpy.env.snapRaster = Asia_SRTM
    arcpy.env.extent = clipshp

    SRTM_DEM_clip = arcpy.sa.ExtractByMask(Asia_DEM, clipshp
    )
    SRTM_DEM_clip.save(outDEM)

    prjDEM = Path_SRTM + year_day + "_DEM_prj.tif"
    arcpy.ProjectRaster_management(outDEM, prjDEM, prj, '
        BILINEAR', '30')
    arcpy.Delete_management(outDEM)

    HS = arcpy.sa.Hillshade(prjDEM, Sun_azi, Sun_elev, "
        SHADOWS") #Create Hillshade image
    HS.save(HSname)
    del HS

    #Find shadowed areas and return a binary raster to use
    as a mask
    HS_Maskout = Path_SRTM + year_day + "_HSMask.tif"
    HS_Mask = arcpy.sa.Con(HSname, 1, 0, '"VALUE" < 10')
    HS_Mask.save(HS_Maskout)
    del HS_Mask
    arcpy.Delete_management(HSname)

    return HS_Maskout, prjDEM

def ExtMatch(year_day, band, clipshp, tiffilename, arcrast,
prjfile, foldername):
    """This function matches all rasters to the same grid.
        It resamples when necessary using bilinear resampling
        .
        Returns the clipped version of the input dataset."""

```

```

arcpy.env.snapRaster = arcrast
arcpy.env.extent = clipshp

outprj = TMP + year_day + '_' + band + '_tightprj.tif'
arcpy.ProjectRaster_management(tiffile, outprj, prjfile,
    'BILINEAR', '30')

out_tight = TMP + year_day + "_" + band + '_' +
    foldername + "_tight.tif"
cliptif = arcpy.sa.ExtractByMask(outprj, clipshp)
cliptif.save(out_tight)
del cliptif
arcpy.Delete_management(outprj)

cell = arcpy.GetRasterProperties_management(out_tight, '
    CELLSIZE')
cellsize = cell.getOutput(0)
if cellsize <> 30:
    out_tight_res = TMP + year_day + "_" + band + "
        _tight_res.tif"
    arcpy.Resample_management(out_tight, out_tight_res,
        '30', 'BILINEAR') #Resample all datasets to same
        resolution
    arcpy.Delete_management(out_tight)
    arcpy.CopyRaster_management(out_tight_res, out_tight
        )
    arcpy.Delete_management(out_tight_res)

return out_tight

def ConvertShp(dataset, year_day, timestamp, filename, typed
, G, prjfile):
    """This function takes a raster dataset and converts it
    to polygons. It also adds a set
    of metadata, including year, timestamp, image ID, and
    centroid to each polygon."""

arcpy.env.extent = Asia_SRTM

if typed == 'G':
    outfeats = Path_Glac_Ext + year_day + '_' + filename
        + '_glaciers_tmp.shp'
    R35 = Glac_Base + '_35.tif'
if typed == 'L':
    outfeats = Path_Lake_Ext + year_day + '_' + filename

```



```

        + '_lakes_tmp.shp'

arcpy.CalculateStatistics_management(dataset)
outfeats_t = basepath + '\\TMP\\tmp_shp.shp'
arcpy.RasterToPolygon_conversion(dataset, outfeats_t, '
    NO_SIMPLIFY') #Convert raster back to polygon
arcpy.Select_analysis(outfeats_t, outfeats, '"GRIDCODE"
    = 1')
arcpy.Delete_management(outfeats_t)
arcpy.DefineProjection_management(outfeats, prjfile)

arcpy.AddField_management(outfeats, 'IMG', 'TEXT') #Add
    Fields
arcpy.AddField_management(outfeats, 'Date', 'TEXT')
arcpy.AddField_management(outfeats, 'Timestamp', 'TEXT')
arcpy.AddField_management(outfeats, 'Area', 'FLOAT')
arcpy.AddField_management(outfeats, 'Perimeter', 'FLOAT'
    )
arcpy.AddField_management(outfeats, 'Error', 'FLOAT')
arcpy.AddField_management(outfeats, 'CentX', 'FLOAT')
arcpy.AddField_management(outfeats, 'CentY', 'FLOAT')

opt1 = "'%s'" % filename
arcpy.CalculateField_management(outfeats, 'IMG', opt1, '
    PYTHON') #Add filename
operator = "'%s'" % year_day
arcpy.CalculateField_management(outfeats, 'Date',
    operator, 'PYTHON') #Add Date
opt2 = "'%s'" % timestamp
arcpy.CalculateField_management(outfeats, 'Timestamp',
    opt2, 'PYTHON') #Add Timestamp
arcpy.CalculateField_management(outfeats, 'Area', "float
    (!SHAPE.AREA!)", 'PYTHON') #Calculate Area
arcpy.CalculateField_management(outfeats, 'CentX', "
    float(!SHAPE.CENTROID!.split()[0])", 'PYTHON') #Add
    centroids
arcpy.CalculateField_management(outfeats, 'CentY', "
    float(!SHAPE.CENTROID!.split()[1])", 'PYTHON')

print 'Fields calculated for ' + year_day + ' on ' +
    outfeats

cur = arcpy.UpdateCursor(outfeats)
shapeName = arcpy.Describe(outfeats).shapeFieldName
for shape in cur:

```

```

    feat = shape.getValue(shapeName)
    shape.PERIMETER = feat.length #Add perimeter
    shape.ERROR = feat.length/(G * 0.6872 * (math.pow(G
        ,2))/2) # (P/G) * 0.6872 * G2/2 where P is
        perimeter and G is spatial resolution pre
        resampling
    ##!Perimeter! /( 30 * 0.6872 * (math.pow(30,2))/2)##
    cur.updateRow(shape)
del cur
if typed == 'G': #For glaciers - trim out small spaces
    if they are not explicitly classified as glacier
        outtable = TMP + 'R35table.dbf'
        arcpy.CalculateStatistics_management(R35)
        outfeats_fin = Path_Glac_Ext + year_day + '_' +
            filename + '_glaciers.shp'
        arcpy.Select_analysis(outfeats, outfeats_fin, '"AREA
            " > 1000 AND "PERIMETER" > 1000')
        out_spec = basepath + '\\Spec_Raw\\' + year_day + '_'
            + filename + '_glaciers_35_tmp.shp'
        out_spec_fin = basepath + '\\Spec_Raw\\' + year_day
            + '_' + filename + '_glaciers_35.shp'
        arcpy.RasterToPolygon_conversion(R35, out_spec, '
            NO_SIMPLIFY') #Convert raster back to polygon
        arcpy.DefineProjection_management(out_spec, prjfile)
        arcpy.Select_analysis(out_spec, out_spec_fin, '"
            GRIDCODE" = 1')
        arcpy.Delete_management(out_spec)
        arcpy.Delete_management(R35)
        arcpy.Delete_management(outfeats)
        print 'Zonal Stats Done for Glacier...'

if typed == 'L': #For Lakes - trim out small areas
    outfeats_fin = Path_Lake_Ext + year_day + '_' +
        filename + '_lakes.shp'
    arcpy.Select_analysis(outfeats, outfeats_fin, '"AREA
        " > 5') #500
    arcpy.Delete_management(outfeats)

return outfeats_fin

## Set up folders for things to be saved to and pulled from

Path_Raw = basepath + #PATH TO RAW DATA (Landsat images each
    in seperate directory)
Path_Glac_Ext = basepath + #PATH TO OUTPUT GLACIAL EXTENT

```

```

SHAPEFILES TO
Path_Lake_Ext = basepath + #PATH TO OUTPUT LAKE EXTENTS TO
Path_SRTM = basepath + #PATH TO SAVE RESAMPLED DEMS INTO
TMP = basepath + #WORKING DIRECTORY WHERE TEMPORARY DATASETS
WILL BE SAVED

for foldername in os.listdir(Path_Raw):
    try:
        print 'Processing ' + foldername
        arcpy.env.extent = Asia_SRTM #Set the default
            processing extent before each tree
        year_day, HS_Mask, clipshp, timestamp, G, Sensor,
            arcrast, Velocity, Slope, prjfile, prjDEM, PATH,
            ROW = LandsatMetadataRead(foldername) #Read
            metadata, prep elevation derived datasets
        arcpy.env.extent = clipshp

        for file in os.listdir(Path_Raw + foldername):
            Landsat = Path_Raw + foldername + "\\\" + file
            ## Clip all the images to the same size
            if Landsat.endswith('.TIF') == True:
                bandtop = Landsat.split('_')[-1]
                band = bandtop.strip('.tifTIF')
                if band == '1':
                    band = 'B6_1'
                elif band == '2':
                    band = 'B6_2'
                if band == 'B1' or band == 'B2' or band == '
                    B3' or band == 'B4' or band == 'B5' or
                    band == 'B7' or band == 'B6':
                    if os.path.exists(TMP + year_day + "_" +
                        band + '_' + foldername + "_tight.
                        tif") == False:
                        ExtMatch(year_day, band, clipshp,
                            Landsat, arcrast, prjfile,
                            foldername) #This function does
                            the clipping and resampling
                        print band + ' extent matched.'
                    else:
                        continue
                else:
                    continue

            if Sensor == 'LE7' or Sensor == 'LT5':
                B1 = TMP + year_day + '_B1_' + foldername + "

```

```

        _tight.tif"
    B3 = TMP + year_day + '_B3_' + foldername + "
        _tight.tif"
    B4 = TMP + year_day + '_B4_' + foldername + "
        _tight.tif"
    B5 = TMP + year_day + '_B5_' + foldername + "
        _tight.tif"
    B7 = TMP + year_day + '_B7_' + foldername + "
        _tight.tif"
elif Sensor == 'LC8': #Rename some files to account
    for different band values on LC8
    B1 = TMP + year_day + '_B2_' + foldername + "
        _tight.tif"
    B3 = TMP + year_day + '_B4_' + foldername + "
        _tight.tif"
    B4 = TMP + year_day + '_B5_' + foldername + "
        _tight.tif"
    B5 = TMP + year_day + '_B6_' + foldername + "
        _tight.tif"
    B7 = TMP + year_day + '_B7_' + foldername + "
        _tight.tif"

band = 'ShadowMask'
if os.path.exists(TMP + year_day + "_" + band + '_'
    + foldername + "_tight.tif") == False:
    Mask = ExtMatch(year_day, band, clipshp, HS_Mask
        , arcrast, prjfile, foldername)
else:
    Mask = TMP + year_day + "_" + band + '_' +
        foldername + "_tight.tif"
print 'Shadow Mask Created'

band = 'SlopeMask'
if os.path.exists(TMP + year_day + "_" + band + '_'
    + foldername + "_tight.tif") == False:
    SlopeMask = ExtMatch(year_day, band, clipshp,
        Slope, arcrast, prjfile, foldername)
else:
    SlopeMask = TMP + year_day + "_" + band + '_' +
        foldername + "_tight.tif"
print 'Slope Mask Created'

band = 'ElevMask'
if os.path.exists(TMP + year_day + "_" + band + '_'
    + foldername + "_tight.tif") == False:

```

