



**Universität  
Zürich**<sup>UZH</sup>

Geographisches Institut

# **Automatische Identifizierung potentieller Lawinenanrissgebiete**

---

**Evaluierung und Vergleich verschiedener Algorithmen  
im Kontext grossflächiger Gefahrenhinweiskartierungen**

GEO 511 Masterarbeit

**Eingereicht durch**

Daniel von Rickenbach

12-718-128

**Betreut durch:**

Prof. Dr. Christian Huggel

Dr. Yves Bühler (buehler@slf.ch)

**Fakultätsvertretung:**

Prof. Dr. Andreas Vieli

30.09.2018

Geographisches Institut, Universität Zürich



# Vorwort

Meine Interessen für die Natur und Umwelt haben sich während des Studiums immer stärker Richtung alpine Naturgefahren entwickelt. Mit der Masterarbeit erhielt ich Einblick in die angewandte Forschung des SLFs (WSL-Institut für Schnee und Lawinenforschung) in diesem Themenbereich. In der Forschungsgruppe Lawindynamik und Risikomanagement beschäftigte ich mich mit *large scale hazard indication mapping for avalanches*. Ich arbeitete an der Schnittstelle zwischen grossflächigen GIS-Kartierungen und numerischen Lawinensimulationen. Dabei war ich bei der Entwicklung neuer SLF Produkte, insbesondere bei der Entwicklung eines neuen Algorithmus zur Identifizierung potentieller Anrissgebiete (Bühler et al. 2018), involviert.

Ein grosser Dank gilt meinen Betreuern Dr. Yves Bühler (SLF) und PD Dr. Christian Huggel (UZH), welche mir diese Arbeit ermöglicht und mich während des ganzen Arbeitsprozesses begleitet haben. Sie standen mir immer mit konstruktiver Kritik und Ratschlägen zur Seite. Dank Yves Bühler erhielt ich sogar Einblick ins wissenschaftliche Publizieren. Weiter möchte ich mich bei Stephan Harvey (SLF) für die Aufbereitung des Referenzdatensatzes und die hilfreiche Unterstützung bei Fragen zur Evaluierung bedanken. Mein Dank gilt ebenfalls den aktuellen und ehemaligen Pisten- und Rettungsverantwortlichen von Davos Klosters Mountains. Im Austausch mit Romano Pajarola, Valentin Meier und Nigg Conrad konnte der verwendete Referenzdatensatz korrigiert und ergänzt werden. Schliesslich möchte ich mich auch noch bei Peter von Rickenbach, Yvonne Ehrensberger, Silvan Schwegler und Andreas Stoffel (SLF) für das Korrekturlesen bedanken.

Davos, September 2018

# Zusammenfassung

In vielen alpinen Regionen der Erde stellen Schneelawinen eine Gefahr für den Menschen und dessen Infrastruktur dar. Das Leben mit dieser Gefahr ist eine grosse Herausforderung und im Laufe der Zeit wurden verschiedene Strategien entwickelt, um damit umzugehen. Die Strategien reichen von der Raumplanung bis hin zu baulichen Massnahmen. In der Schweiz erstellen Lawinenexperten Gefahrenkarten für Siedlungsgebiete, welche durch Lawinen gefährdet sein könnten. Für Regionen ausserhalb der ausgewählten Siedlungsgebiete sind solch zuverlässige Gefahrenkarten jedoch nicht verfügbar. Daher sind Gefahrenhinweiskarten, auch wenn sie weniger detailliert sind, oft das einzige verfügbare Raumplanungsinstrument. Eine automatische Identifizierung potentieller Anrissgebiete in Kombination mit numerischer Lawinensimulation ermöglicht die Erstellung von Gefahrenhinweiskarten für Lawinen über grosse Gebiete. Der erste entscheidende Schritt bei der automatischen Gefahrenhinweiskartierung ist daher die genaue, automatisierte Identifizierung potenzieller Anrissgebiete.

In der Vergangenheit wurden verschiedene Algorithmen entwickelt, die hauptsächlich auf geländespezifischen Parametern basierten, um potentielle Anrissgebiete zu identifizieren. Bisher wurden diese Algorithmen noch nicht miteinander für denselben Referenzdatensatz getestet und verglichen. Aus diesem Grund wurden für diese Arbeit vier Algorithmen ausgewählt und anhand eines Referenzdatensatzes evaluiert und miteinander verglichen. Als Referenzdatensatz dienten sorgfältig kartierte Lawinen dreier Gebiete der Region Davos in der Schweiz. Zur Evaluierung wurden die Algorithmen zuerst kalibriert und anschliessend validiert. Der anschliessende Vergleich zeigte: Je mehr Geländeparameter bei der Anrissgebietsidentifizierung berücksichtigt werden, desto besser sind die Validierungsergebnisse. Am besten hat dabei der Algorithmus von Bühler et al. (2018) abgeschnitten. Dabei unterteilt dieser Algorithmus die potentiellen Anrissflächen mittels objektbasierter Bildanalyse in individuelle Anrissgebiete. Dieser Ansatz öffnet, in Kombination mit numerischer Lawinensimulation, die Tür für grossflächige Gefahrenhinweiskartierungen in allen Regionen, in denen hochwertige digitale Geländemodelle zur Verfügung stehen.

## Summary

In many alpine regions of the world, snow avalanches pose a great threat to humans and their infrastructure. To overcome this challenge, various strategies have been developed over time to mitigate the problem. The strategies range from spatial planning to structural measures. In Switzerland, avalanche experts produce hazard maps for settlement areas that could be endangered by avalanches. However, such reliable hazard maps are not available for regions outside the selected settlements. Therefore, hazard indication maps, although less detailed, are often the only available spatial planning tool. An automatic identification of potential release areas along with numerical avalanche simulations allows for the automated creation of avalanche hazard indication maps over large areas. In order for this to be improved on, there must be a precise and automated identification procedure for potential release areas.

In the past, various algorithms have been developed, mainly based on terrain-specific parameters, to identify potential release areas. So far, these algorithms have not been tested and compared for the same reference data set. For this reason, four algorithms were selected for this work and evaluated using a single reference data set and compared with each other. Carefully mapped avalanches from three areas of the Davos region, Switzerland, served as reference data. For evaluation, the algorithms were first calibrated and then validated. The subsequent comparison showed that for the purpose of release area delineation the more terrain parameters taken into account, the better the validation results. The algorithm of Bühler et al. (2018) performed best, which divides the potential release areas into individual release zones using object-based image analysis. In combination with numerical avalanche simulation, this approach allows for large-scale avalanche hazard mapping to be available in all alpine regions where there is access to high-quality digital terrain models.

# Inhalt

<b>Vorwort</b> .....	<b>i</b>
<b>Zusammenfassung</b> .....	<b>ii</b>
<b>Summary</b> .....	<b>iii</b>
<b>Inhalt</b> .....	<b>iv</b>
<b>Abbildungen</b> .....	<b>vii</b>
<b>Tabellen</b> .....	<b>xi</b>
<b>Glossar</b> .....	<b>xii</b>
<b>1 Einleitung</b> .....	<b>1</b>
1.1 Arbeitsziele und Forschungsfragen .....	2
1.2 Aufbau der Arbeit .....	2
<b>2 Allgemeine Grundlagen</b> .....	<b>4</b>
2.1 Lawinen .....	4
2.1.1 Lawinenklassifikation .....	4
2.1.2 Lawinenbildende Faktoren der Schneebrettlawine .....	6
2.2 Gefahrenhinweis- und Gefahrenkarte .....	11
2.2.1 Gefahrenhinweiskarte .....	11
2.2.2 Gefahrenkarte .....	12
2.3 Geomorphometrie .....	12
2.3.1 Digitale Modelle der Erdoberfläche .....	13
2.3.2 Elementare Geländeparameter .....	15
2.4 Objektbasierte Bildanalyse .....	21
2.4.1 Multiskalige Segmentation .....	21

2.5	Räumliches Modellieren .....	22
<b>3</b>	<b>Algorithmen zur automatischen Identifizierung potentieller Anrissgebiete.....</b>	<b>24</b>
3.1	Forschungsstand .....	24
3.1.1	Bestehende Algorithmen .....	25
3.1.2	Ausgewählte Algorithmen.....	26
3.2	Verwendete Software .....	30
3.2.1	Software zur Geodatenverarbeitung .....	31
3.2.2	Statistikprogramm .....	31
3.3	Verwendetes DHM .....	31
3.3.1	Wahl der räumlichen Auflösung.....	32
<b>4</b>	<b>Methodik .....</b>	<b>33</b>
4.1	Referenzdatensatz.....	33
4.1.1	Auswahl der Lawinenumrisse.....	34
4.1.2	Ableitung der Anrissgebiete aus den Lawinenumrissen .....	35
4.1.3	Beschreibung der Untersuchungsgebiete .....	38
4.1.4	Beschreibung der Anrissgebiete des Referenzdatensatzes .....	40
4.2	Prognosebeurteilung .....	42
4.2.1	Prognosebeurteilung für binäre Klassifikationen ( <i>Boolean</i> ) .....	43
4.3	Ablauf der Evaluierung .....	46
4.3.1	Gütemasse zur Prognosebeurteilung .....	47
4.3.2	Trainings- und Testgebiete.....	48
4.3.3	Kalibrierung .....	48
4.3.4	Validierung und Vergleich .....	49
<b>5</b>	<b>Resultate .....</b>	<b>50</b>
5.1	Kalibrierung.....	50
5.1.1	Bühler et al. (2013) .....	50
5.1.2	Veitinger et al. (2016).....	51

5.1.3	Bühler et al. (2018) .....	54
5.2	Validierung und Vergleich .....	55
5.2.1	Parsenn .....	55
5.2.2	Jakobs- und Rinerhorn .....	59
<b>6</b>	<b>Diskussion .....</b>	<b>64</b>
6.1	Algorithmen zur automatischen Identifizierung potentieller Anrissgebiete .....	64
6.2	Methodik .....	64
6.3	Resultate .....	66
6.3.1	Kalibrierung.....	66
6.3.2	Validierung und Vergleich .....	68
6.4	Erkenntnisse aus dem Algorithmenvergleich .....	71
<b>7</b>	<b>Schlussfolgerungen.....</b>	<b>74</b>
<b>8</b>	<b>Literatur.....</b>	<b>76</b>
<b>9</b>	<b>Anhang .....</b>	<b>85</b>
9.1	Python-Skripte .....	85
9.1.1	Umsetzung Voellmy (1955) .....	85
9.1.2	Tabelle mit den Parameternwerten zur Kalibrierung .....	87
9.1.3	Kalibrierung Bühler et al. (2013).....	88
9.1.4	Kalibrierung Veitinger et al. (2016) .....	94
9.1.5	Kalibrierung Bühler et al. (2018).....	102
9.1.6	Validierung.....	110
9.2	R-Skript .....	128
9.2.1	Validierung.....	128
	<b>Persönliche Erklärung .....</b>	<b>140</b>



# Abbildungen

Abb. 1.1: Themenüberblick und Zusammenhänge dieser Arbeit.....	3
Abb. 2.1: Lawinenklassifikation analog zu jener der UNESCO (1981) aus Kienholz et al. (1998)	4
Abb. 2.2: Trockene (a) und nasse (b) Lockerschneelawine (Harvey et al. 2013) .....	5
Abb. 2.3: Unterteilung einer Schneebrettlawine in Teilbereiche (Harvey et al. 2013) .....	6
Abb. 2.4: Schematische Darstellung von Prozessen, die sich auf die lawinenbildenden Faktoren Schneedecke und Wetter beziehen (SLF 2018a) .....	10
Abb. 2.5: Gefahrenhinweiskarte, potentielle Gefahrengebiete in rot (Camenzind-Wildi et al. 2000) .....	11
Abb. 2.6: Gefahrenstufen als Resultat von Intensität und Wahrscheinlichkeit (Camenzind-Wildi et al. 2000) .....	12
Abb. 2.7: Gefahrenkarte (Camenzind-Wildi et al. 2000) .....	12
Abb. 2.8: Geomorphometrie im Kontext ihrer Ursprünge und ihrer Anwendungen (Pike et al. 2009) .....	13
Abb. 2.9: Schematische Darstellung der Werte abgespeichert in einem digitalen Oberflächen- und in einem digitalen Höhenmodell (Asharyanto et al. 2015) .....	14
Abb. 2.10: Rastermodell, definiert durch fünf Parameter (Pike et al. 2009) .....	14
Abb. 2.11: Vergleich verschiedener digitalen Modelle: Rastermodell (a) und Vektormodell als TIN (b) (Hengl und Evans 2009) .....	15
Abb. 2.12: Gleitendes Zellfenster (Olaya 2009) .....	15
Abb. 2.13: Rasteroberfläche in einem Koordinatensystem (Burrough et al. 2015) .....	15
Abb. 2.14: Geländeparameter exemplarisch dargestellt für die Region Weissfluhjoch: Hangneigung (a), Exposition, klassiert in die 8 Haupthimmelsrichtungen (b), Plankurvatur (c) und Rauigkeit (d) [*] .....	16
Abb. 2.15: Geländeformen bezüglich Profil- und Plankurvatur nach Dikau (1989) von Vontobel (2011).....	18
Abb. 2.16: Schematische Darstellung der Vektordispersion der Methode VRM (Sappington et al. 2007) .....	19

Abb. 2.17: Zerlegung eines Einheitsnormalvektors in seine x-, y- und z-Komponenten (Sappington et al. 2007) .....	20
Abb. 2.18: Berechnung des resultierenden Vektorbetrages $r$ (Sappington et al. 2007) .....	20
Abb. 2.19: Verhältnis zwischen dem zu untersuchenden Objekt und der räumlichen Auflösung: tiefe Auflösung (a): Pixel deutlich grösser als Objekt, subpixelbasierte Analyse nötig mittlere Auflösung (b): Pixel und Objekt in der gleichen Grössenordnung, pixelbasierte Analyse angebracht hohe Auflösung (c): Pixel sind deutlich kleiner als Objekte, Regionalisierung von Pixeln zu Pixelgruppen und schliesslich Objekten erforderlich (Blaschke 2010) .....	21
Abb. 2.20: Luftbild einer heterogenen Landschaft (a), feinskalige Segmentierung (b), grobskalige Segmentierung (c) und objektbasierte Klassifikation von Gehölzen (d) (Blaschke 2010) .....	21
Abb. 2.21: Zugehörigkeitsfunktionen graphisch dargestellt: <i>Boolean</i> (a) und <i>Fuzzy</i> (b) (Burrough et al. 2015) .....	22
Abb. 3.1: Schematischer Prozessablauf des Algorithmus nach Voellmy (1955) .....	27
Abb. 3.2: Schematischer Prozessablauf des Algorithmus nach Bühler et al. (2013) .....	28
Abb. 3.3: Schematischer Prozessablauf des Algorithmus nach Veitinger et al. (2016) .....	29
Abb. 3.4: 5x5 pixelgrosses, gleitendes Zellfenster mit dem entsprechenden Gewicht pro Pixel	29
Abb. 3.5: Schematischer Prozessablauf des Algorithmus nach Bühler et al. (2018) .....	30
Abb. 4.1: Unterteilung des Lawinenumrisses in einzelne Sektionen (Harvey et al. 2018a) .....	35
Abb. 4.2: Ableitung des relativen Längenverhältnisses einer schmalen, langgezogenen (a) und einer breiten, kurzen (b) Lawine (Harvey et al. 2018a) .....	36
Abb. 4.3: Primäres Anrissgebiet (Harvey et al. 2018a).....	36
Abb. 4.4: Sekundäres Anrissgebiet (Harvey et al. 2018a) .....	37
Abb. 4.5: Tertiäres Anrissgebiet (Harvey et al. 2018a) .....	37
Abb. 4.6: Region Davos mit den drei Gebieten Parsenn, Jakobs- und Rinerhorn [#] .....	38
Abb. 4.7: Verteilungen der geländespezifischen Eigenschaften (Höhe über Meer (a), Hangneigung (b), Plankurvatur (c) und Rauigkeit (d)) der drei Untersuchungsgebiete Parsenn, Jakobs- und Rinerhorn [*].....	39