```

        ElevMask = ExtMatch(year_day, band, clipshp,
                            prjDEM, arcrast, prjfile, foldername)
    else:
        ElevMask = TMP + year_day + "_" + band + '_' +
                    foldername + "_tight.tif"
    print 'Elevation Mask Created'

    band = 'RivMask'
    if os.path.exists(TMP + year_day + "_" + band + '_' +
                      + foldername + "_tight.tif") == False:
        RivMask = ExtMatch(year_day, band, clipshp,
                            Rivers, arcrast, prjfile, foldername)
    else:
        RivMask = TMP + year_day + "_" + band + '_' +
                    foldername + "_tight.tif"
    print 'River Mask Created'

    band = 'VelMask'
    if os.path.exists(TMP + year_day + "_" + band + '_' +
                      + foldername + "_tight.tif") == False:
        Vel_Arc = TMP + year_day + '_tmpVel.tif' #This
            is necessary for datasets which come from
            Matlab, because Arc is bad with projections
        v = arcpy.Raster(Velocity)
        v.save(Vel_Arc)
        del v
        VelMask = ExtMatch(year_day, band, clipshp,
                            Vel_Arc, arcrast, prjfile, foldername)
        arcpy.Delete_management(Vel_Arc)
    else:
        VelMask = TMP + year_day + "_" + band + '_' +
                    foldername + "_tight.tif"
    print 'Velocity Mask Created'

    band = 'LakeTrace'
    if os.path.exists(TMP + year_day + "_" + band + '_' +
                      + foldername + "_tight.tif") == False:
        Lakes_known = ExtMatch(year_day, band, clipshp,
                                TraceLakes, arcrast, prjfile, foldername)
    else:
        Lakes_known = TMP + year_day + "_" + band + '_' +
                        + foldername + "_tight.tif"
    print 'Lake Training Set Created'

    band = 'ManualDebris'

```

```

if os.path.exists(TMP + year_day + "_" + band + '_' +
    + foldername + "_tight.tif") == False:
    Debris = ExtMatch(year_day, band, clipshp,
        TraceDebris, arcrast, prjfile, foldername)
else:
    Debris = TMP + year_day + "_" + band + '_' +
        foldername + "_tight.tif"
print 'Manual Debris Cover Clipped'

arcpy.Delete_management(arcrast)
if Asia_Slope == 'NULL':
    arcpy.Delete_management(Slope)
print 'Input Datasets Created'

#Set up output paths
Out_Lakes = Path_Lake_Ext + year_day + '_' +
    foldername + '_lakes.tif'
Glac_Base = Path_Glac_Ext + year_day + '_' +
    foldername + '_glaciers'
Out_Glaciers = Path_Glac_Ext + year_day + '_' +
    foldername + '_glaciers.tif'

## Send the inputs to Matlab for processing of Lakes
##
try:
    arcpy.Delete_management(Out_Lakes)
except:
    pass
if os.path.exists(Out_Lakes) == False:
    #print "matlab -nosplash -nodesktop -r
    LakeExtract('%s', '%s', '%s', '%s', '%s', '%s')" %
        (B1, B4, HS_Mask, Out_Lakes, Lakes_known,
        SlopeMask)
    subprocess.call("matlab -nosplash -nodesktop -r
        LakeExtract('%s', '%s', '%s', '%s', '%s', '%s')" %
            (B1, B4, Mask, Out_Lakes, Lakes_known,
            SlopeMask))
    while not os.path.exists(Out_Lakes):
        time.sleep(1)
time.sleep(5)
typed = 'L'
arcpy.DefineProjection_management(Out_Lakes, prjfile
    )
Lake_Shp = ConvertShp(Out_Lakes, year_day, timestamp
    , foldername, typed, G, prjfile)

```

```

## Send inputs to Matlab for processing of Glaciers
##
try:
    arcpy.Delete_management(Out_Glaciers)
except:
    pass
if os.path.exists(Out_Glaciers) == False:
    #Matlab takes a hard number of characters for
    #command line calls, so this writes all of the
    #data to a CSV which is then parsed in Matlab
    procsv = 'E:\\Kyrgyz_Glacier_Cover\\
    Extensive_Landsat\\Procfiles\\Threshold_' +
    str(PATH) + '_' + str(ROW) + '.csv'
    tmproc = 'E:\\Kyrgyz_Glacier_Cover\\
    Extensive_Landsat\\Procfiles\\procfile_' +
    foldername + '.csv'
    csvwrite = csv.writer(open(tmproc, 'wb'))
    csvwrite.writerow([B1,B3,B4,B5,B7,Mask,Glac_Base
    ,Out_Lakes,SlopeMask,ElevMask,RivMask,Debris,
    VelMask,procsv])
    del csvwrite
    #print "matlab -nosplash -nodesktop -r
    #GlacExtract('%s')" % tmproc
    subprocess.call("matlab -nosplash -nodesktop -r
    GlacExtract('%s')" % tmproc)
    d = 0
    while not os.path.exists(Out_Glaciers):
        time.sleep(1)
        d = d + 1
        if d > 1000:
            break

print 'Matlab done. Vectorizing...'
time.sleep(5) #Allow five seconds for datasets to be
properly saved out

## Do final attribute adding and conversion
typed = 'G'
GlacOut = Glac_Base + '.tif'
Glac_Shp = ConvertShp(GlacOut, year_day, timestamp,
foldername, typed, G, prjfile)
arcpy.DefineProjection_management(Glac_Shp, prjfile)
## Clean up unnecessary datasets ##
arcpy.Delete_management(B1)

```

```
arcpy.Delete_management(B3)
arcpy.Delete_management(B4)
arcpy.Delete_management(B5)
arcpy.Delete_management(Mask)
arcpy.Delete_management(B7)
arcpy.Delete_management(SlopeMask)
arcpy.Delete_management(ElevMask)
arcpy.Delete_management(RivMask)
arcpy.Delete_management(Out_Lakes)
arcpy.Delete_management(Debris)
arcpy.Delete_management(Out_Glaciers)
arcpy.Delete_management(VelMask)
print 'Done with ' + foldername
```

```
except:
    print foldername + ' failed.'
    traceback.print_exc()
    print 'I hope that error code helps.'
    try:
        arcpy.Delete_management(B1)
        arcpy.Delete_management(B3)
        arcpy.Delete_management(B4)
        arcpy.Delete_management(B5)
        arcpy.Delete_management(Mask)
        arcpy.Delete_management(B7)
        arcpy.Delete_management(SlopeMask)
        arcpy.Delete_management(ElevMask)
        arcpy.Delete_management(RivMask)
        arcpy.Delete_management(Out_Lakes)
        arcpy.Delete_management(Debris)
        arcpy.Delete_management(Out_Glaciers)
        arcpy.Delete_management(VelMask)
    except:
        pass
```


Appendix B. Matlab Code - Lake Delineation

```
function LakeExtract(TM1_r, TM4_r, Mask_r, outpath,
    tracerLakes, Slope_r)
%LakeExtract - Lake extraction via Landsat Band Ratios
% LakeExtract(TM1, TM4, Mask, outpath, TracerLakes, Slope)
% returns a TIF file delineating lake areas in a given
    scene.
% Functions by checking band ratios in a set of training
    lakes
% (TracerLakes) and using them to delineate other lakes.

%Created by Taylor Smith, June 2014

[TM1, refmat, bbox] = geotiffread(TM1_r); %Read in datasets
TMinfo = geotiffinfo(TM1_r);
TM1 = single(TM1); %Integerize
idx0 = find(TM1 <= 0); TM1(idx0) = NaN; %Recast nodata as
    NaN
disp('TM1 Loaded')

[TM4, refmat, bbox] = geotiffread(TM4_r);
TM4 = single(TM4); %Integerize
idx0 = find(TM4 <= 0); TM4(idx0) = NaN; %Recast nodata as
    NaN
disp('TM4 Loaded')

[Mask, refmat, bbox] = geotiffread(Mask_r);
Mask = single(Mask); %Integerize
idx0 = find(Mask <= 0); Mask(idx0) = NaN; %Recast nodata as
    NaN
disp('Mask Loaded')

Ratio = (TM4 - TM1)./(TM4 + TM1); %NDWI to identify lakes
clear TM4 bbox idx0
disp('Bands Ratiod')

[LakeT, refmat, bbox] = geotiffread(tracerLakes); %Read in
    manually classified lakes as training
LakeT = single(LakeT);
target = find(LakeT == 1);
LTemp = Ratio;
LRatio = nanmean(LTemp(target)) + 0.05; %Add buffer for
    sediment-laden lakes. Can be removed if lakes are highly
```

```

    constrained
clear LakeT bbox refmat idx0 LTemp target

ratioidx = find(Ratio < LRatio); % Identify lakes based on
    NDWI
Maskidx = find(Mask > 0);

TM1(ratioidx) = 1; %Remove misclassified areas in shadow
TM1(Maskidx) = NaN;
clear ratioidx Maskidx Mask Ratio
restidx = find(TM1 > 1);
TM1(restidx) = NaN;
clear restidx

[Slope, refmat, bbox] = geotiffread(Slope_r);
Slope = single(Slope); %Integerize
idx0 = find(Slope <= 0); Slope(idx0) = NaN;
disp('Slope Loaded')
clear idx0

Slopeidx = find(Slope > 5); %Identify slopes greater than 5
    degrees and remove lakes
TM1(Slopeidx) = NaN;
clear Slope Slopeidx

B3 = int16(TM1);
B4 = reshape(B3, TMinfo.Height, TMinfo.Width); %Reshape to
    save out
clear B3

geotiffwrite(outpath, B4, TMinfo.SpatialRef, '
    CoordRefSysCode', TMinfo.GeoTIFFCodes.PCS);
disp(strcat(outpath, ' created. '))
exit

```

Appendix C. Matlab Code - Glacier Delineation

```
function GlacExtract(csvfile)
%GlacExtract(csvfile)
%   This function performs glacier extraction via Landsat
%   Band comparisons.
%   It requires an extensive set of input datasets, read
%   from a CSV. These
%   are TM1, TM3, TM4, TM5, TM7, Shadow Mask, Base Export
%   Path, Lakes from
%   Lake_Extraction, Slope, Elevation, River Network, Manual
%   Debris, Velocity,
%   and a CSV containing thresholds for the region.

%Created by Taylor Smith, June 2014

%% Data read-in
fid = fopen(csvfile, 'r'); %Read attributes from a csv
nextLine = fgetl(fid);
Cells = strsplit(nextLine, ',');
TM1_r = Cells{1};
TM3_r = Cells{2};
TM4_r = Cells{3};
TM5_r = Cells{4};
TM7_r = Cells{5};
Mask_r = Cells{6};
Glac_Base = Cells{7};
Out_Lakes = Cells{8};
Slope_r = Cells{9};
Elev_r = Cells{10};
Riv_r = Cells{11};
TraceDebris_r = Cells{12};
Vel_r = Cells{13};
Attscsv = Cells{14};
fclose(fid);

fid = fopen(Attscsv, 'r'); %Pull elevation, min and max
    velocity, and NDVI threshold for each individual Landsat
    scene
nextLine = fgetl(fid);
Cells = strsplit(nextLine, ',');
ElevThreshold = str2double(Cells{1});
MinVelocity = str2double(Cells{2});
MaxVelocity = str2double(Cells{3});
```

```

NDVIThreshold = str2double(Cells{4});
fclose(fid);

%% Pure Glacial Delineation
[TM3, refmat, bbox] = geotiffread(TM3_r);
TM3 = single(TM3); %Integerize
idx0 = find(TM3 <= 0); TM3(idx0) = NaN; %Recast nodata as
    NaN

[TM5, refmat, bbox] = geotiffread(TM5_r);
TM5 = single(TM5); %Integerize
idx0 = find(TM5 <= 0); TM5(idx0) = NaN; %Recast nodata as
    NaN

Ratio = TM3./TM5; %Create the ratio of TM3/TM5
disp('Bands Ratioid')
clear TM5 bbox idx0 TM3
TMinfo = geotiffinfo(TM1_r);

[TM1, refmat, bbox] = geotiffread(TM1_r);
TMinfo = geotiffinfo(TM1_r);
TM1 = single(TM1); %Integerize
idx01 = find(TM1 <= 0); TM1(idx01) = NaN; %Recast nodata as
    NaN

ratioidx = find(Ratio >= 2 & TM1 > 250); %Use both the ratio
    and TM band 1 to pull out glaciated areas.
% This calculation is very bad at pulling thick debris cover
    , so the rest of this script works to identify debris
    cover
TM1(ratioidx) = 1;
clear ratioidx bbox idx01 Ratio

[Mask, refmat, bbox] = geotiffread(Mask_r); %Mask out
    shadows
Mask = single(Mask); %Integerize
idx0 = find(Mask <= 0); Mask(idx0) = NaN; %Recast nodata as
    NaN
disp('Mask Loaded')

Maskidx = find(Mask > 0);
TM1(Maskidx) = NaN;
clear Maskidx Mask bbox idx0

[Lake, refmat, bbox] = geotiffread(Out_Lakes); %Mask out

```

```

    lakes
Lake = single(Lake); %Integerize
idx0 = find(Lake <= 0); Lake(idx0) = NaN; %Recast nodata as
    NaN
clear bbox idx0

Lakeidx = find(Lake == 1); %Create an index of just lakes
TM1(Lakeidx) = 1; %Temporarily add lakes to glacier dataset
    for processing
clear Lake

restidx2 = find(TM1 > 1);
TM1(restidx2) = NaN;
restidx = find(TM1 > 0);

Ttmp = TM1;
Ttmp(Lakeidx) = NaN;

outpath = strcat(Glac_Base, '_35.tif');
geotiffwrite(outpath, int16(Ttmp), TMinfo.SpatialRef, '
    CoordRefSysCode', TMinfo.GeoTIFFCodes.PCS); %Write out
    the spectral-only outlines for future processing
clear Ttmp

[Seeds, refmat, bbox] = geotiffread(TraceDebris_r); %Add
    manual seed points to index of glaciers
Seeds = single(Seeds);
Sidx = find(Seeds == 1); clear Seeds
TM1(Sidx) = 1; clear Sidx
testseed = find(TM1 > 0);
clear TM1 restidx2

%% Examine Debris Areas
% At this point, these are pure glacier outlines. Next find
    debris

[Slope, refmat, bbox] = geotiffread(Slope_r);
Slope = single(Slope); %Integerize
idx0 = find(Slope <= 0); Slope(idx0) = NaN;
disp('Slope Loaded')
clear idx0

Slopeidx = find(Slope > 24); %Identify slopes between 1 and
    24 degris (After Paul et al., 2004)
Slope(Slopeidx) = NaN;

```

```

Slope2idx = find(Slope < 1);
Slope(Slope2idx) = NaN;
S = stdfilt(Slope, ones(9)); %Identify and remove areas of
    low slope variability (plains, riverbeds, etc)
Stdidx = find(S > 2);
Slope(Stdidx) = NaN;
clear S Stdidx
Slopeidxr = find(Slope > 0);
Slope(Slopeidxr) = 1;
clear Slope2idx Slopeidx Slopeidxr

[Elev, refmat, bbox] = geotiffread(Elev_r); %Filter for
    elevation. This can be set dependent on lowest glacier
    elevation in study region
Elev = single(Elev); %Integerize
idx0 = find(Elev <= 0); Elev(idx0) = NaN;
disp('Elev Loaded')
clear idx0
elevmask = find(Elev < ElevThreshold); %Low elev, not
    glacier
Slope(elevmask) = NaN;
clear Elev refmat bbox

[Riv, refmat, bbox] = geotiffread(Riv_r); %Read in rivers to
    remove mis-identified areas of wet soil
Riv = single(Riv); %Integerize
idx0 = find(Riv > 1); Riv(idx0) = NaN;
Riv(elevmask) = NaN;
rivseed = find(Riv == 1);
clear Riv refmat bbox

[Vel, refmat, bbox] = geotiffread(Vel_r); %Read velocity
    data
Vel = single(Vel);
idx0 = find(Vel < 0); Vel(idx0) = NaN;
idx1 = find(Vel > MaxVelocity); Vel(idx1) = NaN;
disp('Velocity Loaded')
velmask = find(Vel < MinVelocity);
Slope(velmask) = NaN;
clear Vel refmat bbox idx0 idx1

%Use a distance weighting metric based on distance from
    rivers (center of valleys) to identify debris areas
T = graydist(Slope, rivseed);
Tidx = find(T > 90);

```

```

Slope(Tidx) = NaN; Slope(elevmask) = NaN;
clear Tidx T elevmask

seed2idx = find(Slope > 0);
Slope(seed2idx) = 1;
Slope(restidx) = NaN;
clear seed2idx rivseed
maskidx = isnan(Slope);

%% Filter Overclassifications
%Debris maximum extent is now identified. The next steps try
    to remove misclassified areas

%% NDWI Masking
[TM1, refmat, bbox] = geotiffread(TM1_r);
TM1 = single(TM1); %Integerize
idx0 = find(TM1 <= 0); TM1(idx0) = NaN; %Recast nodata as
    NaN
disp('TM1 Loaded')

[TM4, refmat, bbox] = geotiffread(TM4_r);
TM4 = single(TM4); %Integerize
idx0 = find(TM4 <= 0); TM4(idx0) = NaN; %Recast nodata as
    NaN
disp('TM4 Loaded')

Ratio = (TM4 - TM1)./(TM4 + TM1); %NDWI Calculation
clear TM4 TM1
Ratio(maskidx) = NaN;
RadIdx = find(Ratio > 0); % Remove positive water areas
Slope(RadIdx) = NaN;
clear RadIdx Ratio

%% NDVI Masking
maskidx = isnan(Slope);
[TM3, refmat, bbox] = geotiffread(TM3_r);
TM3 = single(TM3); %Integerize
idx0 = find(TM3 <= 0); TM3(idx0) = NaN; %Recast nodata as
    NaN

[TM4, refmat, bbox] = geotiffread(TM4_r);
TM4 = single(TM4); %Integerize
idx0 = find(TM4 <= 0); TM4(idx0) = NaN; %Recast nodata as
    NaN

```

```

NDVI = (TM4 - TM3)./(TM4 + TM3); clear TM4 TM3 idx0 bbox
      refmat
NDVI(maskidx) = NaN;
NDVIdx = find(NDVI > NDVIThreshold); %Remove vegetated areas
Slope(NDVIdx) = NaN;
clear NDVI NDVIdx

S2 = Slope; % Do a second distance weighting from only
      manual seed points and lakes
S2(restidx) = 1;
T2 = graydist(S2,testseed);
T2idx = find(T2 > 400);
Slope(T2idx) = NaN;
clear S2 T2 T2idx

%% Statistical Filtering
slopeidx = isnan(Slope);
Slope(slopeidx) = 0; clear slopeidx
B = medfilt2(Slope, [3 3]);
new = ~bwareaopen(~B, 100000);
clear B Slope

new(restidx) = 1;
new(testseed) = 1;
new(Lakeidx) = 0;
clear TM1 restidx Lakeidx testseed

B2 = medfilt2(new, [3 3]); %Single median filter pass
disp('Pass 2 Done. Saving...')
new2 = bwareaopen(B2, 1000); % Was 1500
clear B2
new4 = bwmorph(new2,'open'); %Perform an image opening to
      remove more edge effects
new4 = single(new4);
clear new2

[Slope, refmat, bbox] = geotiffread(Slope_r);
Slope = single(Slope);
idx0 = find(Slope <= 0); Slope(idx0) = NaN;
disp('Slope Loaded')
clear idx0

Slopeidx = find(Slope > 60); %Make sure that no new high-
      slope areas have been added during distance weighting/
      statistical filtering

```



```

new4(Slopeidx) = NaN;
clear Slope Slopeidx
new5 = medfilt2(new4, [5 5]);
clear new4
new5(find(new5 == 0)) = NaN;

BW = im2bw(new5, 0.1);
new6 = bwmorph(BW, 'bridge');
clear BW
clear maskidx new new5 velmask
new8 = +new6;

outpath = strcat(Glac_Base, '.tif');
geotiffwrite(outpath, int16(new8), TMinfo.SpatialRef, '
    CoordRefSysCode', TMinfo.GeoTIFFCodes.PCS); %Save out the
    glaciated areas

exit

```

Appendix D. Python Code - DEM Correction

```
##This script was created to prepare ASTER data for DEM
  Extraction, and then for optimization via Matlab. It
  takes
##inputs as GeoTiff files, and exports corrected images into
  a defined file structure.
##REQUIRES CUSTOM MODULES: Numpy, Solar, and Arcpy, with the
  Spatial Analyst Extension available. Designed under
  Python 2.6.5
##Created by Taylor Smith, Mar 2013
## Most recent update: June 2014

#Import Modules
print "Loading modules..."
import arcpy, sys, os, string, csv, numpy, glob
import datetime, solar, time, subprocess

arcpy.env.overwriteOutput = True #Set the script to be able
  to overwrite files
arcpy.CheckOutExtension("Spatial")

basepath = "" #Base directory goes here
#These lines comprise the folder structure I have developed.
  You can add or change these as necessary for your
  purposes.
Path_Corrected = basepath + #Corrected DEM Output Directory
Path_CorrStat = basepath + #Correction Stats output
Path_Extents = basepath + #Extents
Path_SRTM = basepath + #SRTM Path
Path_ASTER = basepath + #ASTER DEMs Stored here. Script
  loops through here

def conf_reader(config_file): #Read config files with
  comments
  f = open(config_file, 'r')
  conf_list = []
  for line in f:
    raw = line.split('#')
    data = raw[0]
    new = data.replace('\r\n', '')
    new_2 = new.replace(' ', '')
    conf_list.append(new_2)
  f.close()
```