Abb. 4.8: Lawinenanrissgebiete des Referenzdatensatzes für das Gebiet Jakobshorn (a), Rinerhorn (b) und Parsenn (c) [#] .....	41
Abb. 4.9: Verteilungen der geländespezifischen Eigenschaften (Höhe über Meer (a), Hangneigung (b), Plankurvatur (c) und Rauigkeit (d)) der Anrissgebiete der drei Untersuchungsgebiete Jakobshorn, Parsenn und Rinerhorn mit 5m räumlicher Auflösung [*].....	42
Abb. 4.10: Schematische Darstellung einer Konfusionsmatrix nach Congalton und Green (2009)	43
Abb. 5.1: Kalibrierung der Parameter Hangneigung (a) (b), Rauigkeit (c) (d), Plankurvatur (e) sowie Anrissfläche (f) des Algorithmus von Bühler et al. (2013) für das Trainingsgebiet Parsenn; Die blauen Kreuze markieren die optimalen Parameterwerte [*].....	51
Abb. 5.2: Zonierung der Abhängigkeit zwischen der Höhe über Meer und der 100-jährlichen Schneehöhe (Martinec 1990) .....	52
Abb. 5.3: Kalibrierung der Parameter mittlere Schneehöhe (a), Anrissanfälligkeit (b) sowie Anrissfläche (c) des Algorithmus von Veitinger et al. (2016) für das Trainingsgebiet Parsenn; Die blauen Kreuze markieren die optimalen Parameterwerte [*] .....	54
Abb. 5.4: Kalibrierung der Parameter Hangneigung (a) (b), Rauigkeit (c) (d), Plankurvatur (e) sowie Anrissfläche (f) des Algorithmus von Bühler et al. (2018) für das Trainingsgebiet Parsenn; Die blauen Kreuze markieren die optimalen Parameterwerte [*].....	55
Abb. 5.5: Lawinenanrissgebiete des Algorithmus Voellmy (1955) 28-60° für das Gebiet Parsenn [#*] .....	56
Abb. 5.6: Lawinenanrissgebiete des Algorithmus Bühler et al. (2013) das Gebiet Parsenn [#*]	57
Abb. 5.7: Lawinenanrissgebiete des Algorithmus Veitinger et al. (2016) für das Gebiet Parsenn [#*].....	57
Abb. 5.8: Lawinenanrissgebiete des Algorithmus Bühler et al. (2018) Vorauswahl für das Gebiet Parsenn [#*] .....	58
Abb. 5.9: Lawinenanrissgebiete des Algorithmus Bühler et al. (2018) <i>frequent</i> für das Gebiet Parsenn [#*] .....	58
Abb. 5.10: Lawinenanrissgebiete des Algorithmus Bühler et al. (2018) <i>extreme</i> für das Gebiet Parsenn [#*] .....	59
Abb. 5.11: Lawinenanrissgebiete des Algorithmus Voellmy (1955) 28-60° für das Gebiet Jakobshorn [#*] .....	60

Abb. 5.12: Lawinenanrissgebiete des Algorithmus Voellmy (1955) 28-60° für das Gebiet Rinerhorn [#*].....	60
Abb. 5.13: Lawinenanrissgebiete des Algorithmus Bühler et al. (2013) für das Gebiet Jakobshorn [#*].....	61
Abb. 5.14: Lawinenanrissgebiete des Algorithmus Bühler et al. (2013) für das Gebiet Rinerhorn [#*].....	61
Abb. 5.15: Lawinenanrissgebiete des Algorithmus Veitinger et al. (2016) für das Gebiet Jakobshorn [#*].....	61
Abb. 5.16: Lawinenanrissgebiete des Algorithmus Veitinger et al. (2016) für das Gebiet Rinerhorn [#*].....	61
Abb. 5.17: Lawinenanrissgebiete des Algorithmus Bühler et al. (2018) Vorauswahl für das Gebiet Jakobshorn [#*] .....	62
Abb. 5.18: Lawinenanrissgebiete des Algorithmus Bühler et al. (2018) Vorauswahl für das Gebiet Rinerhorn [#*] .....	62
Abb. 5.19: Lawinenanrissgebiete des Algorithmus Bühler et al. (2018) <i>frequent</i> für das Gebiet Jakobshorn [#*].....	62
Abb. 5.20: Lawinenanrissgebiete des Algorithmus Bühler et al. (2018) <i>frequent</i> für das Gebiet Rinerhorn [#*].....	62
Abb. 5.21: Lawinenanrissgebiete des Algorithmus Bühler et al. (2018) <i>extreme</i> für das Gebiet Jakobshorn [#*].....	63
Abb. 5.22: Lawinenanrissgebiete des Algorithmus Bühler et al. (2018) <i>extreme</i> für das Gebiet Rinerhorn [#*].....	63
Abb. 6.1: Numerische Lawinensimulationen auf Basis automatisch identifizierter Lawinenanrissgebiete [#*].....	73

# Tabellen

Tab. 2.1: Übersicht über die Eigenschaften von Gefahrenhinweiskarten und Gefahrenkarten (Camenzind-Wildi et al. 2000) .....	11
Tab. 3.1: Überblick über bestehende Algorithmen, inklusive der verwendeten Parameter .....	26
Tab. 4.1: <i>AvalancheOutlineFilterTool</i> mit einer Beschreibung und den für diese Arbeit verwendeten Filtereinstellungen .....	34
Tab. 5.1: Berechnung der mittleren 100-jährlichen Schneehöhe für das Gebietes Parsenn .....	53
Tab. 5.2: Validierungsergebnisse aller getesteten Algorithmen für das Trainingsgebiet Parsenn	56
Tab. 5.3: Validierungsergebnisse aller getesteten Algorithmen für das Testgebiet Jakobs- und Rinerhorn .....	59
Tab. 6.1: Validierungsergebnisse aller getesteten Algorithmen für das Testgebiet Jakobshorn ..	69
Tab. 6.2: Validierungsergebnisse aller getesteten Algorithmen für das Testgebiet Rinerhorn ....	70

# Glossar

SLF	WSL-Institut für Schnee- und Lawinenforschung SLF
WSL	Eidgenössische Forschungsanstalt für Wald, Schnee und Landschaft
Abb	Abbildung
Tab	Tabelle
PAG	Potentiellies Anrissgebiet einer Schneebrettlawine
nPAG	Nicht Potentiellies Anrissgebiet einer Schneebrettlawine
DHM	Digitales Höhenmodell
DOM	Digitales Oberflächenmodell
DGM	Digitales Geländemodell
TIN	<i>Triangular Irregular Network</i> , zu Deutsch: <i>Unregelmässiges Dreiecksnetz</i>
OBIA	<i>Object-based Image Analysis</i> , zu Deutsch: Objektbasierte Bildanalyse
GIS	Geographisches Informationssystem
°	Grad, Winkelmasseinheit, $1^\circ = \frac{\pi}{180} \text{ rad}$
rad	Radian, Winkelmasseinheit, $1 \text{ rad} = \frac{180^\circ}{\pi}$
m	Meter, Längenmasseinheit
m.ü.M.	Meter über dem Meeresspiegel
[#]	Kennzeichnet Kartendaten, wofür pixmaps©2018 swisstopo (5704 000 000), reproduziert mit Bewilligung von swisstopo (JA100118), verwendet wurde
[*]	Kennzeichnet Geländedaten, wofür DTM-AV DOM-AV©2018 swisstopo (5704 000 000), reproduziert mit Bewilligung von swisstopo (JA100118), verwendet wurde

# 1 Einleitung

Schneelawinen sind eine Naturgefahr in alpinen Regionen rund um den Globus, welche Menschen, Gebäude und Verkehrsinfrastrukturen gefährden. Während der letzten vier Jahrzehnte verursachten Lawinen in Europa durchschnittlich rund 100 Todesfälle pro Jahr. Allein in der Schweiz sterben pro Jahr durchschnittlich 25 Personen in Lawinen, die überwiegende Mehrheit bei Wintersportaktivitäten (Techel et al. 2016). Der Lawinenwinter 1998/1999 hatte in der Schweiz 17 Todesopfer und Sachschäden von insgesamt über 600 Millionen Schweizer Franken zur Folge (SLF 2000).

Die Schweiz verfügt über langjährige Erfahrung im Umgang mit Naturgefahren. Gefahren- und Gefahrenhinweiskarten dienen dabei als wichtige Raumplanungsinstrumente (Camenzind-Wildi et al. 2000, Gruber und Margreth 2001). Zur Naturgefahr wird ein Naturereignis erst dann, wenn dieses in Raum und Zeit in der Lage ist, dem Menschen und seinen Errungenschaften einen potentiellen Schaden zuzuführen (Dikau und Voss 2000).

Als Grundlage zur Abschätzung des Gefahrenpotentials von Lawinen dienen Katasterinformationen, Klimadaten, Geländeanalysen und numerische Simulationen der Lawinendynamik. All diese Informationen und Daten werden schliesslich von erfahrenen Experten zu Gefahrenkarten verarbeitet (Bründl und Romang 2009). Der Prozess der Gefahrenkartierung ist zeitaufwändig, fachlich sehr anspruchsvoll und deshalb teuer. Darum ist die Gefahrenkartierung nur auf einzelne Lawineneinzugsgebiete anwendbar, insbesondere solche, die Siedlungsgebiete und wichtige Infrastrukturen gefährden (Camenzind-Wildi et al. 2000).

Gefahrenhinweiskarten hingegen sind weniger detailliert als Gefahrenkarten, können aber einen räumlich kontinuierlichen Überblick über Naturgefahren auf der Grundlage numerischer Simulationen über grosse Gebiete geben. Dies ist insbesondere für Regionen mit spärlicher Ereignisdokumentation nützlich, wie es für viele alpine Regionen der Welt der Fall ist. Folglich besteht ein Interesse bezüglich automatisierter Gefahrenhinweiskartierung (AG NAGEF 2015, Bühler et al. 2017). Hochaufgelöste digitale Höhenmodelle, die mit Hilfe von modernen Fernerkundungsmethoden generiert werden können, sind zunehmend auch für alpine Regionen verfügbar (Bühler et al. 2012, Fonstad et al. 2013). Dies öffnet die Tür für aussagekräftige numerische Prozesssimulationen über grosse Gebiete, zum Beispiel für den Prozess Lawine (Bühler et al. 2011).

Um dynamische Lawinensimulationen mit moderner Simulationssoftware wie RAMMS (Christen et al. 2010) oder SAMOS (Sampl und Zwinger 2004) durchführen zu können, ist eine genaue Identifizierung der Anrissgebiete und -volumen erforderlich. Ein mögliches Auftreten einer Lawine lässt sich anhand der lawinenbildenden Faktoren abschätzen: geländespezifische (a), schneedeckenspezifische (b) und wetterspezifische (c) (Schweizer et al. 2003). In der Vergangenheit wurden verschiedene Algorithmen entwickelt, um potentielle Anrissgebiete (PAG) zu identifizieren. Die Algorithmen basierten hauptsächlich auf geländespezifischen Parametern (a). Bisher wurden diese Algorithmen zur Identifizierung potentieller Lawinenanrissgebiete noch nicht anhand desselben Referenzdatensatz evaluiert und untereinander verglichen.

## 1.1 Arbeitsziele und Forschungsfragen

Das Hauptziel dieser Studie ist es, eine Evaluierung und einen Vergleich der bestehenden Algorithmen auf der Basis von Lawinenkatasterinformationen in der Region Davos, Schweiz, durchzuführen. Diese Arbeit begrenzt sich auf die automatische Identifizierung potentieller Anrissgebiete. Werden zu diesen noch Anrisshöhen automatisch berechnet, wären dynamische Lawinensimulationen möglich, anhand derer die Möglichkeit bestünde, automatisch Gefahrenhinweiskarten zu produzieren. Deshalb sind die Algorithmen im Hinblick auf die Gefahrenhinweiskartierung zu evaluieren.

Die Erarbeitung wird in die folgenden Teilziele aufgeteilt:

- Überblick über bereits bestehende Algorithmen zur automatischen Identifizierung potentieller Lawinenanrissgebiete
- Erstellung eines aussagekräftigen Referenzdatensatzes und Evaluierungsverfahren im Hinblick auf die Gefahrenhinweiskartierung
- Evaluierung und Vergleich vier ausgewählter Algorithmen, welche automatisch potentielle Anrissgebiete identifizieren

Im Verlauf der Arbeit sollten dabei folgende Forschungsfragen beantwortet werden:

- Welche Algorithmen zur Identifizierung von Anrissgebieten existieren bereits?
- Welche verfügbaren Referenzdaten und welches Verfahren ermöglichen eine aussagekräftige Evaluierung der Algorithmen im Hinblick auf die Gefahrenhinweiskartierung?
- Wie gut schneiden die gewählten Algorithmen im Vergleich ab und welche Erkenntnisse lassen sich daraus gewinnen?

## 1.2 Aufbau der Arbeit

Nach der Einleitung werden im Kapitel 2 allgemein wichtige Grundlagen zu Lawinen, Gefahren- und Gefahrenhinweiskartierung, Geomorphometrie sowie objektbasierten Bildanalyse dargelegt. Im Kapitel 3 erfolgt die Einführung der Algorithmen zur automatischen Anrissgebietsidentifizierung. Dabei wird der Forschungsstand, die bestehenden Algorithmen, die verwendete Software und das digitale Höhenmodell thematisiert. Das Untersuchungsgebiet, der Referenzdatensatz und die verwendete Methodik zur Evaluierung sowie für den Vergleich der Algorithmen werden im Kapitel 4 erläutert. Im Kapitel 5 werden die Resultate der Evaluierung und des Algorithmenvergleichs präsentiert. Anschliessend werden die Resultate dieser Arbeit in Kapitel 6 anhand der Fragestellungen diskutiert. Eine Schlussfolgerung zu den gewonnenen Erkenntnissen und einen Ausblick auf zukünftige Forschungsschwerpunkte in Kapitel 7 bilden den Abschluss dieser Arbeit. Die nachfolgende Abb. 1.1 gibt einen Überblick über die thematischen Zusammenhänge dieser Arbeit.



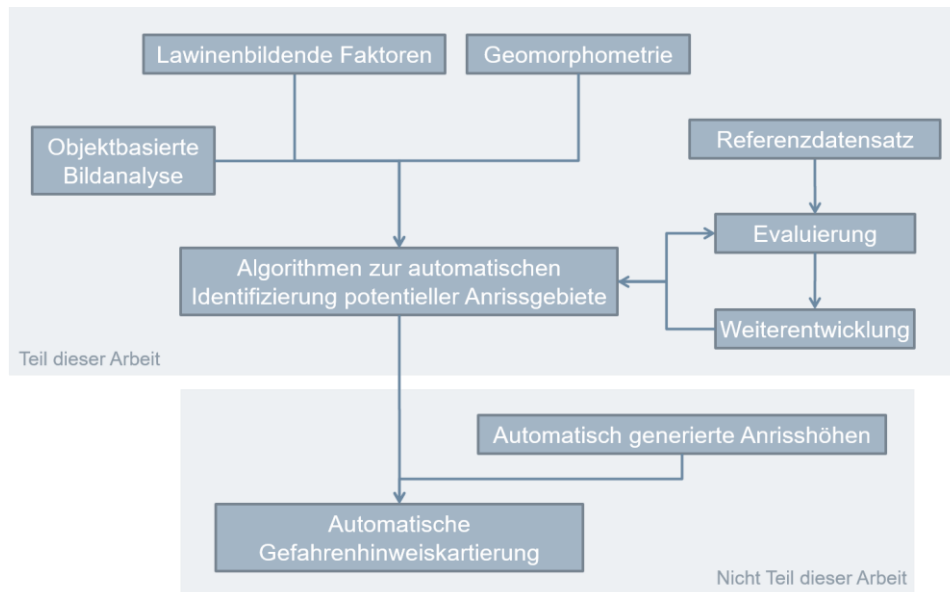


Abb. 1.1: Themenüberblick und Zusammenhänge dieser Arbeit

## 2 Allgemeine Grundlagen

### 2.1 Lawinen

Der Prozess Lawine gehört zu den gravitativen Massenbewegungen (Kienholz et al. 1998). Lawinen sind das zentrale Thema dieser Arbeit. Deswegen wird im Folgendem einen Überblick über den Prozess Lawine gegeben.