```

    return conf_list

if os.path.exists(basepath + "DEM_Correction.conf"): #Read
in necessary parameters from config file
    f = open(basepath + "DEM_Correction.conf",'r')
    conf = f.read().split('\n')
    SRTM_Cont = conf[0]
    process = conf[2]
    correct = conf[3]
    maskin = conf[4]
    Path_Mask = conf[5]
    buffdist = float(conf[6])
    Resampled_SRTM = conf[7]
    f.close()

def SRTMtoASTSpace(shortname,timestamp,area,code,ASTER):
    """This function clips two DEMs to the same extent, and
    equalizes their cellsize. This
    preps the rasters for use as Matlab matrices"""

    arcpy.env.snapRaster = Resampled_SRTM
    arcpy.env.extent = Resampled_SRTM

    clipshp_tmp = Path_SRTM + shortname + "_" + timestamp +
        "_" + code + "_DEM_clip_tmp.shp"
    clipshp = Path_SRTM + shortname + "_" + timestamp + "_"
        + code + "_DEM_clip.shp"
    ASTER_DEM_data = arcpy.sa.Con(ASTER, 1, 0, ' "VALUE" > 0'
        )
    arcpy.RasterToPolygon_conversion(ASTER_DEM_data,
        clipshp_tmp, "NO_SIMPLIFY", "VALUE")

    if buffdist == 0:
        clipshp_tmp2 = Path_SRTM + shortname + "_" +
            timestamp + "_" + code + "_DEM_clip_tmp2.shp"
        arcpy.Select_analysis(clipshp_tmp, clipshp_tmp2, ' "
            GRIDCODE" = 1')
        clipshp = clipshp_tmp2
    else:
        clipshp_tmp2 = Path_SRTM + shortname + "_" +
            timestamp + "_" + code + "_DEM_clip_tmp2.shp"
        arcpy.Buffer_analysis(clipshp_tmp, clipshp, "%f
            Kilometers" % buffdist) #Remove edge effects
        arcpy.Select_analysis(clipshp, clipshp_tmp2, ' "
            GRIDCODE" = 1')

```

```

        clipshp = clipshp_tmp2

arcpy.env.extent = clipshp
outDEM_tight = Path_SRTM + shortname + "_" + timestamp +
    "_" + code + "_DEM_tight.tif"
SRTM_DEM_clip = arcpy.sa.ExtractByMask(Resampled_SRTM,
    clipshp)
SRTM_DEM_clip.save(outDEM_tight)

outname = Path_Corrected + shortname + "_" + timestamp +
    "_" + code + "_ArcForm.tif"
AST_DEM_clip = arcpy.sa.ExtractByMask(ASTER, clipshp)
AST_DEM_clip.save(outname)

if MASK != 'NULL':
    MSKout = Path_Corrected + shortname + "_" +
        timestamp + "_" + code + "_MASK_clip.tif"
    MSK_Clip = arcpy.sa.ExtractByMask(MASK, clipshp)
    MSK_Clip.save(MSKout)
    MSK = MSKout

arcpy.Delete_management(clipshp)
try:
    arcpy.Delete_management(chipshp_tmp)
except:
    pass
try:
    arcpy.Delete_management(clipshp_tmp2)
except:
    pass

def ResampleReclip(shortname, timestamp, Area, code, ASTER_esp):
    """This function functions pretty much the same as above
    , except it calls different export paths."""
    arcpy.env.extent = Resampled_SRTM #Reset the environment
    extent
    arcpy.env.snapRaster = Resampled_SRTM

    clipshp = Path_SRTM + shortname + "_" + timestamp + "_"
        + code + "_DEM_clip.shp"
    time.sleep(5)
    ASTER_DEM_data = arcpy.sa.Con(ASTER_esp, 1, 0, 'VALUE'
        > 0')
    arcpy.RasterToPolygon_conversion(ASTER_DEM_data, clipshp
        , "NO_SIMPLIFY", "VALUE")

```

```

del ASTER_DEM_data

arcpy.env.extent = clipshp

outDEM_tight = Path_SRTM + shortname + "_" + timestamp +
    "_" + code + "_ASP_SRTM_tight.tif"
SRTM_DEM_clip = arcpy.sa.ExtractByMask(Resampled_SRTM,
    clipshp)
SRTM_DEM_clip.save(outDEM_tight)

outname = Path_Corrected + shortname + "_" + timestamp +
    "_" + code + "_AspCorr.tif"
AST_DEM_clip = arcpy.sa.ExtractByMask(ASTER_asp, clipshp
)
AST_DEM_clip.save(outname)

if MASK != 'NULL':
    MSKout = Path_Corrected + shortname + "_" +
        timestamp + "_" + code + "_ASP_MASK_clip.tif"
    MSK_Clip = arcpy.sa.ExtractByMask(MASK, clipshp)
    MSK_Clip.save(MSKout)

arcpy.Delete_management(clipshp)

def MaskMaker(shortname,timestamp,code,MASK):
    """This function rasterizes a shapefile mask if a mask
        is being used during statistical analysis"""
    MSK = Path_Corrected + shortname + "_" + timestamp + "_"
        + code + "_MASK.tif"
    try:
        arcpy.DeleteField_management(MASK, "DATA")
    except:
        pass
    arcpy.AddField_management(MASK, "DATA", "SHORT")
    arcpy.CalculateField_management(MASK, "DATA", '!DATA! ==
        0', "PYTHON")
    arcpy.PolygonToRaster_conversion(MASK, "DATA", MSK, "
        CELL_CENTER", "#", 15)
    return MSK

def RegReclip(Reg_AST):
    """Reclips SRTM after coregistration"""
    arcpy.env.extent = Reg_AST
    clipshp = Path_SRTM + shortname + "_" + timestamp + "_"
        + code + "_clip_reg.shp"

```

```

time.sleep(5)
ASTER_DEM_data = arcpy.sa.Con(Reg_AST, 1, 0, 'VALUE' >
    0')
arcpy.RasterToPolygon_conversion(ASTER_DEM_data, clipshp
    , "NO_SIMPLIFY", "VALUE")
del ASTER_DEM_data
arcpy.env.extent = clipshp

SRTM_tight = Path_SRTM + shortname + "_" + timestamp + "
    _" + code + "_SRTM_tight_reg.tif"
SRTM_DEM_clip = arcpy.sa.ExtractByMask(Resampled_SRTM,
    clipshp)
SRTM_DEM_clip.save(SRTM_tight)
return SRTM_tight

def DataClean(Elev_Corr):
    """Adds stats to final exported product so it is easier
        to use in ArcGIS. Also
        creates a final difference map between the two DEMs"""
    outSetNull = arcpy.sa.SetNull(Elev_Corr, Elev_Corr, '
        VALUE' < 0')
    outSetNull.save(Elev_Corr)
    arcpy.CalculateStatistics_management(Elev_Corr)

    elev_outline = Path_Corrected + shortname + "_" +
        timestamp + "_" + code + "_finaloutline.shp"
    elev_data = arcpy.sa.Con(Elev_Corr, 1, 0, 'VALUE' > 0')
    arcpy.RasterToPolygon_conversion(elev_data, elev_outline
        , "NO_SIMPLIFY", "VALUE")
    del elev_data
    arcpy.env.extent = elev_outline

    final_DEM_tight = Path_SRTM + shortname + "_" +
        timestamp + "_" + code + "_elev_SRTM_tight.tif"
    SRTM_DEM_clip = arcpy.sa.ExtractByMask(Resampled_SRTM,
        elev_outline)
    SRTM_DEM_clip.save(final_DEM_tight)

    finaldifmap = Path_Corrected + shortname + "_" +
        timestamp + "_" + code + "_finaldifmap.tif"
    minus = arcpy.sa.Minus(final_DEM_tight, Elev_Corr)
    minus.save(finaldifmap)

for root, dirs, files in os.walk(Path_ASTER): #The script
    will walk the entire directory under Path_ASTER looking

```

```

for .tif files. If you don't want them processed, hide
them, or divert the walk function
for name in files:
    ASTER = os.path.join(root,name)
    if ASTER.endswith('.tif') == True:
        longname = name.split('_')
        shortname = longname[0]
        timestamp = longname[1]
        code = longname[2]
        print "Processing: " + ASTER

    if maskin == 'Y':
        if Path_Mask.endswith('.tif'):
            MASK = Path_Mask
        elif Path_Mask.endswith('.shp'):
            MASK = MaskMaker(shortname,timestamp,
                code,Path_Mask) #Rasterize input mask
                data
            print "Mask Generated."
    elif maskin == 'N':
        MASK = 'NULL'

    SRTMtoASTSpace(shortname,timestamp,Area,code,
        ASTER) #Match the two DEMs to same cell
        extents
    ASTER = Path_Corrected + shortname + "_" +
        timestamp + "_" + code + "_ArcForm.tif"
    if MASK == 'NULL':
        MSK = 'NULL'
    else:
        MSK = Path_Corrected + shortname + "_" +
            timestamp + "_" + code + "_MASK_clip.tif"
    SRTM_tight = Path_SRTM + shortname + "_" +
        timestamp + "_" + code + "_DEM_tight.tif"
    print "Data clipped. Starting registration..."

## Image Registration ##
Reg_AST = Path_Corrected + shortname + "_" +
    timestamp + "_" + code + "_Registered.tif"
if os.path.exists(Reg_AST) == False:
    #print "matlab -nosplash -nodesktop -r
        Coregistration('%s','%s','%s')" % (
            SRTM_tight,ASTER,Reg_AST)
    subprocess.call("matlab -nosplash -nodesktop
        -r Coregistration('%s','%s','%s')" % (

```

```

        SRTM_tight, ASTER, Reg_AST)) #Note that
        this is processor intensive, and may
        require a lot of time
    while not os.path.exists(Reg_AST):
        time.sleep(1)
    ASTER = Reg_AST
else:
    ASTER = Reg_AST

SRTM_tight = RegReclip(Reg_AST) #Re-align the
    DEMs
print "Registration done. Tilt correcting..."

## Tilt Correction ##
csvwrite = csv.writer(open(Path_CorrStat +
    shortname + "_" + timestamp + "_" + code + "
    _TiltCorrectionParameters.csv", 'wb'),
    lineterminator='\r\n')
csvwrite.writerow(['fitresult.p00', 'fitresult.
    p10', 'fitresult.p01', 'gof.sse', 'gof.
    rsquare', 'gof.dfe', 'gof.adjrsquare', 'gof.
    rmse'])
del csvwrite

tilt_ASTER = Path_Corrected + shortname + "_" +
    timestamp + "_" + code + "_Tilt_Corrected.tif
"
if os.path.exists(tilt_ASTER) == True:
    arcpy.Delete_management(tilt_ASTER)

subprocess.call("matlab -nosplash -nodesktop -r
    Tilt_Correction('%s', '%s', '%s', '%s', '%s', '%s
    ', '%s', '%s', '%s')" % (Area, shortname,
    timestamp, code, ASTER, SRTM_tight,
    Path_Corrected, Path_CorrStat, MSK)) #Calls for
    area, name and time, code, SRTM, ASTER, and
    savepath inputs
while not os.path.exists(tilt_ASTER):
    time.sleep(1)

print shortname + " tilt corrected."

## Aspect Correction ##
csvwrite = csv.writer(open(Path_CorrStat +
    shortname + "_" + timestamp + "_" + code + "

```



```

    _AspectCorrectionParameters.csv", 'wb'),
    lineterminator='\r\n')
csvwrite.writerow(['dh_tan_fit(1)', 'dh_tan_fit
(2)', 'dh_tan_fit(3)', 'resnorm(1)', 'dh_mean',
'nanmean(tmp_dh)', 'nanstd(tmp_dh)', 'xshift
(m)', 'yshift (m)'])
del csvwrite

#initial conditions
stdevchange = 1
stdinit = 1
AST = tilt_ASTER
loop = 0
while stdevchange > 0.02 and loop < 20: #Loop
    through aspect corrections until the data is
    properly matched. Reclip/resample the DEMs
    each time to keep them matched.
    Aspect_Corr = Path_Corrected + shortname + "
    _" + timestamp + "_" + code + "
    _Aspect_Corrected.tif"
    if os.path.exists(Aspect_Corr) == True:
        arcpy.Delete_management(Aspect_Corr)
    figureout_filename = Path_Corrected +
        shortname + "_" + timestamp + "_" + code
        + "_aspectfit_" + str(loop) + ".jpg"
    subprocess.call("matlab -nosplash -nodesktop
        -r Aspect_Correction('%s', '%s', '%s', '%s
        ', '%s', '%s', '%s', '%s', '%s', '%s')" % (Area
        , shortname, timestamp, code, AST, SRTM_tight,
        Path_Corrected, Path_CorrStat, MSK,
        figureout_filename))
    while not os.path.exists(Aspect_Corr):
        time.sleep(1)
    AST = Aspect_Corr
    ResampleReclip(shortname, timestamp, Area, code
        , AST) #Reclip after each loop

    AST = Path_Corrected + shortname + "_" +
        timestamp + "_" + code + "_AspCorr.tif"
    SRTM_tight = Path_SRTM + shortname + "_" +
        timestamp + "_" + code + "_ASP_SRTM_tight
        .tif"
    if MASK != 'NULL':
        MSK = Path_Corrected + shortname + "_" +
            timestamp + "_" + code + "

```

```

        _ASP_MASK_clip.tif"
else:
    MSK = 'NULL'

f = open(Path_CorrStat + "Aspect_looper.txt"
        , 'r') #This looks for a correction
            parameter
itr = f.read()
out = itr.split(',')
mean = out[0]
std = out[1]
stdevchange = abs((abs(float(std)) - stdinit
                )/stdinit)
stdinit = float(std)
f.close()
stdlooper = stdevchange * 100
print "loop: " + str(loop) + ", dh mean: " +
        str(mean) + ", dh stdev: " + str(std) +
        "m. Change in stdev (%) is " + str(
            stdlooper)
loop = loop + 1

rst = arcpy.Raster(AST)
rst.save(Aspect_Corr)
del rst
print shortname + " aspect corrected after " +
        str(loop)+ " loops. Final std is " + str(
            stdinit)

## Elevation Correction ##
csvwrite = csv.writer(open(Path_CorrStat +
        shortname + "_" + timestamp + "_" + code + "
        _ElevCorrectionParameters.csv", 'wb'),
        lineterminator='\r\n')
csvwrite.writerow(['fitresult.p1', 'fitresult.p2
        ', 'fitresult.p3', 'gof.sse', 'gof.rsquare',
        'gof.dfe', 'gof.adjrsquare', 'gof.rmse'])
del csvwrite

Elev_Corr = Path_Corrected + shortname + "_" +
        timestamp + "_" + code + "
        _Elevation_Corrected.tif"
if os.path.exists(Elev_Corr) == True:
    arcpy.Delete_management(Elev_Corr)

```

```

subprocess.call("matlab -nosplash -nodesktop -r
    Elevation_Correction('%s','%s','%s','%s','%s'
    ,'%s','%s','%s','%s','%s')" % (Area,shortname,
    timestamp,code,Aspect_Corr,SRTM_tight,
    Path_Corrected,Path_CorrStat,MSK))
while not os.path.exists(Elev_Corr):
    time.sleep(1)
print shortname + " elevation corrected."
AST = Elev_Corr

## Compare final image with first image ##
AST_init = Path_Corrected + shortname + "_" +
    timestamp + "_" + code + "_ArcForm.tif"
SRTM_init = Path_SRTM + shortname + "_" +
    timestamp + "_" + code + "_DEM_tight.tif"
Reg_SRTM = Path_SRTM + shortname + "_" +
    timestamp + "_" + code + "_SRTM_tight_reg.tif"
"
figureout = Path_CorrStat + shortname + "_" +
    timestamp + "_" + code + "_Corrections_hist.
    jpg"
figureout2 = Path_CorrStat + shortname + "_" +
    timestamp + "_" + code + "_Corrections_dh.jpg"
"
Reg_AST = Path_Corrected + shortname + "_" +
    timestamp + "_" + code + "_Registered.tif"
tilt_ASTER = Path_Corrected + shortname + "_" +
    timestamp + "_" + code + "_Tilt_Corrected.tif"
"
SRTM_aspect = Path_SRTM + shortname + "_" +
    timestamp + "_" + code + "_ASP_SRTM_tight.tif"
"
Aspect_Corr = Path_Corrected + shortname + "_" +
    timestamp + "_" + code + "_Aspect_Corrected.
    tif"
Elev_Corr = Path_Corrected + shortname + "_" +
    timestamp + "_" + code + "
    _Elevation_Corrected.tif"

figpth = Path_CorrStat + shortname + "_" +
    timestamp + "_" + code
Corrpath = Path_Corrected + shortname + "_" +
    timestamp + "_" + code

if os.path.exists(figureout) == True:

```

```

        os.remove(figureout)
    if os.path.exists(figureout2) == True:
        os.remove(figureout2)

    subprocess.call("matlab -nosplash -nodesktop -r
        compare_DEMs('%s','%s','%s','%s','%s','%s','%s','%s','%s')
        % (AST_init, SRTM_init, MSK,
        Reg_AST, Reg_SRTM, SRTM_aspect, Corrpath,
        figpth))
    while not os.path.exists(figureout2):
        time.sleep(2)

    ## Clean up Data, export bounding box ##
    time.sleep(10)
    DataClean(Elev_Corr)
    print "Supplementary Data Generated..."

    ## Delete Superfluous Datasets ##
    try:
        arcpy.Delete_management(Path_Corrected +
            shortname + "_" + timestamp + "_" + code
            + "_AspCorr.tif")
    except:
        pass
    try:
        arcpy.Delete_management(Path_SRTM +
            shortname + "_" + timestamp + "_" + code
            + "_ASP_SRTM_tight.tif")
    except:
        pass
    try:
        arcpy.Delete_management(Reg_SRTM)
    except:
        pass
    if maskin == 'Y':
        arcpy.Delete_management(Path_Corrected +
            shortname + "_" + timestamp + "_" + code
            + "_MASK_clip.tif")

    print shortname + " finished."
    time.sleep(10)
else:
    continue

```

Appendix E. Matlab Code - DEM Correction: Step 1 - Coregistration

```
function Coregistration(SRTM, ASTER, ASTER_out)
%Coregistration(SRTM, ASTER, ASTER_out)
%   Takes a master and slave DEM for coregistration.

%Created by Taylor Smith, March 2013

%Import SRTM DEM
[SRTM_dem,SRTM_refmat,SRTM_bbox] = geotiffread(SRTM);
SRTM_info = geotiffinfo(SRTM);
SRTM_dem = single(SRTM_dem); %Integerize SRTM DEM
idx0 = find(SRTM_dem < 0); SRTM_dem(idx0) = NaN; %Recast
    nodata as NaN % was ==0 %%%%
idxfinite = find(~isfinite(SRTM_dem)); SRTM_dem(idxfinite) =
    0;

%Import ASTER DEM
[AST_dem,AST_refmat,AST_bbox] = geotiffread(ASTER);
ASTER_info = geotiffinfo(ASTER);
AST_dem = single(AST_dem);
idxnnan = find(AST_dem < 0); AST_dem(idxnnan) = NaN;
idxfinite = find(~isfinite(AST_dem)); AST_dem(idxfinite) =
    0;

%perform intensity-based image registration
[optimizer, metric] = imregconfig('multimodal');
optimizer.MaximumIterations = 5000;
[AST_dem_moved_reg] = imregister(AST_dem, SRTM_dem, 'rigid',
    optimizer, metric);
AST_dem_moved_reg(idxfinite) = NaN;
geotiffwrite(ASTER_out, int16(AST_dem_moved_reg), SRTM_info.
    SpatialRef, 'CoordRefSysCode', ASTER_info.GeoTIFFCodes.
    PCS);
exit
```

Appendix F. Matlab Code - DEM Correction: Step 2 - Tilt Correction

```
function Tilt_Correction(area,date,time,vers,ASTpath,
    SRTMpath,Savepath,Corrstats,MSK)
%Tilt_Correction(area, date, time, version, Slave Path,
    Master Path, save path, correlation stats, Mask)
% Performs a correction to fix tilt in the slave DEM.
% Functions by fitting a plane to common low-slope
% surfaces between the two DEMs.

%Created by Taylor Smith, March 2013

disp(strcat(date,' being processed for Tilt Correction.'))
%Import SRTM DEM
[SRTM_dem,SRTM_refmat,SRTM_bbox] = geotiffread(SRTMpath);
SRTM_info = geotiffinfo(SRTMpath);
SRTM_dem = single(SRTM_dem); %Integerize SRTM DEM
idx0 = find(SRTM_dem <= 0); SRTM_dem(idx0) = NaN; %Recast
    nodata as NaN

%Import ASTER DEM
[AST_dem,AST_refmat,AST_bbox] = geotiffread(ASTpath);
ASTER_info = geotiffinfo(ASTpath);
AST_dem = single(AST_dem);
idxnan = find(AST_dem <= 0); AST_dem(idxnan) = NaN;

%Perform masking if desired
if strcmp(MSK, 'NULL') ~= 1
    [AST_msk,AST_msk_refmat,AST_msk_bbox] = geotiffread(MSK)
    ;
    idxmsk = find(AST_msk >= 1 & AST_msk < 255);
    if length(idxmsk) == numel(AST_msk)
        idxmsk = find(AST_msk == 127);
    end
    AST_dem_msk = AST_dem; AST_dem_msk(idxmsk) = NaN;
    SRTM_dem(idxmsk) = NaN;
    MASK = single(AST_dem_msk);
end
if strcmp(MSK, 'NULL') == 1
    MASK = single(AST_dem);
end

%Apply Tilt Correction
```

```

slope_threshold = 20;
dh_m_threshold = 100;
STD_dif = 4;
warning off

%Create gridbox for use in rest of loop
DEM_X_size = AST_refmat(2,1); DEM_Y_size = AST_refmat(2,1);
[AST_utm_y, AST_utm_x] = meshgrid(AST_bbox(3):DEM_Y_size:
    AST_bbox(4)-DEM_Y_size, AST_bbox(1):DEM_X_size:AST_bbox
    (2)-DEM_X_size);
AST_utm_y = rot90(AST_utm_y); AST_utm_x = rot90(AST_utm_x);

stdinit = 1;
STD_calc = STD_dif;
counter = 1;
max_nr_iterations = 100;
for i = 1:max_nr_iterations
    %calculate SLOPE and ASPECT
    [fx, fy] = gradient(MASK, DEM_X_size, DEM_Y_size);
    AST_SLOPE = sqrt(fx.^2 + fy.^2); clear fx fy
    AST_SLOPE_DEG = atand(AST_SLOPE);