#### 2.1.1 Lawinenklassifikation

Je nach Unterscheidungskriterium können Lawinen in verschiedene Lawinenarten eingeteilt werden (Abb. 2.1). Gemäss Rudolf-Miklau et al. (2011) wurde die erste detaillierte Lawinenklassifikation von der UNESCO (1981) publiziert.

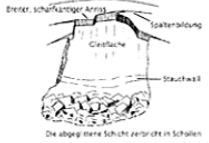

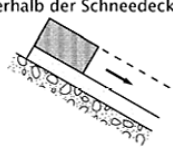
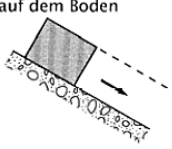


Lawinenklassifikation und Begriffsbestimmung		
äusseres Merkmal	Unterscheidung und Namensgebung	
Form des Anrisses	linienförmig, scharfkantig, senkrecht zur Gleitfläche  ⇒ <b>Schneebrettlawine</b>	punktförmig  ⇒ <b>Lockerschneelawine</b>
Lage der Gleitfläche	innerhalb der Schneedecke  ⇒ <b>Oberlawine</b>	auf dem Boden  ⇒ <b>Bodenlawine</b>
Form der Bewegung	vorwiegend stiebend  ⇒ <b>Staublawine</b>	vorwiegend fließend  ⇒ <b>Fließlawine</b>
Feuchtigkeit des abgleitenden Schnees	trocken ⇒ <b>Trockenschneelawine</b>	nass ⇒ <b>Nassschneelawine</b>
Form der Bahn (Querprofil)	flächige Bahn ⇒ <b>Flächenlawine</b>	runsenförmige Bahn ⇒ <b>Runsenlawine</b>
Länge der Bahn	vom Berg ins Tal ⇒ <b>Tallawine</b>	am Hangfuss zum Stillstand kommend ⇒ <b>Hanglawine</b>
Art des Schadens	Heimstätte, Hab und Gut, Verkehr, Wald ⇒ <b>Katastrophen- oder Schadenlawine</b>	Skifahrer und Bergsteiger im freien Skigelände ⇒ <b>Touristen- oder Skifahrerlawine</b>
Art des anbrechenden Materials	Schnee ⇒ <b>Schneelawine</b>	(Gletscher-) Eis ⇒ <b>Eislawine (Gletscherabbruch)</b>

Abb. 2.1: Lawinenklassifikation analog zu jener der UNESCO (1981) aus Kienholz et al. (1998)

In dieser Arbeit wird die Form des Anrisses (äusseres Merkmal) als Unterscheidungskriterium verwendet. Dementsprechend wird zwischen Schneebrett- und Lockerschneelawine unterschieden. Hierbei ist zu erwähnen, dass beispielsweise die Feuchtigkeit des Schnees bei beiden Lawinenarten unterschiedlich hoch sein kann. Somit gibt es beispielsweise trockene als auch nasse Schneebrett- und Lockerschneelawinen. In Abb. 2.2 sind diese zwei Fälle am Beispiel der Lockerschneelawine ersichtlich. (Harvey et al. 2013).

### Lockerschneelawine

Bei dieser Lawinenart (Abb. 2.2) gerät Schnee mit nur schwacher Bindung bei Hangneigungen von allgemein über  $35^\circ$  punktuell in Bewegung (Kienholz et al. 1998). „Der sich bewegende Schnee bringt fortlaufend weiteren Schnee in Bewegung. Die Lockerschneelawine wird dadurch immer grösser und grösser. Sie ist im Vergleich zur Schneebrettlawine langsam. Damit sie sich bildet, braucht es unverfestigten Schnee an der Oberfläche mit sehr schwachen Bindungen. Typischerweise treten Lockerschneelawinen nach Neuschneefällen oder bei oberflächlicher Anfeuchtung des Schnees auf“ (Harvey et al. 2013: 39). Im Gegensatz zur Schneebrettlawine steht die Initiierung einer Lockerschneelawine nicht in Zusammenhang mit Scher- und Druckbelastungen (Kienholz et al. 1998, Harvey et al. 2013).

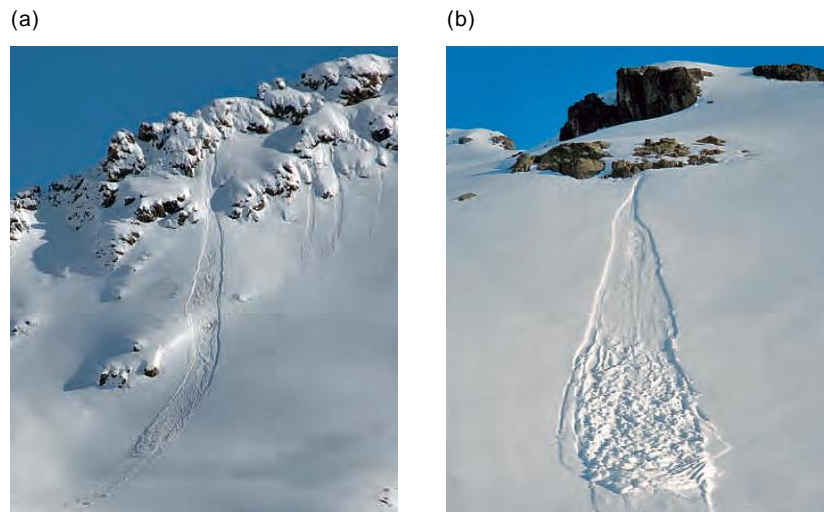


Abb. 2.2: Trockene (a) und nasse (b) Lockerschneelawine (Harvey et al. 2013)

### Schneebrettlawine

Dies ist eine Lawinenart, bei welcher eine gebundene Schneeschicht, ein sogenanntes Schneebrett, als Folge eines Kollapses einer darunterliegenden Schwachsicht abgleitet. Die Schwachsicht eines Schneebretts wird in Kapitel 2.1.2 detaillierter thematisiert. Die Schneebrettlawine kann räumlich in ein Anrissgebiet, eine Sturzbahn und ein Ablagerungsgebiet aufgeteilt werden (Abb. 2.3). Der Anriss ist durch eine scharfe Anrisskante gekennzeichnet (Kienholz et al. 1998, Harvey et al. 2013). „Der Bruch ist nicht –wie bisher meist angenommen – ein einfacher Scherbruch, sondern beruht auf einer seltenen Art des Versagens, welches in der Fachliteratur als „Antiriss“ bezeichnet wird. Die Schwachsicht kollabiert aufgrund der kombinierten Scher- und Druckbelastung“ (Harvey et al. 2013: 32).

Im Fokus dieser Masterarbeit steht die Schneebrettlawine. Diese Lawinenart ist die gefährlichste und fordert über 90% der Lawinenopfer (SLF 2018b). Die Identifizierung potentieller Anrissgebiete bezieht sich in der Folge ausschliesslich auf die Schneebrettlawine. Wenn im weiteren Verlauf der Begriff Lawine verwendet wird, ist damit immer eine Schneebrettlawine gemeint.



Abb. 2.3: Unterteilung einer Schneebrettlawine in Teilbereiche (Harvey et al. 2013)

### 2.1.2 Lawinenbildende Faktoren der Schneebrettlawine

Die Entstehung einer Schneebrettlawine basiert auf einer komplexen Wechselwirkung zwischen Gelände, Schneedecke und Wetterverhältnissen (Schweizer et al. 2003). In diesem Kapitel werden diese lawinenbildenden Faktoren genauer erklärt.

#### Gelände

Im Gegensatz zur Schneedecke und den Wetterverhältnissen ist das Gelände der einzige lawinenbildende Faktor, der über die Zeit als konstant angesehen werden kann. Einerseits gibt es geländespezifische Eigenschaften, welche zwingend für eine Schneebrettauslösung gegeben sein müssen, andererseits beeinflusst das Gelände auch indirekt die Faktoren Schneedecke und Wetter. Letzteres zeigt sich beispielsweise bei der Beeinflussung der Schneeverfrachtung sowie -ablagerung, der Sonneneinstrahlung und durch lokale Temperaturunterschiede (Gleason 1994, Statham et al. 2017).

#### Hangneigung

Die Hangneigung ist eine geländespezifische Eigenschaft, welche die Bildung einer Schneebrettlawine direkt bedingt. Bei rund 30° Hangneigung liegt die kritische Neigung, ob ein Schneebrett nach dem Bruch der Schwachschicht abrutscht oder nicht (Voellmy 1955, Harvey et al. 2013). Im extrem

steilen Gelände ( $>60^\circ$ ) wird der Schnee kontinuierlich durch Lockerschneelawinen und den Wind abgetragen (Wirz et al. 2011, Phillips et al. 2017). In diesem Gelände werden selten Schneebrettlawinen beobachtet (McClung und Schaerer 1993, Phillips et al. 2017). Deshalb weisen Anrissgebiete von Schneebrettlawinen in der Regel Hangneigungen von rund  $28^\circ$  bis  $50^\circ$ , maximal  $60^\circ$  auf (Schweizer et al. 2003, Nairz et al. 2011).

### Exposition

Die Neigung und Exposition des Geländes beeinflusst die Energiemenge, welche eine Einheitsfläche (beispielsweise ein Quadratmeter) durch die Sonneneinstrahlung erhält. Zur Sonne hin geneigte Hänge sind mehr solarer Strahlung pro Einheitsfläche ausgesetzt als ebenes Gelände. Das gleiche gilt für südexponiertes Gelände verglichen mit nordexponiertem. Die Strahlung beeinflusst die Schneedecke und damit unter anderem auch die Auslösung von Schneebrettlawinen. Die Rolle der Strahlung im Zusammenhang mit Lawinen wird im Folgendem noch näher besprochen (Khavrus und Shelevytsky 2012, Harvey et al. 2013).

### Kurvatur

Die Geländeform hat ebenfalls einen Einfluss auf die Schneebrettauslösung. Die Auswertung von kartierten Schneebrettlawinen hat ergeben, dass Anrissgebiete häufig eine leicht konkave Profil- und Plankurvatur aufweisen. So befanden sich die untersuchten Anrisskanten nicht direkt auf einer Kuppe oder einem Grat, sondern in geringem Abstand unterhalb davon (Vontobel 2011). In stark ausgeprägten Gräben und Runsen wird der Schnee typischerweise in grossen Mengen abgelagert und bleibt ungestört liegen, wodurch dort mächtige Schneeschichten entstehen. Eine Bruchinitiierung ist bei einem mächtigen Schneebrett weniger wahrscheinlich, da sehr grosse Kräfte nötig sind, um einen Bruch in einer tiefliegenden Schwachschicht zu initiieren. Zudem werden tieferliegende Schwachschichten aufgrund der hohen Überlast tendenziell stabiler (Nairz et al. 2011, Harvey et al. 2013).

### Geländerauigkeit

Eine weitere Eigenschaft des Geländes ist dessen Rauigkeit: Eine hohe Geländerauigkeit kann Ursache für eine hohe Variabilität in der Stratigraphie von dünnen Schneedecken sein und den Aufbau einer kontinuierlichen, flächig ausgedehnten Schwachschicht verhindern (Schweizer et al. 2003, Phillips et al. 2017). Dies vermindert die Bruchfortpflanzung (Harvey et al. 2013). Dabei führt felsiges Gelände mit hoher Geländerauigkeit, aufgrund von grösseren Temperaturgradienten rund um Felsblöcke, aber auch zur Bildung von Becherkristallen, welche als Schwachschicht dienen können (Schweizer et al. 2003). Sollte es zum Abgang einer Schneebrettlawine kommen, kann eine raue Geländeoberfläche jedoch auch das Abgleiten einzelner Schollen eines Schneebretts hemmen (van Herwijnen und Heierli 2009).

### Vegetation

Im Zusammenhang mit Schneebrettlawinen ist die vorherrschende Vegetation, vor allem die Existenz von Wald, von grosser Bedeutung. Ein genügend dichter Wald wirkt stabilisierend auf die Schneedecke und kann Lawinenanbrüche verhindern. Die stabilisierende Wirkung beruht auf ver-

schiedenen Eigenarten des Waldes, welche die Schneeeinterzeption, den Strahlungshaushalt, die Wind- sowie Stützwirkung der Stämme betreffen (Margreth 2004, Schneebeli und Bebi 2004).

### **Schneedecke**

Die Schneedeckenstratigraphie wird als wichtiger lawinenbildender Faktor angesehen. Eine Schneebrettlawine setzt sowohl ein gebundenes Brett als auch eine darunterliegende Schwachschicht voraus. Eine Zusatzlast, beispielsweise durch Neu- oder Triebsschnee oder durch eine Temperaturerhöhung, löst keine Lawine aus, solange keine Schwachschicht unterhalb eines Schneebretts vorhanden ist. Besteht die Schneedecke komplett aus kohäsionslosen Kristallen ist auch keine Auslösung einer Schneebrettlawine möglich, weil keine gebundene Schicht für ein Brett existiert (Schweizer et al. 2003, Harvey et al. 2013).

### **Schwachschicht**

Eine Schwachschicht in einer Schneedecke ist weich, bruchanfällig und kollabierfähig. Das heisst, sie fällt leicht in einzelne Teile auseinander, und die einzelnen Eiskörner können nach einem Bruch durch eine Zusatzlast dichter gepackt werden. Dies ist jedoch nur möglich, wenn die Schwachschicht sich im Wirkungsbereich der Zusatzlast, beispielsweise Neu- oder Triebsschnee, befindet. Die Schwachschicht, welche initial kollabiert, befindet sich demnach meist in den oberen Schichten der Schneedecke. Bodennahe Schwachschichten sind nur an wenigen schneearmen Stellen auslösbar (Harvey et al. 2013).

### **Schneebrett**

Sowohl weicher, gebundener Schnee als auch dichter, harter Schnee können als Schneebrett fungieren. Die Eigenschaften und Masse eines Schneebrettes beeinflussen die Bruchinitierung sowie -fortpflanzung. So wirkt beispielsweise eine Zusatzbelastung bei warmem Schnee mehr in die Tiefe als bei kaltem Schnee. Ein Kollaps einer allfällig darunterliegenden Schwachschicht wird dadurch wahrscheinlicher. Auch die Bruchausbreitung wird durch eine höhere Verformbarkeit eines Bretts tendenziell begünstigt. Ein deformierbares, elastisches Brett fördert im Gegensatz zu einem spröden, brüchigen Brett die Bruchfortpflanzung. Je mächtiger das Schneebrett, desto mehr Energie liegt nach der Bruchinitierung für dessen Ausbreitung vor. Dadurch können auch stabilere Bereiche einer Schwachschicht brechen, wodurch sich der Bruch wiederum weiter fortpflanzen kann (Schweizer et al. 2003, Harvey et al. 2013).

### **Wetter**

Der Faktor Wetter kann durch die wesentlichen Punkte Niederschlag, Wind, Schneedeckentemperatur und Strahlung zusammengefasst werden.

### **Niederschlag: Neuschnee**

In der Regel meint man mit Neuschnee denjenigen Schnee, der in den letzten ein bis drei Tagen gefallen ist. Abhängig von der Temperatur verändert sich Neuschnee relativ schnell in unterschiedlicher Art und Weise. Der Neuschnee wirkt als Zusatzlast auf die darunterliegenden Schneeschichten und bildet gleichzeitig die neue, oberste Schicht der bestehenden Schneedecke. Abhängig von den

vorherrschenden Bedingungen kann es sein, dass nach einer Neuschneeperiode sich auf dem Altschnee gebundener Neuschnee mit einer darunterliegenden Schwachschicht entwickelt. Bei genügend grosser Zusatzlast kann infolgedessen die gebundene Neuschneesicht als Schneebrettlawine abgleiten. Neuschnee kann aber auch, je nach Beschaffenheit und Verbindungen der Schneesichten, bedingt durch ihre Zusatzlast, eine Lawinenauslösung in einer tieferen, im Altschnee liegenden Schwachschicht verursachen (Schweizer et al. 2003, Harvey et al. 2013).

### Niederschlag: Regen

Durch Regen dringt Wasser in die oberflächennahen Schneesichten ein. Mit zunehmendem Wassergehalt entsteht freies Wasser, welches der Schwerkraft folgend innerhalb der Schneedecke nach unten sickert und sich unter Umständen an markanten Schichtgrenzen staut. Dort kann es durch den lokal höheren Wassergehalt zu einem Festigkeitsverlust und letztlich zu einer Lawinenauslösung kommen. Daneben wirkt die Anfeuchtung des Schnees durch den Regen als Zusatzlast auf die Schneedecke (Harvey et al. 2013).

### Wind

Der Wind transportiert und lagert Schnee um und wirkt damit auf die Schneedecke ein. Durch die Verfrachtung entsteht sogenannter Trieb Schnee. Hierbei ist der Einfluss des Windes abhängig von den Eigenschaften des oberflächennahen Schnees und des Geländes. Trieb Schnee entsteht, wenn es während eines Schneefalls windet oder wenn der Wind den Schnee einer lockeren Schneeoberfläche umlagert. Dabei wird entweder Neuschnee oder lockerer Altschnee verfrachtet. Die lokale Topografie bestimmt den Wind in Bodennähe. Dies bedeutet, dass die Windrichtung in Bodennähe kleinräumig stark variiert und sich oft von der Hauptwindrichtung unterscheidet. In den oberen Höhenlagen sowie Pass- und Kammlagen ist der Wind allgemein kräftiger als unterhalb der Waldgrenze. An den vom Wind abgewandten, Lee-Seiten (beispielsweise hinter Geländekanten) akkumuliert sich der Trieb Schnee. Auf der dem Wind zugewandten, Luv-Seite wird der Schnee erodiert. Durch die Abtragung des Schnees bleibt eine raue, harte und kleinräumig sehr unregelmässige Schneeoberfläche zurück. Solche Oberflächen sind im Hinblick auf die Lawinengefahr vorteilhaft, da sich dank ihrer kleinräumigen Variabilität kaum eine flächig ausgedehnte Schwachschicht bilden kann. Offene, gleichmässige Hänge bieten grosse Angriffsflächen für hangparallele Winde und begünstigen die Entstehung von Trieb Schnee. Neben der Schneeverfrachtung kann der Wind auch warme oder kalte Luft zuführen und dadurch die Energiebilanz an der Schneeoberfläche beeinflussen (Schweizer et al. 2003, Harvey et al. 2013).

### Schneedeckentemperatur

Entscheidend für eine Veränderung der Schneedeckentemperatur ist deren Energiebilanz. Die Energiebilanz umfasst alle beteiligten Energieflüsse und hängt unter anderem von den Ein- und Ausstrahlungsbedingungen sowie vom Wind ab. Im Falle einer positiven Energiebilanz erwärmt sich die Oberfläche der Schneedecke, ansonsten kühlt sie sich ab (Harvey et al. 2013). Änderungen der Schneeoberflächentemperatur wirken sich im Allgemeinen nur auf die oberen Schichten aus, da der Schnee eine geringe Wärmeleitfähigkeit aufweist. Detaillierte Temperaturmessungen in der Schneedecke haben ergeben, dass sich tageszeitliche Variationen primär auf die oberen 10-20 cm

auswirken. (Schweizer et al. 2003). Aufgrund der relativ tiefen Wärmeleitfähigkeit von Schnee wirkt sich eine Änderung der Schneeoberflächentemperatur verzögert und abgeschwächt auf die darunterliegenden Schneeschichten aus. Je nachdem wird eine tieferliegende Schwachschicht durch die Änderung der Schneeoberflächentemperatur gar nicht beeinflusst. Weil sich jedoch die Eigenschaften der oberen Schneeschichten durch den Temperaturanstieg verändert haben, kann sich die Lawinengefahr trotzdem erhöhen. Dies gilt jedoch nur, solange sich die Schneetemperaturen deutlich unter  $0^{\circ}$  bewegen. Andernfalls beginnt der Schnee zu schmelzen und durchfeuchtet die Schneedecke. Dies führt zu denselben Prozessen, welche Regen in der Schneedecke auslösen (Schweizer et al. 2003, Harvey et al. 2013).

### Strahlung

Die weisse Schneedecke reflektiert zu einem grossen Teil die kurzwellige Sonneneinstrahlung. Ein kleiner Teil der Strahlung dringt in die oberflächennahen Schichten ein und wird dort absorbiert. Gleichzeitig gibt die Schneedecke langwellige Strahlung ab. Bei klarem Himmel trifft während des Tages die kurzwellige, solare Strahlung ungehindert auf die Schneedecke. Zugleich kann die Schneedecke bei klarem Himmel ungehindert im langwelligen Bereich abstrahlen. Bei Wolken und bei Nebel ändert sich die Strahlungsbilanz (Nairz et al. 2011, Harvey et al. 2013): Die direkte solare Einstrahlung wird teilweise gestreut und erreicht abgeschwächt als diffuse Strahlung die Erdoberfläche (Khavrus und Shelevytsky 2012). Die langwellige Abstrahlung der Schneedecke wird von Wolken und Nebel in der Atmosphäre absorbiert und wieder emittiert. Ein Teil davon gelangt wieder auf die Erdoberfläche und wird durch den Schnee aufgenommen (Harvey et al. 2013).

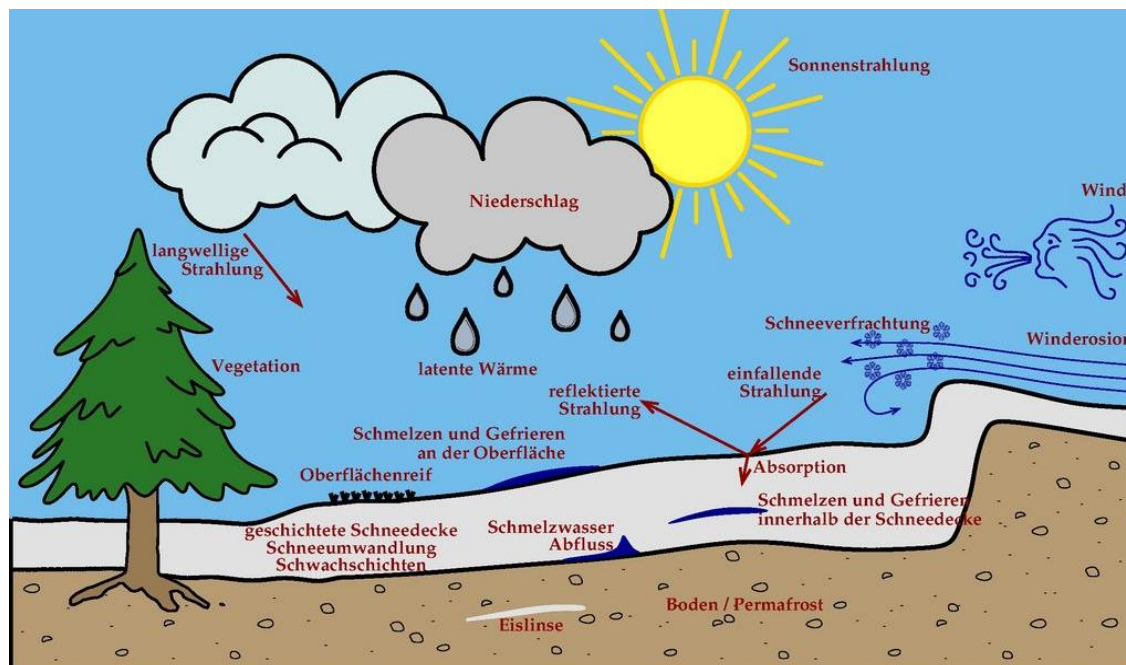


Abb. 2.4: Schematische Darstellung von Prozessen, die sich auf die lawinenbildenden Faktoren Schneedecke und Wetter beziehen (SLF 2018a)



## 2.2 Gefahrenhinweis- und Gefahrenkarte

Der Umgang mit Naturgefahren verlangt nach raumplanerischen Mitteln (Camenzind-Wildi et al. 2000). Dafür existieren in der Schweiz Gefahrenhinweis- und Gefahrenkarten. Tab. 2.1 gibt eine Übersicht über die Eigenschaften der beiden Kartenarten.

Tab. 2.1: Übersicht über die Eigenschaften von Gefahrenhinweiskarten und Gefahrenkarten (Camenzind-Wildi et al. 2000)

	Gefahrenhinweiskarten	Gefahrenkarten
<b>Inhalt</b>	<ul style="list-style-type: none"> <li>Gefahr vorhanden/nicht vorhanden</li> <li>Gefahrenart</li> </ul>	<ul style="list-style-type: none"> <li>Genauer Lokalisierung der Gefahrengebiete</li> <li>Gefahrenart</li> <li>5 Gefahrenstufen (abgestuft nach Intensität und Wahrscheinlichkeit)</li> </ul>
<b>Zweck</b>	<ul style="list-style-type: none"> <li>Grundlage für die Richtplanung</li> <li>Erkennung von Konfliktgebieten</li> <li>Grundlage für die Beurteilung von Baugesuchen ausserhalb der Bauzone</li> <li>Prioritätensetzung für die Gefahrenkartierung</li> </ul>	<ul style="list-style-type: none"> <li>Grundlage für die Ausscheidung von Gefahrenzonen in der Nutzungsplanung</li> <li>Grundlage für die Formulierung von Bauauflagen</li> <li>Projektierung von Schutzmassnahmen</li> <li>Notfallplanung</li> </ul>
<b>Bearbeitungstiefe</b>	Gering, grobe Übersicht, tiefe Abgrenzungsgenauigkeit	Hoch, detaillierte Übersicht, hohe Abgrenzungsgenauigkeit (parzellengenau)
<b>Massstab</b>	1:10000 bis 1:50000	1:2000 bis 1:10000
<b>Perimeter</b>	Kanton (flächendeckend)	Region/Gemeinde (nicht flächendeckend)
<b>Überprüfung</b>	Periodisch, zum Beispiel im Rahmen der Richtplanrevision	Periodisch im Rahmen der Nutzungsplanrevision oder bei erheblich veränderter Gefahrensituation (zum Beispiel infolge von Schutzmassnahmen oder einem Ereignis)
<b>Produkte</b>	Karte und technischer Bericht	Karte und technischer Bericht

### 2.2.1 Gefahrenhinweiskarte

Die Gefahrenhinweiskarte liefert eine grobe Übersicht über potentielle Gefahrengebiete, beispielhaft dargestellt in Abb. 2.5. Die Erstellung einer Gefahrenhinweiskarte basiert auf Modellrechnungen und wird mit vorhandenen Ereigniskatastern plausibilisiert. Bezüglich der räumlichen Abgrenzung der Prozessräume kann sie Ungenauigkeiten enthalten und die Gefährdung nicht in jedem Fall genau wiedergeben. Die Gefahrenhinweiskarte zeigt für grosse Gebiete – meist für einen Kanton – flächendeckend auf, wo mit welchen Gefahren gerechnet werden muss; dies jedoch ohne Angabe der Gefahrenstufe. Je nach dem vereint die Karte die Prozessräume von verschiedenen Naturgefahren. Daraus lassen sich mit geringem Aufwand mögliche Konfliktstellen zwischen Gefahren- und Schadenpotential ableiten. Die Gefahrenhinweiskarte dient der Richtplanung, der Prü-

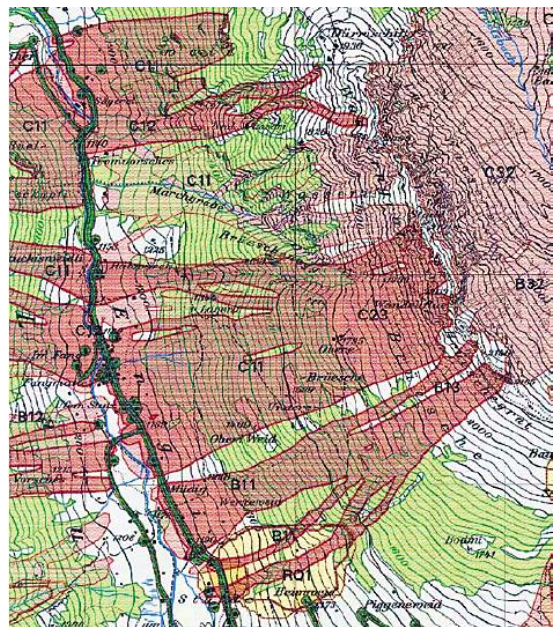


Abb. 2.5: Gefahrenhinweiskarte, potentielle Gefahrengebiete in rot (Camenzind-Wildi et al. 2000)

fung von Baugesuchen ausserhalb der Bauzonen sowie der Prioritätensetzung bei der Ausarbeitung von Gefahrenkarten (Camenzind-Wildi et al. 2000).

### 2.2.2 Gefahrenkarte

Die Gefahrenkarte gibt eine detaillierte Übersicht über die Gefährdungssituation, exemplarisch abgebildet in Abb. 2.7. Gefahrenkarten und die zugehörigen technischen Berichte enthalten detaillierte Angaben über Prozessursache und -ablauf, räumliche Ausdehnung, Intensität, sowie Eintrittswahrscheinlichkeit von Naturgefahren. Das erste Zwischenprodukt bei der Ausarbeitung der Gefahrenkarte sind die Intensitätskarten. Sie zeigen für vordefinierte Eintrittswahrscheinlichkeiten die zu erwartenden Intensitäten der Gefahrenprozesse. Daraus lassen sich eintrittswahrscheinlichkeits-spezifische und anhand derer eine finale Gefahrenkarte für ein Naturgefahrenprozess ableiten (Camenzind-Wildi et al. 2000). Durch die Verknüpfung von Intensität und Wahrscheinlichkeit können in einer Gefahrenkarte im Allgemeinen die folgenden Gefahrenstufen auftreten (Abb. 2.6):

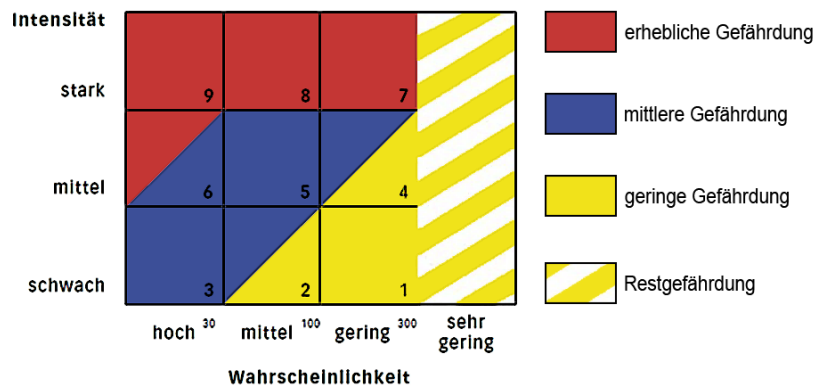


Abb. 2.6: Gefahrenstufen als Resultat von Intensität und Wahrscheinlichkeit (Camenzind-Wildi et al. 2000)



Abb. 2.7: Gefahrenkarte (Camenzind-Wildi et al. 2000)

Integrale Gefahrenkarten umfassen alle vorkommenden Naturgefahrenprozesse in einem betrachteten Gebiet. Sie dienen, im spezifischen deren Gefahrenstufen, als Grundlage für die Ausscheidung von Gefahrenzonen und für die Festlegung der Zonennutzungsauflagen in der Raumplanung (Camenzind-Wildi et al. 2000).