    %Do Plane Fitting
    DH = SRTM_dem - MASK;
    if strcmp(MSK, 'NULL') ~= 1
        DH(idxmsk) = NaN;
    end
    x = reshape(AST_utm_x, 1, numel(AST_utm_x));
    y = reshape(AST_utm_y, 1, numel(AST_utm_y));
    z = reshape(DH, 1, numel(DH));

    %Trim data by slope and error
    idx_slope = find(AST_SLOPE_DEG < slope_threshold); x = x
        (idx_slope); y = y(idx_slope); z = z(idx_slope)';
    idx_dh = find(abs(z) < dh_m_threshold); x = x(idx_dh); y
        = y(idx_dh); z = z(idx_dh);

    [xData, yData, zData] = prepareSurfaceData(x,y,z);
    ft = fittype('poly11'); opts = fitoptions(ft); opts.
        Lower = [-Inf -Inf -Inf]; opts.Robust = 'Bisquare';
        opts.Upper = [Inf Inf Inf];
    [ASTER_DEM_plane_fitresult, ASTER_DEM_plane_gof] = fit(
        [xData, yData], zData, ft, opts );
    ASTER_DEM_plane = ASTER_DEM_plane_fitresult.p00 +
        ASTER_DEM_plane_fitresult.p10 .* AST_utm_x +

```

```

    ASTER_DEM_plane_fitresult.p01 .* AST_utm_y;

Corr_ASTER = single(AST_dem) + ASTER_DEM_plane;

tmp_dh = SRTM_dem - Corr_ASTER;
dh_stdev = nanstd(tmp_dh(:));
AST_dem = Corr_ASTER;

dh_param(i) = nanmean(tmp_dh(:));
if i > 1
    if abs((dh_param(i-1)-dh_param(i))/dh_param(i)) <
        0.005
        %less than 0.5% change, stop
        break
    end
    fprintf(1,'delta h between SRTM and ASTER DEM mean:
        %3.3f m, stdev: %3.3f m, rmse: %3.3f m, percent
        mean elev. change = %1.3f\n', nanmean(tmp_dh(:)),
        dh_stdev, ASTER_DEM_plane_gof.rmse, (dh_param(i
        -1)-dh_param(i))/dh_param(i));
else
    fprintf(1,'delta h between SRTM and ASTER DEM mean:
        %3.3f m, stdev: %3.3f m, rmse: %3.3f m, percent
        mean elev. change = %1.3f\n', nanmean(tmp_dh(:)),
        dh_stdev, ASTER_DEM_plane_gof.rmse, dh_param(i))
    ;
end

%removing areas identified by Mask from ASTER DEM
if strcmp(MSK, 'NULL') ~= 1
    MASK = AST_dem; MASK(idxmsk) = NaN;
end

%Store the fit parameters in a csv
txt = strcat(Corrstats,date,'_',time,'_',vers, '_
    _TiltCorrectionParameters.csv');
towrite = [ASTER_DEM_plane_fitresult.p00
    ASTER_DEM_plane_fitresult.p10
    ASTER_DEM_plane_fitresult.p01 ASTER_DEM_plane_gof.sse
    ASTER_DEM_plane_gof.rsquare ASTER_DEM_plane_gof.dfe
    ASTER_DEM_plane_gof.adjrsquare ASTER_DEM_plane_gof.
    rmse];
dlmwrite(txt,towrite,'-append');

MASK = Corr_ASTER;

```



```
end
```

```
outname = strcat(Savepath,date, '_',time, '_',vers, '  
_Tilt_Corrected.tif');  
geotiffwrite(outname, int16(Corr_ASTER), ASTER_info.  
SpatialRef, 'CoordRefSysCode', ASTER_info.GeoTIFFCodes.  
PCS);
```

```
exit
```

Appendix G. Matlab Code - DEM Correction: Step 3 - Aspect Correction

```
function Aspect_Correction(area,date,time,vers,ASTpath,
    SRTMpath,Savepath,Corrstats,MSK, figureout)
%Aspect_Correction(area, date, time, version, Slave Path,
    Master Path, save path, correlation stats, Mask, outpath
    for figure)
%   Performs a correction to fix aspect bias in the slave
    DEM.
%   Functions by fitting a plane to common high-slope
%   surfaces between the two DEMs.

%Created by Taylor Smith, March 2013

disp(strcat(date,' being processed for Aspect Correction.'))
%Import SRTM DEM
SRTM = SRTMpath;
[SRTM_dem,SRTM_refmat,SRTM_bbox] = geotiffread(SRTM);
SRTM_info = geotiffinfo(SRTM);
SRTM_dem = single(SRTM_dem); %Integerize SRTM DEM
idx0 = find(SRTM_dem <= 0); SRTM_dem(idx0) = NaN; %Recast
    nodata as NaN

%Import ASTER DEM
ASTER = ASTpath;
[AST_dem,AST_refmat,AST_bbox] = geotiffread(ASTER);
ASTER_info = geotiffinfo(ASTER);
AST_dem = single(AST_dem);
idxnan = find(AST_dem <= 0); AST_dem(idxnan) = NaN;

%Perform masking if desired
if strcmp(MSK, 'NULL') ~= 1
    [AST_msk,AST_msk_refmat,AST_msk_bbox] = geotiffread(MSK)
        ;
    idxmsk = find(AST_msk >= 1 & AST_msk < 255);
    if length(idxmsk) == numel(AST_msk)
        idxmsk = find(AST_msk == 127);
    end
    AST_dem_msk = AST_dem; AST_dem_msk(idxmsk) = NaN;
    SRTM_dem(idxmsk) = NaN;
    MASK = single(AST_dem_msk);
end
if strcmp(MSK, 'NULL') == 1
```

```

        MASK = single(AST_dem);
end

slope_threshold = 20;
dh_m_threshold = 200;
nr_of_random_points = 1e5;
STD_dif = 0.02;
warning off

%Create gridbox for use in rest of loop
DEM_X_size = AST_refmat(2,1); DEM_Y_size = AST_refmat(2,1);
[AST_utm_y, AST_utm_x] = meshgrid(AST_bbox(3):DEM_Y_size:
    AST_bbox(4)-DEM_Y_size, AST_bbox(1):DEM_X_size:AST_bbox
    (2)-DEM_X_size);
AST_utm_y = rot90(AST_utm_y); AST_utm_x = rot90(AST_utm_x);

%%%%% ASPECT CORRECTION %%%%%%
[fx, fy] = gradient(MASK, DEM_X_size, DEM_Y_size);
AST_SLOPE = sqrt(fx.^2 + fy.^2); clear fx fy
AST_SLOPE_DEG = atand(AST_SLOPE);
AST_ASPECT = aspect(AST_utm_x, AST_utm_y, AST_dem);

%Extract slope and DEM Differences
AST_dem_idx = find(MASK > 0);

ASTER_DEM = single(AST_dem(AST_dem_idx));
SRTM_DEM = single(SRTM_dem(AST_dem_idx));
ASTER_ASPECT = AST_ASPECT(AST_dem_idx);
ASTER_SLOPE = AST_SLOPE_DEG(AST_dem_idx);

%Grab only above slope threshold points for aspect fitting
slope_idx = find(ASTER_SLOPE > slope_threshold & ASTER_SLOPE
    < 90);
tmp_aspect = ASTER_ASPECT(slope_idx);
tmp_slope = ASTER_SLOPE(slope_idx);
tmp_aster = ASTER_DEM(slope_idx);
tmp_srtm = SRTM_DEM(slope_idx);

DH = tmp_srtm - tmp_aster;
dh_index = find(abs(DH) < dh_m_threshold);

tmp_aspect = tmp_aspect(dh_index);
tmp_slope = tmp_slope(dh_index);
tmp_aster = tmp_aster(dh_index);
tmp_srtm = tmp_srtm(dh_index);

```

```

tmp_dh = DH(dh_index);
dh_tan = tmp_dh ./ tand(tmp_slope);

%Choose random points for fit and get 'c' coefficient
idx2plot = randi(length(tmp_slope), 1, nr_of_random_points);
c = mean(tmp_dh(idx2plot))./(tand(mean(tmp_slope(idx2plot))))
    );

aster_aspect_data = double(tmp_aspect(idx2plot));
dh_tan_data = double(dh_tan(idx2plot));
aster_slope_data = double(tmp_slope(idx2plot));
dh_tan_0 = [0; 0; 0]; %Set start position

%Setup a multistart problem that tries many different values
model=@(x,aster_aspect_data) x(1)*cosd(x(2)-
    aster_aspect_data)+x(3);
opt_options = optimset('lsqcurvefit');
opt_options.Display = 'off';
problem = createOptimProblem('lsqcurvefit','objective',
    model, 'xdata', aster_aspect_data, 'ydata', dh_tan_data,
    'x0',dh_tan_0,'lb', [0 0 -100], 'ub', [200 360 100],'
    options',opt_options);
%Use Multistart (and not GlobalSearch), because Globalsearch
    doesn't work
%with lsqcurvefit
ms=MultiStart('Display', 'off');
[dh_tan_fit,resnorm,flag,outpt,allmins] = run(ms,problem,20)
    ;

%calculate mean bias (following equation 4 in Nuth and Kaab,
    2011):
dh_mean = dh_tan_fit(3) .* tand(mean(aster_slope_data));

h = figure;
clf
subplot(2,1,1,'align')
plot(aster_slope_data, dh_tan_data, '.'), grid
title('DEM Finite Difference Slope vs. dh/tan(\alpha)', '
    Fontsize', 12)
xlabel('Terrain slope (degree)', 'Fontsize', 12), ylabel('dh
    /tan(\alpha) (meters)', 'Fontsize', 12)
axis([0 90 -100 100])

subplot(2,1,2,'align')
plot(aster_aspect_data, dh_tan_data, 'b.'), grid, hold

```

```

plot([1:1:360], equation3(dh_tan_fit, [1:1:360]), 'k-', '
    linewidth', 2)
title('DEM aspect vs. dh/tan(\alpha)', 'FontSize', 12)
xlabel('Terrain Aspect (degree), 0=North, clockwise', '
    Fontsize', 12), ylabel('dh/tan(\alpha) (meters)', '
    Fontsize', 12)
axis([0 360 -100 100])
text2print = sprintf('%s: dh-mean: %2.2f m\ na: %2.2f, b:
    %2.2f, c: %2.2f', date, dh_mean, dh_tan_fit(1),
    dh_tan_fit(2), dh_tan_fit(3));
text(10, 60, text2print, 'FontSize', 16);
set(gcf, 'OuterPosition', [49 162 923 899]);
print(h, '-djpeg90', figureout);

%adding mean bias to ASTER DEM (this is likely to be small)
Corr_aspect_ASTER = single(AST_dem) + dh_mean;