## 2.3 Geomorphometrie

Die Geomorphometrie ist die Wissenschaft der quantitativen Beschreibung und Analyse der Erdoberfläche. Diese interdisziplinäre Wissenschaft hat sich aus der Mathematik, den Geowissenschaften und zuletzt aufgrund der Informatik kontinuierlich weiterentwickelt (Abb. 2.8). Nachdem die Geomorphometrie lange nur als Methodik innerhalb der Geographie, Geomorphologie und Bodenkunde galt, ist sie mittlerweile eine eigene Forschungsdisziplin (Pike et al. 2009).

Die grundlegende Operation in der Geomorphometrie ist die Extraktion von Geländeparametern und -objekten aus einem digitalen Modell der Erdoberfläche. Ein Geländeparameter ist ein beschreibendes Mass für die Oberflächenform (beispielsweise die Hangneigung oder Krümmung). Jedem Punkt im Gelände kann ein Parameterwert zugewiesen werden. Dieses kontinuierliche Wertefeld wird meist in einem Rasterdatensatz repräsentiert. Ein Geländeobjekt ist ein räumlich diskretes Merkmal (beispielsweise eine Rinne oder Mulde), häufig dargestellt als Punkt, Linie oder Polygon in einem Vektordatensatz (Pike et al. 2009).

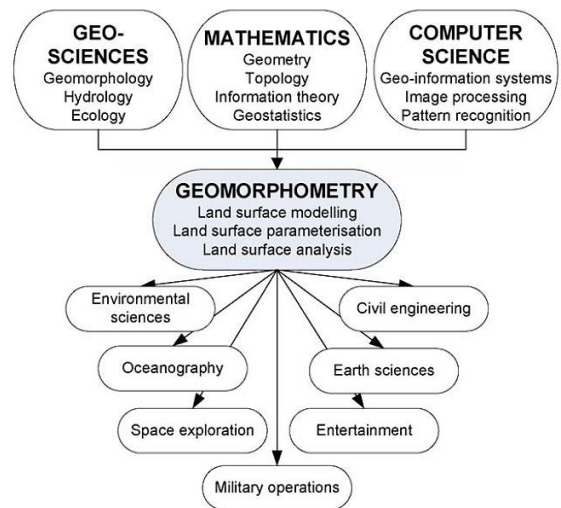


Abb. 2.8: Geomorphometrie im Kontext ihrer Ursprünge und ihrer Anwendungen (Pike et al. 2009)

### 2.3.1 Digitale Modelle der Erdoberfläche

Digitale Modelle der Erdoberfläche zeichnen sich aus durch ihre quasi kontinuierliche Beschreibung grosser Gebiete zu relativ geringen Kosten (Hengl und MacMillan 2009). Im Zug der fortschreitenden technologischen Entwicklungen in der Fernerkundung verbessert sich die räumliche Auflösung der digitalen Erdoberflächenmodelle fortlaufend. Gerade in alpinen Regionen sind hochaufgelöste, digitale Modelle ( $\leq 10\text{m}/\text{Pixel}$ ) nötig, um kleinräumig stark ändernde Geländeparameter abbilden zu können (Hengl und Evans 2009, Pike et al. 2009). Solch hochaufgelöste digitale Modelle, die mit Hilfe von modernen Fernerkundungsmethoden generiert werden können, sind zunehmend auch für alpine Regionen verfügbar (Bühler et al. 2012, Fonstad et al. 2013).

Es existieren verschiedene Begriffe für die verschiedenen digitalen Modelle von topographischen Daten. Das generische digitale Höhenmodell (DHM) impliziert normalerweise x-, y-Koordinaten und die dazugehörigen Höhenwerte der Erdoberfläche, jedoch ohne Vegetation und künstliche, menschgemachte Objekte. Das digitale Oberflächenmodell (DOM) besteht ebenso aus x-, y-Koordinaten und dazugehörigen Höhenwerten. Die Höhenwerte beziehen sich aber auf die Oberfläche von Objekten, beispielsweise von Gebäuden oder Vegetation, über der Erdoberfläche (Abb. 2.9). Des Weiteren wird der Begriff digitales Geländemodell (DGM) verwendet. Vielfach ist das DGM gleichbeutend mit dem DHM. In dieser Arbeit wird das digitale Geländemodell (DGM) auch als Synonym für das digitale Höhenmodell (DHM) gebraucht. Manchmal jedoch, beispielsweise in den Vereinigten Staaten von Amerika, ist ein DGM ein Vektordatensatz aus unregelmässig angeordneten Punkten mit Höhenwerten als Attribut, ähnlich einer Punktwolke. In diesem Fall ist ein DGM kein regelmässiges Gitternetz, sondern ein diskreter und nicht kontinuierlicher Punktdatensatz (Heidemann 2018).

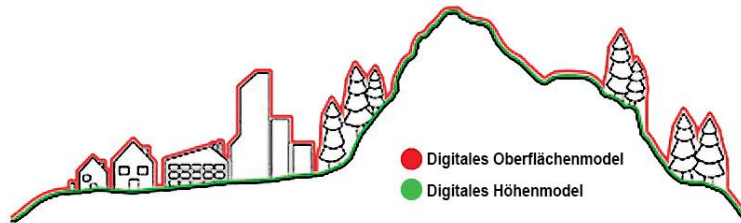


Abb. 2.9: Schematische Darstellung der Werte abgespeichert in einem digitalen Oberflächen- und in einem digitalen Höhenmodell (Asharyanto et al. 2015)

Die Erdoberfläche hat eine kontinuierliche räumliche Existenz. Das impliziert, dass für jeden beliebigen Punkt der Oberfläche ein Wert bestimmbar ist. Dieses Raumkonzept der Erdoberfläche als kontinuierliches Feld wird bei der Umwandlung in ein digitales Modell zwangsläufig abstrahiert, da es in eine bestimmte Datenstruktur eingebettet werden muss (Burrough et al. 2015). Im Allgemeinen werden digitale Modelle der Erdoberfläche in den folgenden zwei Datenstrukturen, Raster- und Vektormodell, gespeichert (Abb. 2.11).

### Rastermodell

Im Rastermodell werden Höhenwerte in einem regelmässigen Gitternetz gespeichert. Das Gitternetz setzt sich aus einzelnen quadratischen Pixeln zusammen, die alle die gleiche Grösse aufweisen. Jedem Pixel wird dabei der entsprechende Höhenwert zugewiesen. Neben der Zellgrösse benötigt man ausserdem die Koordinaten von mindestens einem Gitterpunkt (in der Regel die linke untere Ecke des gesamten Gitternetzes) und die Anzahl der Zeilen und Spalten. Durch diese fünf Parameter ist ein Rastermodell definier- und georeferenzierbar (Abb. 2.10). Obwohl gerasterte Oberflächenmodelle gleichmässig und kontinuierlich scheinen, sind sie trotzdem eine diskrete Repräsentation der Erdoberfläche, denn der betrachtete Raum wird in Zellen unterteilt. Die Zellgrösse bestimmt dabei die räumliche Auflösung des Rastermodells und die Genauigkeit, mit welcher die tatsächliche Oberfläche repräsentiert wird (Hengl und Evans 2009, Pike et al. 2009).

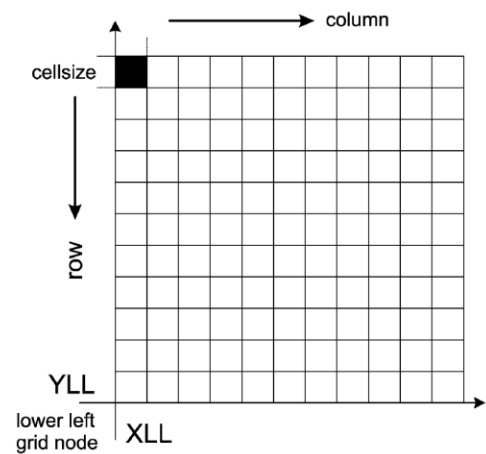


Abb. 2.10: Rastermodell, definiert durch fünf Parameter (Pike et al. 2009)

### Vektormodell

Als Grundlage für Vektormodelle der Erdoberfläche dienen meistens unregelmässige Punktdatensätze. Den einzelnen Punkten ist ein Höhenwert als Attribut hinterlegt. Gebiete mit grossen Höhenunterschieden sowie rauen Oberflächen verlangen nach kleineren Punktabständen und für Gebiete mit geringfügigen Höhenunterschieden sowie gleichmässigen Oberflächen reichen grössere Punktabstände. Die Punktdichte kann also je nach Oberflächenbeschaffenheit angepasst werden. Unregelmässige Punktdatensätze können durch Triangulation in Oberflächenmodelle umgewandelt werden. Hierbei werden Dreiecke zwischen den einzelnen Punkten berechnet. Solch ein unregelmässiges Dreiecksnetz wird *Triangular Irregular Network* (TIN) genannt (Hengl und Evans 2009).

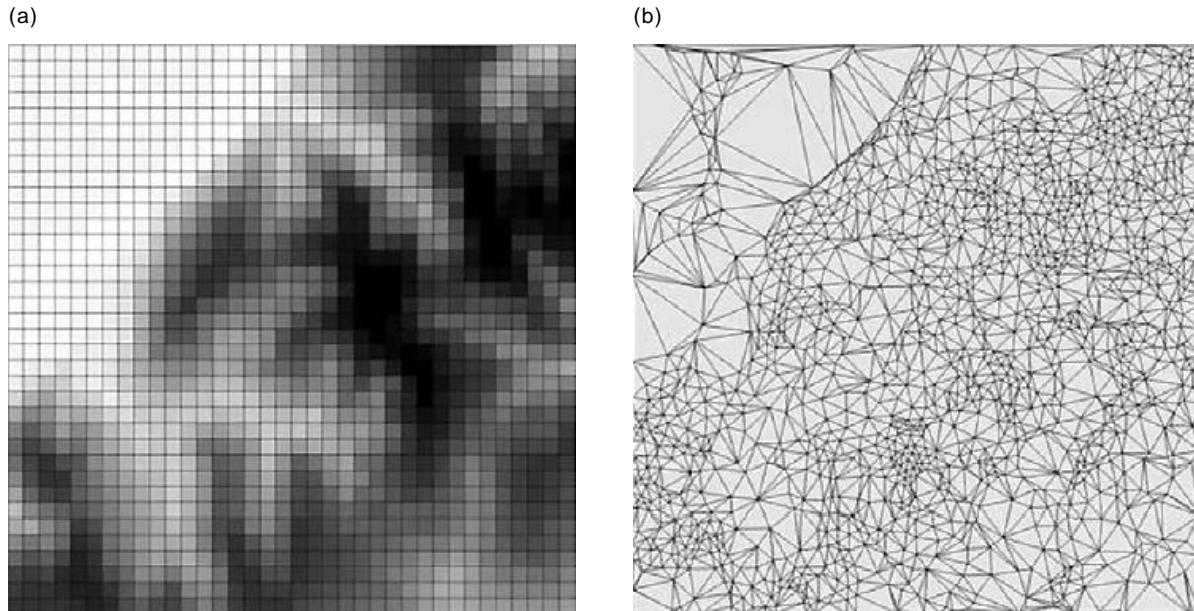


Abb. 2.11: Vergleich verschiedener digitaler Modelle: Rastermodell (a) und Vektormodell als TIN (b) (Hengl und Evans 2009)

### 2.3.2 Elementare Geländeparameter

Wie in Kapitel 2.3 erwähnt, können durch die Geomorphometrie Geländeparameter aus einem DHM abgeleitet werden. Das Gelände kann mathematisch betrachtet als kontinuierliche, stetige und darum differenzierbare Oberfläche aufgefasst werden. Daher ist es im Prinzip möglich, für jeden beliebigen Punkt auf der Oberfläche dessen Ableitung zu berechnen. Durch Übertragung in ein Raster- oder Vektormodell wird die Oberfläche jedoch diskretisiert. Im Nachfolgenden wird ausschliesslich

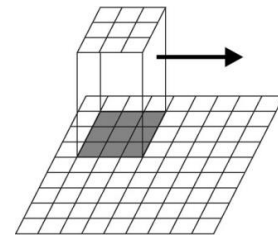


Abb. 2.12: Gleitendes Zellfenster (Olaya 2009)

das Rastermodell für die Darstellung von Höhenwerten verwendet, weshalb auch die Herleitung der Geländeparameter ausschliesslich für ein Rastermodell beschrieben wird. Die Ableitungen für jede Zelle eines Rastermodells werden jeweils lokal innerhalb eines bestimmten Zellfensters abgeschätzt. Die nachfolgenden Erklärungen zu den Geländeparametern gelten alle für ein 3x3 pixelgrosses Zellfenster (Abb. 2.12). Der errechnete Geländeparameter bezieht sich dabei auf das Zentralpixel des jeweiligen Zellfensters. Um ein Geländeparameter für jede Zelle eines Rastermodells zu bestimmen, wird das Zellfenster systematisch über das Gitternetz bewegt. Für die mathematischen Herleitungen der Hangneigung, Exposition und Krümmung gelten die Bezeichnungen, welche in Abb. 2.13 verwendet werden (Olaya 2009, Burrough et al. 2015).

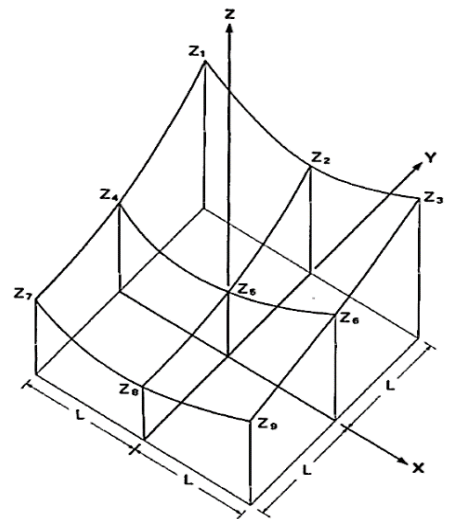


Abb. 2.13: Rasteroberfläche in einem Koordinatensystem (Burrough et al. 2015)

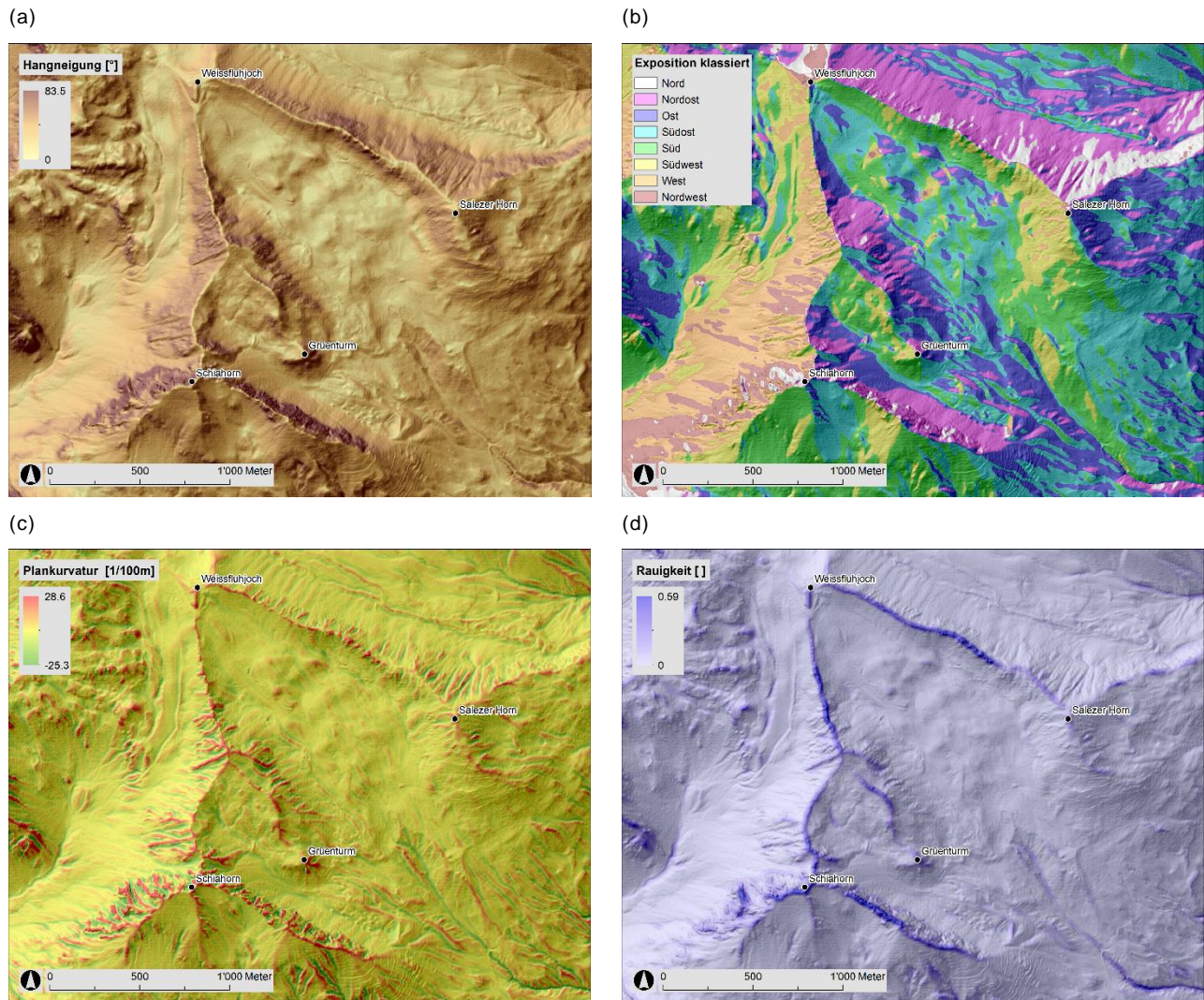


Abb. 2.14: Geländeparameter exemplarisch dargestellt für die Region Weissfluhjoch: Hangneigung (a), Exposition, klassiert in die 8 Haupthimmelsrichtungen (b), Plankurvatur (c) und Rauigkeit (d) [°]

## Hangneigung

Der Geländeparameter Hangneigung wird durch die Ableitung ersten Grades des DHMs definiert. Durch die erste Ableitung erhält man, analog zur Tangente einer Kurve, eine Tangentialebene einer Oberfläche. Die Hangneigung entspricht der maximalen Höhenänderungsrate der Tangentialebene, auch als Gradient bezeichnet. Es gibt verschiedene Möglichkeiten, um die Hangneigung des Zentralpixels innerhalb des Zellfensters abzuschätzen. Zumeist werden dafür finite Differenzen oder Polynome gebildet (Burrough et al. 2015).

Der Ansatz von Horn (1981) berücksichtigt zur Berechnung der finiten Differenzen alle 8 Randpixel eines 3x3 pixelgrossen Zellfensters. Die Höhendifferenzen, die durch das Zentralpixel führen, werden bei dieser Methode doppelt gewichtet. Der Gradient in x-Richtung berechnet sich folgendermassen (Gl. 2.1 und Abb. 2.13):

$$\frac{dz}{dx} = \frac{(z_3 - z_1) + 2(z_6 - z_4) + (z_9 - z_7)}{8\Delta x} \quad \text{Gl. 2.1}$$

Analog dazu wird der Gradient in y-Richtung berechnet (Gl. 2.2 und Abb. 2.13):

$$\frac{dz}{dy} = \frac{(z_9 - z_3) + 2(z_8 - z_2) + (z_7 - z_1)}{8\Delta y} \quad \text{Gl. 2.2}$$

$\Delta x = \Delta y = L = \text{Zellgrösse des Gitternetzes}$

Daraus kann die Hangneigung folgendermassen berechnet werden (Gl. 2.3):

$$\text{Hangneigung} = \tan^{-1} \sqrt{\left(\frac{dz}{dx}\right)^2 + \left(\frac{dz}{dy}\right)^2} \quad \text{Gl. 2.3}$$

Der Wertebereich der Hangneigung reicht dabei von  $0^\circ$  bis  $90^\circ$  (Zevenbergen und Thorne 1987).

### Exposition

Der Geländeparameter Exposition wird ebenfalls durch die Ableitung ersten Grades des DHMs definiert. Die Exposition ist bestimmt durch die Himmelsrichtung, in welche die maximale Höhenänderungsrate der Tangentialebene zeigt. Die Exposition ist gegeben durch Gl. 2.4 (Olaya 2009):

$$\text{Exposition} = 180 - \tan^{-1} \left( \frac{\frac{dz}{dy}}{\frac{dz}{dx}} \right) + 90 \times \frac{\frac{dz}{dx}}{\left| \frac{dz}{dx} \right|} \quad \text{Gl. 2.4}$$

Der Wertebereich der Exposition reicht dabei von  $0^\circ$  bis  $360^\circ$  (Zevenbergen und Thorne 1987).

### Kurvatur

Die Krümmung ist gegeben durch die zweite Ableitung des DHMs. Dieser Geländeparameter beschreibt die Krümmung der Oberfläche, welche als konkav, konvex oder plan (Abb. 2.15) bezeichnet werden kann (Olaya 2009). Hierbei kann zwischen Profil- und Plankurvatur unterschieden werden. Die Profilkurvatur entspricht der Änderungsrate der Hangneigung in Richtung des steilsten Gefälles. Die Plankurvatur entspricht der Änderungsrate der Exposition quer zur Richtung des steilsten Gefälles (Zevenbergen und Thorne 1987).

Nach Zevenbergen und Thorne (1987) kann die Krümmung für ein 3x3 pixelgroße Zellenfenster folgendermaßen berechnet werden:

Zuerst wird ein lokales Polynom vierten Grades (Gl. 2.5) mit neun Parametern (Gl. 2.6 bis Gl. 2.14 und Abb. 2.13) definiert, welches durch alle neun Höhenwerte des 3x3 pixelgroßen Zellenfensters verläuft.

		Plankrümmung		
		konvex	gerade	konkav
Profilkrümmung	konvex			
	gerade			
	konkav			

Abb. 2.15: Geländeformen bezüglich Profil- und Plankrümmung nach Dikau (1989) von Vontobel (2011)

$$Z = f(x, y) = Ax^2y^2 + Bx^2y + Cxy^2 + Dx^2 + Ey^2 + Fxy + Gx + Hy + I \tag{Gl. 2.5}$$

$$A = \frac{\frac{z_1 + z_3 + z_7 + z_9}{4} - \frac{z_2 + z_4 + z_6 + z_8}{2} + z_5}{L^4} \tag{Gl. 2.6}$$

$$B = \frac{\frac{z_1 + z_3 + z_7 + z_9}{4} + \frac{z_2 - z_8}{2}}{L^3} \tag{Gl. 2.7}$$

$$C = \frac{\frac{z_1 + z_3 + z_7 + z_9}{4} + \frac{z_4 - z_6}{2}}{L^3} \tag{Gl. 2.8}$$

$$D = \frac{\frac{z_4 + z_6}{2} - z_5}{L^2} \tag{Gl. 2.9}$$

$$E = \frac{\frac{z_2 + z_8}{2} - z_5}{L^2} \tag{Gl. 2.10}$$

$$F = \frac{-z_1 + z_3 + z_7 - z_9}{4L^2} \tag{Gl. 2.11}$$

$$G = \frac{-z_4 + z_6}{2L} \tag{Gl. 2.12}$$



$$H = \frac{z_2 - z_8}{2L} \quad \text{Gl. 2.13}$$

$$I = z_5 \quad \text{Gl. 2.14}$$

Anschliessend kann daraus die Plankurvatur (Gl. 2.15) und Profilkurvatur (Gl. 2.16) berechnet werden:

$$\text{Plankurvatur} = \frac{-2(DG^2 + EH^2 + FGH)}{G^2 + H^2} \quad \text{Gl. 2.15}$$

$$\text{Profilkurvatur} = \frac{2(DH^2 + EG^2 + FGH)}{G^2 + H^2} \quad \text{Gl. 2.16}$$

Die hieraus folgenden Krümmungswerte liegen in  $\left[\frac{1}{LE}\right]$  vor, wobei LE lineare Einheit, beispielsweise 1m, bedeutet. Da die Krümmungswerte häufig sehr klein ausfallen, werden sie oftmals, wie auch für diese Arbeit, mit dem Faktor 100 multipliziert und daher als  $\left[\frac{1}{100 LE}\right]$  gelesen werden. Für die Profilkurvatur bedeuten positive Werte konkave und negative Werte konvexe Krümmung. Bezüglich der Plankurvatur bedeuten negative Werte konkave und positive Werte konvexe Krümmung. Ein Wert von 0 entspricht einer planen Oberfläche (Jeness 2013).

### Geländerauigkeit

Geländerauigkeit wird im Allgemeinen als Mass für topographische Unebenheit angesehen. Ein ideales Mass für die Rauigkeit sollte die Variabilität der Hangneigung und Exposition gleichzeitig abbilden können. Die meisten Methoden zur Quantifizierung der Geländerauigkeit basieren auf der Hangneigung und korrelieren darum stark mit dieser. Daher können diese Indizes nicht klar zwischen steilem, ebenem Gelände (hohe Hangneigung und geringe Rauigkeit) und steilem, unebenem Gelände (hohe Hangneigung und hohe Rauigkeit) unterscheiden (Sappington et al. 2007).

Die Methode *Vector Ruggedness Measure* (VRM) nach Sappington et al. (2007) basiert auf der Erfassung der Variabilität der Normalvektoren der Pixel innerhalb des Zellfensters (Abb. 2.16). Diese Methode zur Berechnung der Geländerauigkeit korreliert kaum mit der Hangneigung und wird deswegen im weiteren Verlauf verwendet. Dabei werden die Einheitsnormalvektoren bei bekannter Hangneigung und Exposition mit Hilfe trigonometrischen Operationen in ihre x-, y und z-Komponenten (Gl. 2.17 bis Gl. 2.20) zerlegt. Der Winkel  $\alpha$  entspricht dabei

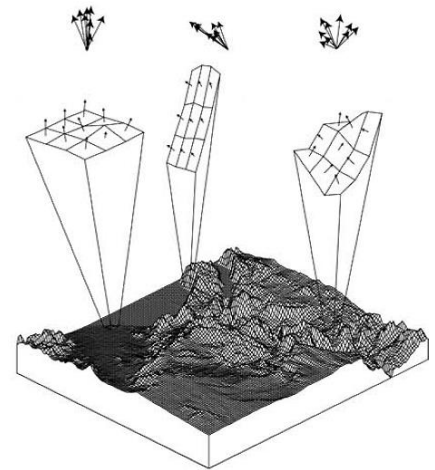


Abb. 2.16: Schematische Darstellung der Vektordispersion der Methode VRM (Sappington et al. 2007)

der Hangneigung, der Winkel  $\beta$  der Exposition (Abb. 2.17). Durch das Aufsummieren der einzelnen Komponenten jedes einzelnen Pixels erhält man einen resultierenden Vektorbetrag pro Zellfenster (Gl. 2.21 und Abb. 2.18). Der Betrag des resultierenden Vektors wird durch die Division anhand der Anzahl Pixel des Zellfensters normalisiert und von 1 subtrahiert (Gl. 2.19). Der finale Wert für die Geländerauigkeit ist dimensionslos und reicht von 0 bis 1. Je höher der Wert, desto rauer die Oberfläche (Sappington et al. 2007).

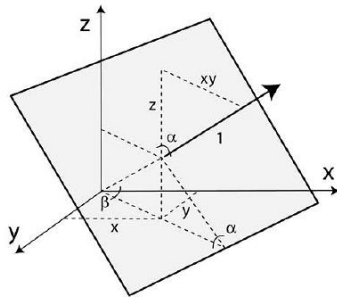


Abb. 2.17: Zerlegung eines Einheitsnormalvektors in seine x-, y- und z-Komponenten (Sappington et al. 2007)

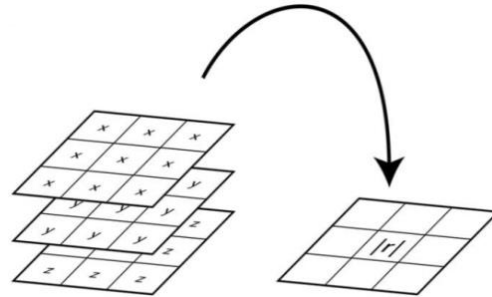


Abb. 2.18: Berechnung des resultierenden Vektorbetrages  $|r|$  (Sappington et al. 2007)

$$xy = 1 \times \sin(\alpha) \quad \text{Gl. 2.17}$$

$$x = xy \times \sin(\beta) \quad \text{Gl. 2.18}$$

$$y = xy \times \cos(\beta) \quad \text{Gl. 2.19}$$

$$z = 1 \times \cos(\alpha) \quad \text{Gl. 2.20}$$

$$|r| = \sqrt{(\sum x)^2 + (\sum y)^2 + (\sum z)^2} \quad \text{Gl. 2.21}$$

$$R = 1 - \frac{|r|}{n} \quad \text{Gl. 2.22}$$

## 2.4 Objektbasierte Bildanalyse

Durch die Fernerkundung gewonnene Primärdaten können mit Hilfe digitaler Verarbeitung anwendungsbezogen in bestimmte Sekundärdaten umgewandelt und, in Verbindung mit anderen Datensätzen, auch in geographischen Informationssystemen (GIS) verwendet werden. In der Vergangenheit wiesen Fernerkundungsdaten Pixelgrößen auf, die typischerweise grösser oder bestenfalls ähnlich der Grösse der zu untersuchenden Objekte waren. Bildanalysen, beispielsweise zur Detektion von Objekten, waren pixel- oder sogar subpixelbasiert. Durch die fortschreitenden technologischen Entwicklungen wurde jedoch die räumliche Auflösung von Fernerkundungsdaten kontinuierlich verbessert. Aufgrund der erhöhten räumlichen Auflösung können aus Fernerkundungsdaten Objekte, bestehend aus mehreren Pixeln, hergeleitet werden (Abb. 2.19). Damit hat das Konzept der objektbasierten Bildanalyse auch in der Fernerkundung Einzug gehalten. Der englische Begriff für objektbasierte Bildanalyse lautet *Object Based Image Analysis* und wird meist mit OBIA abgekürzt. Im Gegensatz zur pixelbasierten Analyse kann bei der objektbasierten Analyse zusätzlich zu spektralen Signatur eines Pixels oder Objektes die Textur, Form und Nachbarschaft zur Analyse verwendet werden (Blaschke 2010).