%adding shift vector to coordinates
if (dh_tan_fit(2) <= 90 & dh_tan_fit(2) > 0)
    %positive x and y shift
    yshift = sind(dh_tan_fit(2))*dh_tan_fit(1);
    xshift = cosd(dh_tan_fit(2))*dh_tan_fit(1);
elseif (dh_tan_fit(2) > 90 & dh_tan_fit(2) <=180)
    %positive x and negative y shift
    yshift = abs(sind(dh_tan_fit(2)))*dh_tan_fit(1)*(-1);
    xshift = abs(cosd(dh_tan_fit(2)))*dh_tan_fit(1);
elseif (dh_tan_fit(2) > 180 & dh_tan_fit(2) <= 270)
    %negative x and negative y shift
    yshift = abs(sind(dh_tan_fit(2)))*dh_tan_fit(1)*(-1);
    xshift = abs(cosd(dh_tan_fit(2)))*dh_tan_fit(1)*(-1);
elseif (dh_tan_fit(2) > 270 & dh_tan_fit(2) <= 360)
    %negative x and positive y shift
    yshift = abs(sind(dh_tan_fit(2)))*dh_tan_fit(1);
    xshift = abs(cosd(dh_tan_fit(2)))*dh_tan_fit(1)*(-1);
end
%Fix spatial referencing to account for shifts
ASTER_info.SpatialRef.XLimWorld = [(ASTER_info.SpatialRef.
    XLimWorld(1) + xshift) (ASTER_info.SpatialRef.XLimWorld
    (2) + xshift)];
ASTER_info.SpatialRef.YLimWorld = [(ASTER_info.SpatialRef.
    YLimWorld(1) + yshift) (ASTER_info.SpatialRef.YLimWorld
    (2) + yshift)];

Corr_ASTER = Corr_aspect_ASTER;

```

```

txt = strcat(Corrstats,date,'_',time,'_',vers,'
    _AspectCorrectionParameters.csv');
towrite = [dh_tan_fit(1) dh_tan_fit(2) dh_tan_fit(3) resnorm
    (1) dh_mean nanmean(tmp_dh) nanstd(tmp_dh) xshift yshift
    ];
dlmwrite(txt,towrite,'-append');

tmp_dh = SRTM_dem - Corr_ASTER;
news = strcat(Corrstats,'Aspect_looper.txt');
fileID = fopen(news,'w');
fprintf(fileID,'%s,%s',num2str(nanmean(tmp_dh(:))), num2str(
    nanstd(tmp_dh(:))));
fclose(fileID);

outname = strcat(Savepath,date,'_',time,'_',vers,'
    _Aspect_Corrected.tif');
geotiffwrite(outname, int16(Corr_ASTER), ASTER_info.
    SpatialRef, 'CoordRefSysCode', ASTER_info.GeoTIFFCodes.
    PCS);
exit

```

Appendix H. Matlab Code - DEM Correction: Step 4 - Elevation Correction

```
function Elevation_Correction(area,date,time,vers,AST,SRTM,
    Savepath,Corrstats,MSK)
%Elevation_Correction(area, date, time, version, Slave Path,
    Master Path, save path, correlation stats, Mask)
% Performs a correction to fix aspect bias in the slave
    DEM.
% Functions by fitting a plane to common high-slope
% surfaces between the two DEMs.

%Created by Taylor Smith, March 2013

disp(strcat(date,' being processed for Elevation Correction.
    '))

%Import SRTM DEM
[SRTM_dem,SRTM_refmat,SRTM_bbox] = geotiffread(SRTM);
SRTM_info = geotiffinfo(SRTM);
SRTM_dem = single(SRTM_dem); %Integerize SRTM DEM
idx0 = find(SRTM_dem <= 0); SRTM_dem(idx0) = NaN; %Recast
    nodata as NaN

%Import ASTER DEM
[AST_dem,AST_refmat,AST_bbox] = geotiffread(AST);
ASTER_info = geotiffinfo(AST);
AST_dem = single(AST_dem);
idxnan = find(AST_dem <= 0); AST_dem(idxnan) = NaN;

%Do masking if desired
if strcmp(MSK, 'NULL') ~= 1
    [AST_msk,AST_msk_refmat,AST_msk_bbox] = geotiffread(MSK)
        ;
    idxmsk = find(AST_msk >= 1 & AST_msk < 255);
    if length(idxmsk) == numel(AST_msk)
        idxmsk = find(AST_msk == 127);
    end
    AST_dem_msk = AST_dem; AST_dem_msk(idxmsk) = NaN;
    SRTM_dem(idxmsk) = NaN;
    MASK = single(AST_dem_msk);
end
if strcmp(MSK, 'NULL') == 1
    MASK = single(AST_dem);
```

```

end

%Create gridbox for use in rest of loop
DEM_X_size = AST_refmat(2,1); DEM_Y_size = AST_refmat(2,1);
[AST_utm_y, AST_utm_x] = meshgrid(AST_bbox(3):DEM_Y_size:
    AST_bbox(4)-DEM_Y_size, AST_bbox(1):DEM_X_size:AST_bbox
    (2)-DEM_X_size);
AST_utm_y = rot90(AST_utm_y); AST_utm_x = rot90(AST_utm_x);

%Apply Elevation Correction
dh_m_threshold = 100;
nr_of_random_points = 1e6;
STD_dif = 0.02;
warning off

%Extract slope and DEM Differences
AST_dem_idx = find(MASK > 0);

ASTER_DEM = single(AST_dem(AST_dem_idx));
SRTM_DEM = single(SRTM_dem(AST_dem_idx));

DH = SRTM_DEM - ASTER_DEM;
dh_index = find(abs(DH) < dh_m_threshold);

tmp_aster = ASTER_DEM(dh_index);
tmp_srtm = SRTM_DEM(dh_index);
tmp_dh = DH(dh_index);

%Choose random subset
idx2plot = randi(length(tmp_aster), 1, nr_of_random_points);

aster_elev_data = tmp_aster(idx2plot);
dif_data = tmp_dh(idx2plot);

[xData, yData] = prepareCurveData(aster_elev_data, dif_data)
;

% Set up fittype and options.
ft = fittype( 'poly2' );
opts = fitoptions( ft );
opts.Lower = [-Inf -Inf -Inf -Inf];
%opts.Robust = 'Bisquare';
opts.Upper = [Inf Inf Inf Inf];
%opts.Weights = wData;

```



```

% Fit model to data.
[fitresult, gof] = fit( xData, yData, ft, opts );
figure;plot(xData, yData, '.'), hold, grid, plot(fitresult,
    'k-');

Newplane = fitresult.p1 .*(AST_dem).^2 + fitresult.p2 .*(
    AST_dem) + fitresult.p3;

Corrected_AST = single(AST_dem) - single(Newplane);

txt = strcat(Corrstats,date,'_',time,'_',vers,'
    _ElevCorrectionParameters.csv');
towrite = [fitresult.p1 fitresult.p2 fitresult.p3 gof.sse
    gof.rsquare gof.dfe gof.adjrsquare gof.rmse];
dlmwrite(txt,towrite,'-append');

outname = strcat(Savepath,date,'_',time,'_',vers,'
    _Elevation_Corrected.tif');
geotiffwrite(outname, int16(Corrected_AST), ASTER_info.
    SpatialRef, 'CoordRefSysCode', ASTER_info.GeoTIFFCodes.
    PCS);
exit

```

Appendix I. Matlab Code - DEM Correction: Step 5 - DEM Comparison

```
function compare_DEMs(AST_init,SRTM_init,MSK,AST_imregister,
    Reg_SRTM,SRTM_aspect,Corrpath,figpth)
%compare_DEMs(Initial Slave, Initial Master, Mask,
    Registered ASTER, Registered SRTM, Aspect corrected SRTM,
    Correlation path, figure outpath)
% This simply compares all of the datasets to show
    corrections.

%Created by Taylor Smith and Bodo Bookhagen, March 2013

figureout = strcat(figpth, '_Corrections_hist.jpg');
figureout2 = strcat(figpth, '_Corrections_dh.jpg');
AST_elev = strcat(Corrpath, '_Elevation_Corrected.tif');
AST_aspect = strcat(Corrpath, '_Aspect_Corrected.tif');
AST_tilt = strcat(Corrpath, '_Tilt_Corrected.tif');

dh_bins = -500:10:500;
%1. SRTM and uncorrected ASTER (from ENVI)
[SRTM_dem,SRTM_refmat,SRTM_bbox] = geotiffread(SRTM_init);
SRTM_dem = single(SRTM_dem); %Integerize SRTM DEM
idx0 = find(SRTM_dem <= 0); SRTM_dem(idx0) = NaN; %Recast
    no data as NaN
%Import ASTER DEM
[AST_dem,AST_refmat,AST_bbox] = geotiffread(AST_init);
AST_dem = single(AST_dem);
idxnnan = find(AST_dem <= 0); AST_dem(idxnnan) = NaN;
if strcmp(MSK, 'NULL') == 0
    %remove areas stored in ASTER_mask
    [AST_msk,AST_msk_refmat,AST_msk_bbox] = geotiffread(MSK)
        ;
    idxnnan = find(AST_msk >= 1 & AST_msk < 255);
    if length(idxnnan) == numel(AST_msk)
        idxnnan = find(AST_msk == 127);
    end
    AST_dem(idxnnan) = NaN;
end

Dh1_SRTM_ASTER = SRTM_dem-AST_dem; Dh1_SRTM_ASTER = reshape(
    Dh1_SRTM_ASTER, 1, numel(Dh1_SRTM_ASTER));
Dh1_SRTM_ASTER_hist = histc(Dh1_SRTM_ASTER, dh_bins);
Dh1_SRTM_ASTER_stats = [nanmean(Dh1_SRTM_ASTER) nanstd(
```

```

    Dh1_SRTM_ASTER) nanmedian(Dh1_SRTM_ASTER) prctile(
    Dh1_SRTM_ASTER, [1 5 10 25 50 75 90 95 99]);
Dh1_IQ = [AverageDifference(SRTM_dem, AST_dem)
    MaximumDifference(SRTM_dem, AST_dem) MeanSquareError(
    SRTM_dem, AST_dem) NormalizedAbsoluteError(SRTM_dem,
    AST_dem) PeakSignaltoNoiseRatio(SRTM_dem, AST_dem)
    NormalizedCrossCorrelation(SRTM_dem, AST_dem)
    StructuralContent(SRTM_dem,AST_dem)];
%convert SRTM_DEM to uint8 for wavelet analysis
clear AST_dem* SRTM_dem*