### 2.4.1 Multiskalige Segmentation

Eine wichtige Methode bei der objektorientierten Bildverarbeitung ist die Bildsegmentierung (Baatz und Schäpe 2000). Ein Segmentierungsprozess unterteilt ein Bild (Abb. 2.20 (a)) in relativ gleichartige, ähnliche Gruppen von Pixeln. Diese Gruppen sind mögliche Objektkandidaten, die in weiteren Verarbeitungsschritten auf Basis von Attributwerten zu homogenen Objekten vereinigt werden. Es gilt zu erwähnen, dass in Abhängigkeit der betrachteten räumlichen Skala, unterschiedliche Objekte detektiert werden können. Beispielsweise können auf einer groben räumlichen Skala Felder und Wälder als Objekte identifiziert werden (Abb. 2.20 (c)). Auf einer

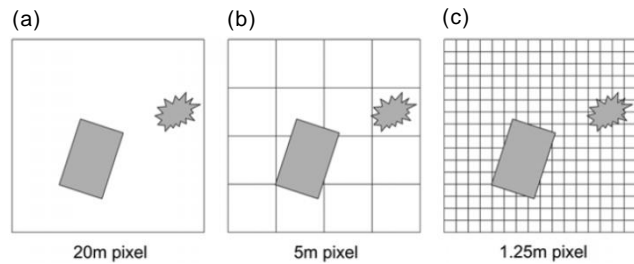


Abb. 2.19: Verhältnis zwischen dem zu untersuchenden Objekt und der räumlichen Auflösung: tiefe Auflösung (a): Pixel deutlich grösser als Objekt, subpixelbasierte Analyse nötig; mittlere Auflösung (b): Pixel und Objekt in der gleichen Grössenordnung, pixelbasierte Analyse angebracht; hohe Auflösung (c): Pixel sind deutlich kleiner als Objekte, Regionalisierung von Pixeln zu Pixelgruppen und schliesslich Objekten erforderlich (Blaschke 2010)

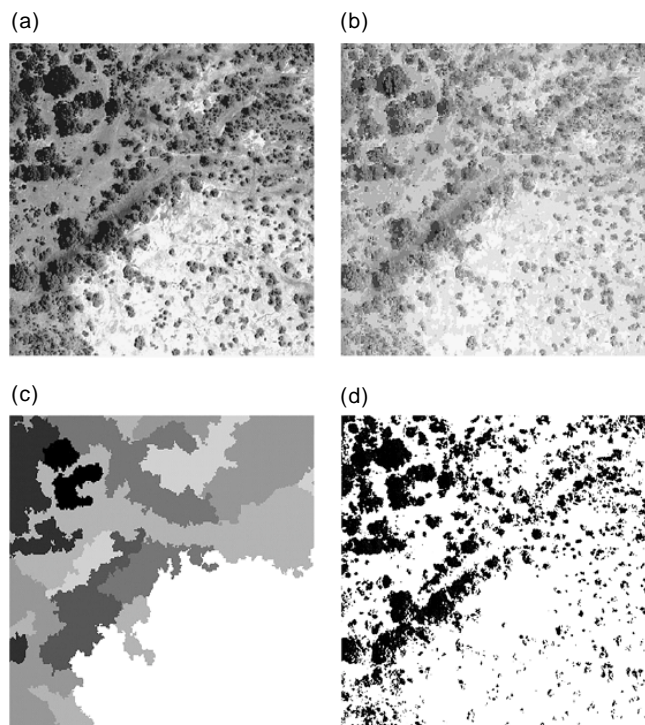


Abb. 2.20: Luftbild einer heterogenen Landschaft (a), feinskalige Segmentierung (b), grobskalige Segmentierung (c) und objektbasierte Klassifikation von Gehölzen (d) (Blaschke 2010)

kleineren räumlichen Skala können einzelne Bäume als Objekte detektiert werden (Abb. 2.20 (b)). Diese aber wiederum zu Wäldern zusammengefasst werden können (Blaschke 2010). Das Konzept der objektbasierten Bildanalyse greift dies auf: Die Entscheidung zur Vereinigung von Objektkandidaten basiert auf den Attributwerten der Objektkandidaten, die aus dem vorhergehenden Vereinigungsschritt resultierten. Die Vereinigungsentscheidung vollzieht sich folglich jeweils variabel auf der Skala der betrachteten Objektkandidaten. Schliesslich können damit Fernerkundungsdaten objektbasiert analysiert und klassiert werden (Abb. 2.20 (d)) (Baatz und Schäpe 2000).

## 2.5 Räumliches Modellieren

In Abhängigkeit des Verständnisses des zu modellierenden Phänomens kann zwischen einem direkten und indirekten Ansatz zur räumlichen Vorhersage unterschieden werden: Beim direkten Ansatz wird angenommen, dass ein Phänomen durch eine bestimmte Anzahl an Eingangsparametern und eine entsprechende, bekannte mathematische Funktion oder ein physikalisches Gesetz modelliert werden kann. Der indirekte Ansatz kommt zur Anwendung, wenn für das zu modellierende Phänomen die Definition der zu verwendenden Eingabeparameter und/oder die genauen Gesetzmässigkeiten/Wirkungsweisen zur Formulierung einer mathematischen Funktion nicht explizit definierbar sind. Die indirekten Modelle können in die zwei Subkategorien pure statistische Modelle und Experten-basierte Modelle, unterteilt werden (Hengl und MacMillan 2009). Häufig verwenden Anwendungen in der Naturgefahrenmodellierung aufgrund der Komplexität der zugrunde liegenden Prozesse und der oft begrenzten Eingangsdaten den indirekten Ansatz (Veitinger 2015).

Zur Erstellung eines Experten-basierten Modells ist eine vernünftige, konzeptionelle Vorstellung über das zu modellierende Phänomen nötig. Die Herausforderung besteht darin, eine Methode zu finden, um mit Hilfe von relevanten Eingabedaten ein Phänomen möglichst akkurat zu prognostizieren. Hengl und MacMillan (2009) unterscheiden dabei die Fälle von sehr limitiertem, partiellem und exaktem Expertenwissen.

Da es für diese Arbeit relevant ist, wird näher auf das Modellieren mit partiellem Expertenwissen eingegangen: Bezüglich räumlicher Modellierung hat in diesem Fall der Experte eine allgemeine Vorstellung über die Bedingungen und Prozesse, unter welchen das betreffende Phänomen sich entwickelt und wo im Raum es typischerweise auftritt. Ziel ist es, einen Algorithmus zu entwickeln, der in der Lage ist, mit Hilfe von Ausgangsdaten das räumliche Auftreten des Phänomens zu modellieren. Diese Algorithmen können iterativ angewendet, überprüft und überarbeitet werden, sodass die Vorhersage mit dem tatsächlichen räumlichen Muster immer besser übereinstimmt (Hengl und MacMillan 2009). Die Bedingungen, unter welchen ein räumliches Phänomen auftritt, können anhand der Ausgangsdaten abgebildet werden. Dafür wird für eine Ausgangsdatei (Hangneigungsraaster) jede Raumeinheit (Pixel) anhand einer Klassifikationsregel, bezüglich einer bestimmten Bedingung (Lawinenanrissgebiet bedingt eine Hangneigung von rund 30-60°), klassiert. Die klassierten

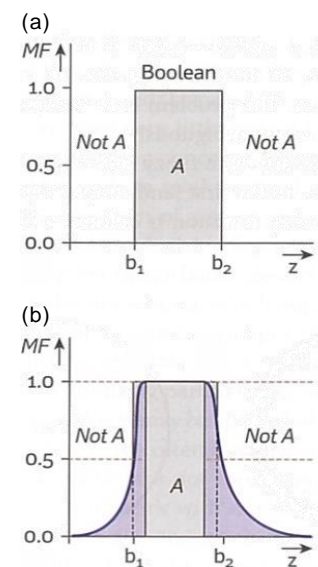


Abb. 2.21: Zugehörigkeitsfunktionen graphisch dargestellt: *Boolean* (a) und *Fuzzy* (b) (Burrough et al. 2015)

Daten werden anschliessend kombiniert, um abzuschätzen, welche Raumeinheit zu welchem Grad die gestellten Bedingungen erfüllt. Diese Klassifikationsregeln können anhand der Logik und Mathematik des Ansatzes *Boolean* oder *Fuzzy* formuliert werden (Veitinger 2015).

Der Grad, ob eine betrachtete Raumeinheit Mitglied einer bestimmten Klasse ist, wird durch eine Zugehörigkeitsfunktion ausgedrückt. Im Falle von *Boolean* ist die binäre Zugehörigkeit durch scharfe Klassengrenzen definiert und die Zugehörigkeitswerte betragen entweder 0 oder 1 (Gl. 2.23 und Abb. 2.21). Im Falle von *Fuzzy* ist der Grad der Zugehörigkeit zu einer Klasse durch eine Zugehörigkeitsfunktion definiert und deren Wertebereich ist kontinuierlich von 0 bis 1 (Gl. 2.24 und Abb. 2.21). Dabei repräsentiert 0 keine Zugehörigkeit und 1 eine volle Zugehörigkeit einer bestimmten Klasse (Burrough et al. 2015).

$$\begin{aligned} MF_A^B(z) &= 1, & \text{if } b_1 \leq z \leq b_2 \\ MF_A^B(z) &= 0, & \text{if } z < b_1 \text{ or } z > b_2 \end{aligned} \tag{Gl. 2.23}$$

$MF_A^B(z)$ : Zugehörigkeitsfunktion Boolean für die Klasse A  
 $b_1, b_2$ : Klassengrenzen der Klasse A

$$MF_A^F(z) = [0,1] \tag{Gl. 2.24}$$

$MF_A^F(z)$ : Zugehörigkeitsfunktion Fuzzy für die Klasse A

Die Kopplung einer automatischen Identifizierung potentieller Anrissgebiete mit dynamischer Lawinensimulationssoftware ermöglicht eine automatische Gefahrenhinweiskartierung. Als Eingabedaten benötigt gängige Lawinensimulationssoftware, wie RAMMS, scharf abgegrenzte Lawinenanrissgebiete als Polygone (Christen et al. 2010). Darum ist bei der Identifizierung potentieller Anrissgebiete am Ende eine *Boolean* Klassierung nötig. Bei natürlichen Phänomenen, wie Lawinenanrissgebieten, sind jedoch in Realität kaum durch scharfe Grenzen abtrennbar. So ist beispielsweise in Abb. 2.3 nur die Anrisskante klar erkennbar. Solch eine scharfe Abgrenzung von Lawinenanrissgebieten ist deshalb eine starke Abstrahierung der Realität (Burrough et al. 2015).

## 3 Algorithmen zur automatischen Identifizierung potentieller Anrissgebiete

### 3.1 Forschungsstand

Die Geomorphometrie (Kapitel 2.3) ermöglicht eine schnelle und konsistente Ableitung von Geländeparametern. Diese Geländeparameter können als Kriterien dienen, um ein mögliches Auftreten von gravitativen Naturgefahren zu lokalisieren (Hengl und MacMillan 2009). In der Literatur sind diverse Ansätze und Fallbeispiele zur Identifizierung von Anrissflächen für Erdbeben (Carrara 1983, Carrara et al. 1995, Singh et al. 2005, Gruber et al. 2009, Pradhan und Buchroithner 2012) sowie für Murgänge und Steinschläge (Singh et al. 2005, Michoud et al. 2012) auf Basis von Geländemodellen zu finden. Bezüglich diesen gravitativen Massenbewegungen empfiehlt Cascini (2008) für eine regionale Gefahrenhinweisabschätzung die Anwendung Experten-basierter Modelle in Kombination mit Geomorphometrie. Der wichtigste Geländeparameter in Zusammenhang mit Erdbeben, Murgängen und Steinschlägen ist die Hangneigung, welche die Verteilung instabiler Gebiete stark bestimmt (Carrara et al. 1995). Die Exposition und die Krümmung werden ebenfalls berücksichtigt (Singh et al. 2005, Pradhan und Buchroithner 2012). Neben diesen Massenbewegungen wurde auch für den Prozess Lawine versucht, anhand des Geländemodells potentielle Anrissgebiete zu identifizieren, was im Folgenden näher besprochen wird.

Wie im Kapitel 2.1.2 beschrieben, stellen das Gelände, die Schneedecke und das Wetter die drei lawinenbildenden Faktoren dar. Die Faktoren Schneedecke und Wetter variieren ständig in Raum und Zeit. Direkte Untersuchungen des Schneedeckenaufbaus und wetterspezifische Messungen sind relativ aufwendig und können nur punktuell durchgeführt werden. Die Beurteilung der Lawinengefahr erfolgt darum grösstenteils indirekt anhand äusserer Indikatoren und Faktoren, beispielsweise dem Gelände (Harvey et al. 2013). Der Faktor Gelände ist der lawinenbildende Faktor, der, im Vergleich zur Schneedecke und dem Wetter, über die Zeit relativ konstant bleibt. Anders gesagt: Das Gelände wird als statisch, die Schneedecke und das Wetter als zeitlich variabel aufgefasst (Hengl und MacMillan 2009). Zudem sind digitale Höhenmodelle dank Fortschritten in der Fernerkundung in hoher räumlicher Auflösung ( $\leq 10\text{m}/\text{Pixel}$ ) verfügbar (Kapitel 2.3.1). Mit Hilfe der Geomorphometrie können aus einem digitalen Höhenmodell Geländeparameter abgeleitet werden (Kapitel 2.3.2). Diese Parameter ermöglichen eine Abschätzung der lawinenbildenden Faktoren (Kapitel 2.1.2), die im Zusammenhang mit dem Gelände stehen. Diese Parameter können als Ausgangsdaten für eine räumliche Modellierung dienen (Kapitel 2.5). Geomorphometrische Modelle beruhen auf einer einfachen Parametrisierung der physikalischen Prozesse und haben dadurch den Vorteil, dass sie wenig Ausgangsdaten (DHM und optional Vegetation) voraussetzen und über vergleichsweise kurze Berechnungszeiten verfügen, was sich für die Verwendung einer regionalen Gefahrenhinweiskartierung besonders gut eignet (Gruber et al. 2009). So wurden inzwischen mehrere Ansätze entwickelt, um mögliche Lawinenanrissgebiete anhand von Geländeparametern zu identifizieren:

### 3.1.1 Bestehende Algorithmen

Bereits Voellmy (1955) stellte fest, dass der Geländeparameter Hangneigung eine entscheidende Rolle bei der Identifizierung potentieller Anrissgebiete spielt. Mit der Entwicklung von GIS und der Fernerkundung konnten aus digitalen Höhenmodellen Geländeparameter abgeleitet werden (Kapitel 2.3). Die ersten automatisierten Algorithmen zur Identifizierung potentieller Anrissgebiete, die mehrere unterschiedliche Geländeparameter berücksichtigten, entstanden nach dem Jahr 2000 mit der Verfügbarkeit von DHMs mit recht grober Auflösung im Bereich von 25 bis 30 Metern (Maggioni et al. 2002, Maggioni und Gruber 2003, Maggioni 2005). Dabei wurden mit Hilfe von Lawinenkatastern die geländespezifischen Eigenschaften von Lawinenanrissgebieten analysiert und die auch in Verbindung mit der Häufigkeit einer Auslösung gebracht. Anhand dieser Studien wurde im Rahmen des Projekts SilvaProtect-CH erstmal schweizweit potentielle Anrissgebiete automatisch identifiziert und eine Gefahrenhinweiskartierung für den Prozess Lawine erstellt (Losey 2013). Dieser Ansatz verwendet *Boolean*-Zugehörigkeitsfunktionen. Ghinoi und Chung (2005) analysierten ebenfalls anhand von historischen Lawinenaufzeichnungen die geländespezifischen Eigenschaften von Anrissgebieten, in Bezug auf drei spezifische, meteorologische Szenarios. Darauf basierend wurde versucht mit Hilfe von Geländeparametern Anrissgebiete mit *Fuzzy*-Zugehörigkeitsfunktionen zu identifizieren. Neben den elementaren Geländeparametern (Hangneigung, Exposition, Krümmung) versuchten Ghinoi und Chung (2005) anhand von Luftbildern die Rauigkeit des Geländes qualitativ zu klassifizieren und ebenfalls bei der Anrissgebietenidentifizierung zu berücksichtigen. Des Weiteren berücksichtigten sie die Höhe über Meer als Indikator, um den Zusammenhang zwischen der Niederschlagsmenge und der Höhe über Meer miteinzubeziehen. Barbolini et al. (2011) nutzt zur Anrissgebietenidentifizierung die Geländeparameter Hangneigung und Plankrümmung sowie räumliche Daten zur Waldbedeckung mit *Boolean*-Zugehörigkeitsfunktionen. Zusätzlich wurde ein Algorithmus entwickelt um die Lawinenbahn und die Auslaufdistancen zu modellieren. Die Modellierung basiert auf einem FD8 (Gruber Schmid und Peckham 2009) multiplen Fließrichtungsalgorithmus und einer Abschätzung des Pauschalgefälles. In Kombination konnten damit Lawinen simuliert und Gefahrenabschätzungen vorgenommen werden. Andres und Chueca Cía (2012) verwenden ebenfalls die Hangneigung, die Profilkrümmung und die Bodenbedeckung (Indikator für die Geländerauigkeit) zusammen mit *Boolean*-Zugehörigkeitsfunktionen zur Anrissgebietenidentifizierung. Darüber hinaus wird eine Untergrenze bezüglich der Höhe über Meer gesetzt, über welcher der Niederschlag theoretisch als Schnee fällt. Als diese Untergrenze dient eine mittlere 0°C Isotherme. Das gleiche Verfahren wurden auch bei der Studie von Chueca Cía et al. (2014) angewendet. Pistocchi und Notarnicola (2013) verwendeten die elementaren Geländeparameter Hangneigung, Exposition und Krümmung sowie räumliche Informationen zur Waldbedeckung. Darüber hinaus wird ein topographischer Feuchtigkeitsindex, die Distanz zum nächstgelegenen Geländekante und räumlich-zeitliche Informationen zur Schneebedeckung zur Anrissgebietenidentifizierung berücksichtigt. Sie nutzten logistische Regressionsanalysen, um eine Pixel auf seine Lawinenanrissanfälligkeit hin zu bewerten.