%2. SRTM and imregistered ASTER
[SRTM_dem,SRTM_refmat,SRTM_bbox] = geotiffread(Reg_SRTM);
SRTM_dem = single(SRTM_dem); %Integerize SRTM DEM
idx0 = find(SRTM_dem <= 0); SRTM_dem(idx0) = NaN; %Recast
    nodata as NaN % was ==0 %%%%
%Import ASTER DEM
[AST_dem,AST_refmat,AST_bbox] = geotiffread(AST_imregister);
AST_dem = single(AST_dem);
idxnnan = find(AST_dem <= 0); AST_dem(idxnnan) = NaN;
if strcmp(MSK, 'NULL') == 0
    %remove areas stored in ASTER_mask
    [AST_msk,AST_msk_refmat,AST_msk_bbox] = geotiffread(MSK)
        ;
    idxnnan = find(AST_msk >= 1 & AST_msk < 255);
    if length(idxnnan) == numel(AST_msk)
        idxnnan = find(AST_msk == 127);
    end
    AST_dem(idxnnan) = NaN;
end
Dh2_SRTM_ASTER = SRTM_dem-AST_dem; Dh2_SRTM_ASTER = reshape(
    Dh2_SRTM_ASTER, 1, numel(Dh2_SRTM_ASTER));
Dh2_SRTM_ASTER_hist = histc(Dh2_SRTM_ASTER, dh_bins);
Dh2_SRTM_ASTER_stats = [nanmean(Dh2_SRTM_ASTER) nanstd(
    Dh2_SRTM_ASTER) nanmedian(Dh2_SRTM_ASTER) prctile(
    Dh2_SRTM_ASTER, [1 5 10 25 50 75 90 95 99])];
Dh2_IQ = [AverageDifference(SRTM_dem, AST_dem)
    MaximumDifference(SRTM_dem, AST_dem) MeanSquareError(
    SRTM_dem, AST_dem) NormalizedAbsoluteError(SRTM_dem,
    AST_dem) PeakSignaltoNoiseRatio(SRTM_dem, AST_dem)
    NormalizedCrossCorrelation(SRTM_dem, AST_dem)
    StructuralContent(SRTM_dem,AST_dem)];
clear AST_dem* SRTM_dem*

%3. SRTM and Tilt Correction

```

```

[SRTM_dem,SRTM_refmat,SRTM_bbox] = geotiffread(Reg_SRTM);
SRTM_dem = single(SRTM_dem); %Integerize SRTM DEM
idx0 = find(SRTM_dem <= 0); SRTM_dem(idx0) = NaN; %Recast
    nodata as NaN
%Import ASTER DEM
[AST_dem,AST_refmat,AST_bbox] = geotiffread(AST_tilt);
AST_dem = single(AST_dem);
idxnan = find(AST_dem <= 0); AST_dem(idxnan) = NaN;
if strcmp(MSK, 'NULL') == 0
    %remove areas stored in ASTER_mask
    [AST_msk,AST_msk_refmat,AST_msk_bbox] = geotiffread(MSK)
        ;
    idxnan = find(AST_msk >= 1 & AST_msk < 255);
    if length(idxnan) == numel(AST_msk)
        idxnan = find(AST_msk == 127);
    end
    AST_dem(idxnan) = NaN;
end
Dh3_SRTM_ASTER = SRTM_dem-AST_dem; Dh3_SRTM_ASTER = reshape(
    Dh3_SRTM_ASTER, 1, numel(Dh3_SRTM_ASTER));
Dh3_SRTM_ASTER_hist = histc(Dh3_SRTM_ASTER, dh_bins);
Dh3_SRTM_ASTER_stats = [nanmean(Dh3_SRTM_ASTER) nanstd(
    Dh3_SRTM_ASTER) nanmedian(Dh3_SRTM_ASTER) prctile(
    Dh3_SRTM_ASTER, [1 5 10 25 50 75 90 95 99])];
Dh3_IQ = [AverageDifference(SRTM_dem, AST_dem)
    MaximumDifference(SRTM_dem, AST_dem) MeanSquareError(
    SRTM_dem, AST_dem) NormalizedAbsoluteError(SRTM_dem,
    AST_dem) PeakSignaltoNoiseRatio(SRTM_dem, AST_dem)
    NormalizedCrossCorrelation(SRTM_dem, AST_dem)
    StructuralContent(SRTM_dem,AST_dem)];
clear AST_dem* SRTM_dem*

%4. SRTM and Aspect Correction
[SRTM_dem,SRTM_refmat,SRTM_bbox] = geotiffread(SRTM_aspect);
SRTM_dem = single(SRTM_dem); %Integerize SRTM DEM
idx0 = find(SRTM_dem <= 0); SRTM_dem(idx0) = NaN; %Recast
    nodata as NaN
%Import ASTER DEM
[AST_dem,AST_refmat,AST_bbox] = geotiffread(AST_aspect);
AST_dem = single(AST_dem);
idxnan = find(AST_dem <= 0); AST_dem(idxnan) = NaN;
if strcmp(MSK, 'NULL') == 0
    %remove areas stored in ASTER_mask
    [AST_msk,AST_msk_refmat,AST_msk_bbox] = geotiffread(MSK)
        ;

```

```

    idxnnan = find(AST_msk >= 1 & AST_msk < 255);
    if length(idxnnan) == numel(AST_msk)
        idxnnan = find(AST_msk == 127);
    end
    AST_dem(idxnnan) = NaN;
end
Dh4_SRTM_ASTER = SRTM_dem-AST_dem; Dh4_SRTM_ASTER = reshape(
    Dh4_SRTM_ASTER, 1, numel(Dh4_SRTM_ASTER));
Dh4_SRTM_ASTER_hist = histc(Dh4_SRTM_ASTER, dh_bins);
Dh4_SRTM_ASTER_stats = [nanmean(Dh4_SRTM_ASTER) nanstd(
    Dh4_SRTM_ASTER) nanmedian(Dh4_SRTM_ASTER) prctile(
    Dh4_SRTM_ASTER, [1 5 10 25 50 75 90 95 99])];
Dh4_IQ = [AverageDifference(SRTM_dem, AST_dem)
    MaximumDifference(SRTM_dem, AST_dem) MeanSquareError(
    SRTM_dem, AST_dem) NormalizedAbsoluteError(SRTM_dem,
    AST_dem) PeakSignaltoNoiseRatio(SRTM_dem, AST_dem)
    NormalizedCrossCorrelation(SRTM_dem, AST_dem)
    StructuralContent(SRTM_dem,AST_dem)];
clear AST_dem* SRTM_dem*

%5. SRTM and Elevation Correction
[SRTM_dem,SRTM_refmat,SRTM_bbox] = geotiffread(SRTM_aspect);
SRTM_dem = single(SRTM_dem); %Integerize SRTM DEM
idx0 = find(SRTM_dem <= 0); SRTM_dem(idx0) = NaN; %Recast
    nodata as NaN
%Import ASTER DEM
[AST_dem,AST_refmat,AST_bbox] = geotiffread(AST_elev);
AST_dem = single(AST_dem);
idxnnan = find(AST_dem <= 0); AST_dem(idxnnan) = NaN;
if strcmp(MSK, 'NULL') == 0
    %remove areas stored in ASTER_mask
    [AST_msk,AST_msk_refmat,AST_msk_bbox] = geotiffread(MSK)
        ;
    idxnnan = find(AST_msk >= 1 & AST_msk < 255);
    if length(idxnnan) == numel(AST_msk)
        idxnnan = find(AST_msk == 127);
    end
    AST_dem(idxnnan) = NaN;
end
Dh5_SRTM_ASTER = SRTM_dem-AST_dem; Dh5_SRTM_ASTER = reshape(
    Dh5_SRTM_ASTER, 1, numel(Dh5_SRTM_ASTER));
Dh5_SRTM_ASTER_hist = histc(Dh5_SRTM_ASTER, dh_bins);
Dh5_SRTM_ASTER_stats = [nanmean(Dh5_SRTM_ASTER) nanstd(
    Dh5_SRTM_ASTER) nanmedian(Dh5_SRTM_ASTER) prctile(
    Dh5_SRTM_ASTER, [1 5 10 25 50 75 90 95 99])];

```

```

Dh5_IQ = [AverageDifference(SRTM_dem, AST_dem)
          MaximumDifference(SRTM_dem, AST_dem) MeanSquareError(
          SRTM_dem, AST_dem) NormalizedAbsoluteError(SRTM_dem,
          AST_dem) PeakSignaltoNoiseRatio(SRTM_dem, AST_dem)
          NormalizedCrossCorrelation(SRTM_dem, AST_dem)
          StructuralContent(SRTM_dem,AST_dem)];
clear AST_dem* SRTM_dem*

Dh_IQ = [Dh1_IQ; Dh2_IQ; Dh3_IQ; Dh4_IQ; Dh5_IQ];
Dh_stats = [Dh1_SRTM_ASTER_stats; Dh2_SRTM_ASTER_stats;
            Dh3_SRTM_ASTER_stats; Dh4_SRTM_ASTER_stats;
            Dh5_SRTM_ASTER_stats];

%% Plot Figure %%

h = figure;
clf
%subplot(1,2,1,'align')
plot(dh_bins, Dh1_SRTM_ASTER_hist./1e3, 'k-', 'Linewidth',
     2), hold, grid,
plot(dh_bins, Dh2_SRTM_ASTER_hist./1e3, 'r-', 'Linewidth',
     2),
plot(dh_bins, Dh3_SRTM_ASTER_hist./1e3, 'b-', 'Linewidth',
     2),
plot(dh_bins, Dh4_SRTM_ASTER_hist./1e3, 'm-', 'Linewidth',
     2),
plot(dh_bins, Dh5_SRTM_ASTER_hist./1e3, 'g-', 'Linewidth',
     2)
legend('Uncorrected', 'IMREGISTER', 'Tilt Corrected', '
      Aspect Corrected', 'Elevation Corrected', 'Location', '
      Northeast')
xlabel('\Delta h (m) between SRTM and ASTER', 'FontSize',
     14), ylabel('# of points in 10-m bins x10^3', 'FontSize',
     14)
title('Histograms of \Delta h (elevation difference between
      SRTM and ASTER DEM) in m', 'FontSize', 14)
axis([-200 200 0 5000])

%subplot(1,2,2,'align')
h2 = figure;
clf
plot([1:5], Dh_stats(:,1), 'k-', 'Linewidth', 2), hold, grid
,
plot([1:5], Dh_stats(:,3), 'r-', 'Linewidth', 2),
plot([1:5], Dh_stats(:,6)./10, 'b-', 'Linewidth', 1),

```

```

plot([1:5], Dh_stats(:,10)./10, 'b-', 'Linewidth', 1),
plot([1:5], Dh_stats(:,4)./10, 'm-', 'Linewidth', 1),
plot([1:5], Dh_stats(:,end)./10, 'm-', 'Linewidth', 1),
legend('mean \Delta h', 'median \Delta h', 'lower 10^{th}
percentile x10', 'upper 90^{th} percentile x10', 'lower
1^{th} percentile x10', 'upper 99^{th} percentile x10')
ylabel('\Delta h (m) between SRTM and ASTER', 'FontSize',
14), xlabel('Step #', 'FontSize', 14)
title('Mean, median, and 1,10, 90, 99^{th} percentile of \
Delta h (elevation difference between SRTM and ASTER DEM)
in m', 'FontSize', 14)
axis([1 5 -100 100])
set(h, 'OuterPosition', [1 1 1920 1080]);
set(h2, 'OuterPosition', [1 1 1920 1080]);

print(h, '-djpeg', '-r300', figureout);
print(h2, '-djpeg', '-r300', figureout2);

exit

```