Durch den anhaltenden technologischen Fortschritt in der Fernerkundung konnten fortlaufend DHMs mit höherer Auflösung prozessiert werden (Kapitel 2.3.1). DHMs mit höherer räumlicher Auflösung ( $\leq 10$  m) ermöglichen die Berechnung einer topographischen abgeleiteten Geländerauigkeit in hochauflöser Form. Neben den anderen elementaren Geländeparametern wird die Rauig-

keit als wichtiger Indikator zur Identifizierung potentieller Lawinenanrissgebiete erachtet (McClung 2001, Schweizer et al. 2003, van Herwijnen und Heierli 2009, Vontobel 2011, Phillips et al. 2017). Daraufhin wurden in der jüngeren Vergangenheit Algorithmen (Bühler et al. 2013, Veitinger et al. 2016, Bühler et al. 2018) entwickelt werden, welche neben anderen Geländeparametern auch die Rauigkeit zur Anrissgebietsidentifizierung miteinbeziehen. Tab. 3.1 gibt einen Überblick über die bestehenden Algorithmen zur Identifizierung potentieller Anrissgebiete, welche in von Experten begutachteten Wissenschaftszeitschriften veröffentlicht wurden.

Tab. 3.1: Überblick über bestehende Algorithmen, inklusive der verwendeten Parameter

Publikationen (chronologisch)	Hangneigung	Plan- kurva- tur	Profil- kurva- tur	Rau- igkeit	Expo- sition	Distanz zur nächsten Geländekante	Wind- schatten- index	Topograph. Feuchtig- keitsindex	Wald
Voellmy (1955)	x								
Maggioni et al. (2002)	x	x			x	x			
Ghinoi und Chung (2005)	x	x	x	(x)	x				
Barbolini et al. (2011)	x	x							x
Andres und Chueca Cía (2012)	x		x	(x)					
Pistocchi und Notarnicola (2013)	x	x	x		x	x		x	x
Bühler et al. (2013)	x	x		x					x
Chueca Cía et al. (2014)	x		x	(x)					
Veitinger et al. (2016)	x			x			x		
Bühler et al. (2018)	x	x	x	x	x				x

### 3.1.2 Ausgewählte Algorithmen

In dieser Arbeit werden aus Tab. 3.1 die folgenden vier Algorithmen miteinander verglichen: Der simple Ansatz von Voellmy (1955) sowie die drei Algorithmen von Bühler et al. (2013), Veitinger et al. (2016) sowie der neu entwickelte von Bühler et al. (2018). Der simple und älteste Ansatz von Voellmy (1955) wurde gewählt, weil er im Algorithmenvergleich als Richtwert für die neueren, verfeinerten Algorithmen dienen soll. Der Algorithmus von Bühler et al. (2013), Veitinger et al. (2016) und Bühler et al. (2018) wurden unter anderem gewählt, da diese die einzigen drei Algorithmen sind, welche neben anderen Geländeparametern auch eine topographisch basierte Geländerauigkeit miteinbeziehen.



Der Algorithmus von Maggioni et al. (2002) wurde im Rahmen eines Projektes für den Kanton Graubünden von Bühler et al. (2017) getestet. Der Algorithmus erzielte jedoch nur bei DHMs mit einer Auflösung im Bereich von 25m sinnvolle Resultate (Bühler et al. 2017). Des Weiteren ist dieser Algorithmus in der AML Programmiersprache geschrieben und kann mit der heutigen Software nicht mehr verwendet werden. Ferner sind die anderen Algorithmen von Ghinoi und Chung (2005), Barbolini et al. (2011), Andres und Chueca Cía (2012), Pistocchi und Notarnicola (2013) sowie Chueca Cía et al. (2014) nicht vorliegend, da sie von anderen Forschungsgruppen entwickelt wurden.

### Voellmy (1955)

Wie bereits in Kapitel 3.1 erwähnt, wurde schon früh erkannt, dass die Auslösung von Lawinen mit der Hangneigung zusammenhängt. Sinnbildlich dafür ist der Artikel von Voellmy (1955) in der Schweizerischen Bauzeitung über die Zerstörungskraft von Lawinen. Darin steht geschrieben, dass der grösste Teil der Lawinenanrisse in Gebieten mit 30 bis 40° Hangneigung auftritt. Gemäss neueren Erkenntnissen (Kapitel 2.1.2) weisen Anrissgebiete von Schneebrettlawinen in der Regel Hangneigungen von rund 28 bis 50°, maximal 60° auf (Schweizer et al. 2003). Dieser Ansatz wird auch von Experten zur Erstellung von Gefahrenkarten für Lawinen angewendet (Nairz et al. 2011). Der simple Ansatz nach Voellmy (1955) zur Identifizierung potentieller Anrissgebiete wird folgendermassen umgesetzt (Abb. 3.1): Auf Basis des digitalen Geländemodelles wird für jedes Pixel dessen Hangneigung berechnet. Die Berechnung der Hangneigung erfolgt anhand der Methode in Kapitel 2.3.2. Hierbei werden Pixel zwischen 28° und 60° als potentielle Anrissgebiete klassiert. Die möglichen Anrissflächen werden aber nicht in einzelne Anrissgebiete unterteilt. Das Ergebnis dieses Ansatzes wird als Vergleichsgrösse für die anderen Algorithmen herangezogen.

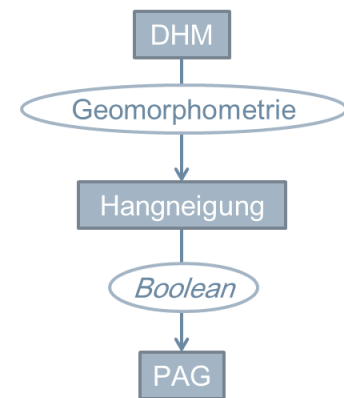


Abb. 3.1: Schematischer Prozessablauf des Algorithmus nach Voellmy (1955)

### Bühler et al. (2013)

Der Algorithmus von Bühler et al. (2013) funktioniert nach dem Ansatz *Boolean*, beschrieben in Kapitel 2.5. Aus dem DHM werden Raster mit den Geländeparametern Hangneigung, Plankurvatur und Rauigkeit gemäss Kapitel 2.3.2 abgeleitet (Abb. 3.2). Dieser Algorithmus ist der erste, welcher zur Identifizierung von potentiellen Anrissgebieten die Rauigkeit, berechnet aus dem DHM, berücksichtigt. Hierbei ist die Grösse des Zellfensters zur Berechnung der Rauigkeit frei wählbar. Für die Parameter Hangneigung und Plankurvatur wird jeweils ein unterer und ein oberer Schwellenwert, für den Parameter Rauigkeit nur eine Obergrenze gesetzt. Für die Plankurvatur wird ein zweiseitiger symmetrischer Schwellenwert gesetzt. Damit ist gemeint, dass beispielsweise alle Pixel im Bereich von  $\pm 5/100\text{m}$  oder  $\pm 6/100\text{m}$  als potentielle Anrissgebiete in Frage kämen. Die Raster der jeweiligen Geländeparameter werden binär kodiert und anschliessend alle binär kodierten Geländeparameterraster zu einem einzigen binär kodierten Raster (0,1) kombiniert: 0 nPAG, 1 PAG. Optional können danach die PAG-Flächen mit einem Raster der Waldbedeckung überlagert werden und PAGs im Wald zu nPAGs umklassiert werden. Als Resultat erhält man die potentiellen Anrissflächen.

Diese Anrissflächen versucht der Algorithmus anschliessend in einzelne Gebiete zu unterteilen. Dafür werden die Hydrologiewerkzeuge *Flow Direction*, *Flow Accumulation* und *Watershed* von ArcMap (ESRI 2018c) benutzt. Nachfolgend werden PAG kleiner als eine bestimmte Minimalfläche ausgeschlossen. Als Resultat erhält man die potentiellen Anrissgebiete. Dieser Algorithmus wurde unter anderem zur Gefahrenbeurteilung bei der Planung eines Strassentunnels in der Region Manali, Himalaya, angewendet (Bühler et al. 2013).

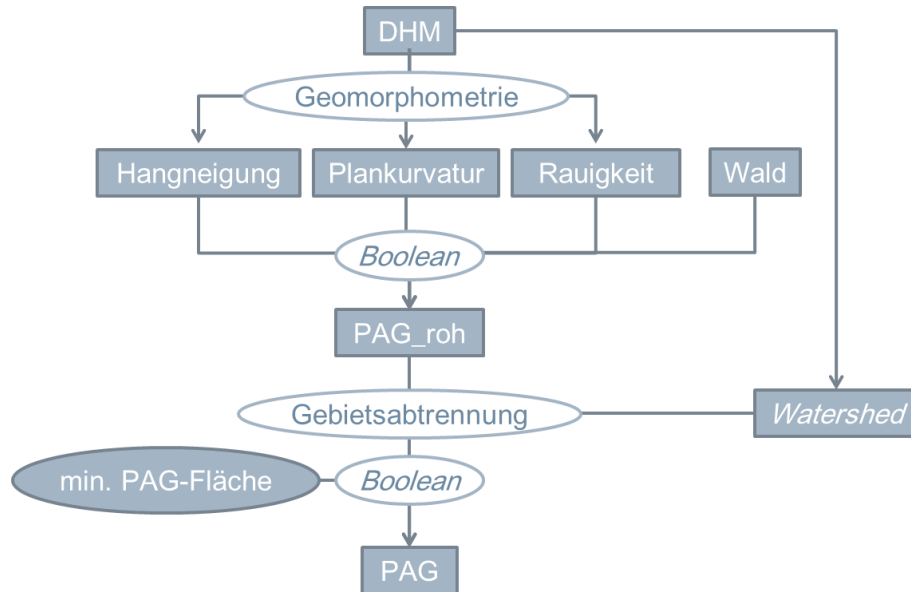


Abb. 3.2: Schematischer Prozessablauf des Algorithmus nach Bühler et al. (2013)

### Veitinger et al. (2016)

Der Algorithmus von Veitinger et al. (2016) funktioniert nach dem Ansatz *Fuzzy*, beschrieben in Kapitel 2.5. Der Algorithmus leitet aus dem DTM die Hangneigung, Geländerauigkeit und einen Windschattenindex ab (Abb. 3.3). Dabei wird eine multiskalige Berechnungsmethode eingeführt: Abhängig von der Schneehöhe wird die Grösse des Zellfensters zur Berechnung der Geländeparameter variiert. Je höher die angenommene Schneehöhe, desto grösser wird das Zellfenster. Folglich werden Eingabewerte für die Schneehöhe und zur Berechnung des Windschattenindex Angaben zur Windrichtung und Windrichtungsvariabilität benötigt. Die vorherrschende Windrichtung und deren Variabilität ist kaum eindeutig zu bestimmen (Schüepp und Urfer 1962), da die Region Davos unterschiedlichen Windregimen ausgesetzt ist. Darüber hinaus kann in den meisten Regionen ein Gefahrenhinweiskartierungsszenario nicht zuverlässig mit einem bestimmten Windregime verknüpft werden. Deswegen wird der Windschattenindex nicht verwendet. Bezüglich der Hangneigung, dem Windschattenindex und der Rauigkeit wendet Veitinger et al. (2016) anschliessend eine entsprechende *Fuzzy*-Zugehörigkeitsfunktion an. Basierend auf diesem multiskaligen Fuzzy-Logik-Ansatz resultiert eine Lawinenanrissanfälligkeit (PAG\_roh in Abb. 3.3) in einem kontinuierlichen Bereich von 0 (nicht anfällig) bis 1 (sehr anfällig). Die Koppelung mit dynamischer Lawinensimulationssoftware verlangt nach scharf abgegrenzten Lawinenanrisspolygonen (Kapitel 2.5). Die anderen Algorithmen in diesem Vergleich verwenden daher *Boolean*-Zugehörigkeitsfunktionen, wodurch deren Modellausgaben die beiden diskreten Klassen 0 (nPAG) und 1 (PAG) aufweisen. Zusätzlich

haben sie die Möglichkeit, PAG-Flächen, die kleiner sind als ein bestimmter Schwellenwert, zu eliminieren. Daher wird der Algorithmus von Veitinger et al. (2016) um die Möglichkeit erweitert, die kontinuierlichen Lawinenanrissanfälligkeitwerte auf Grund eines Schwellenwertes den diskreten Klassen PAG oder nPAG zuzuweisen und die Möglichkeit, eine minimal nötige Anrissgebietsfläche zu definieren. Dies ermöglicht, den Algorithmus mit den anderen vergleichen zu können. Dieser Algorithmus unterteilt die potentiellen Anrissflächen nicht weiter in individuelle Anrissgebiete auf. Er wurde vor allem für kurzfristige Gefährdungsbeurteilungen von einzelnen Lawineneinzugsgebieten entwickelt (Veitinger et al. 2016).

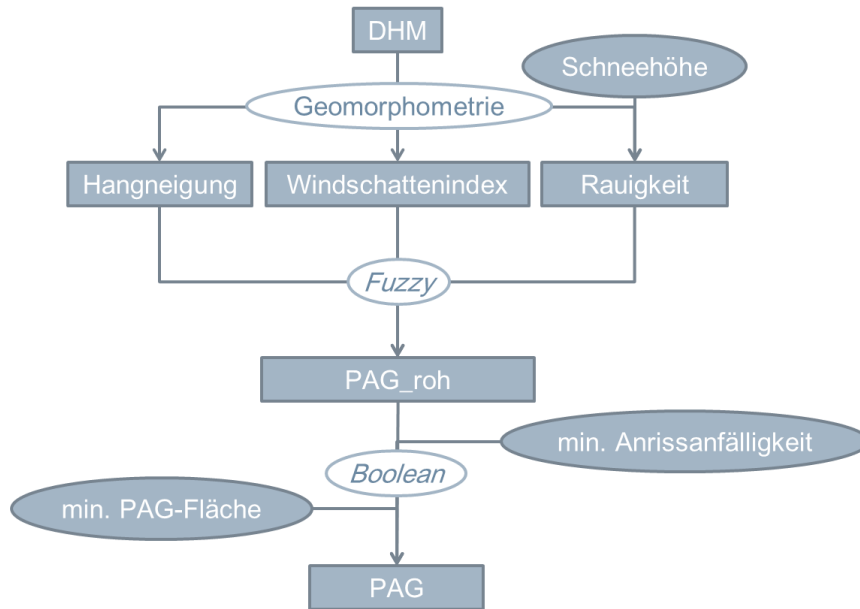


Abb. 3.3: Schematischer Prozessablauf des Algorithmus nach Veitinger et al. (2016)

### Bühler et al. (2018)

Der neu entwickelte Algorithmus von Bühler et al. (2018) ist in zwei Prozessierungsschritte gegliedert (Abb. 3.5): Als erstes wird eine Vorauswahl potentieller Anrissflächen, basierend auf Bühler et al. (2013), getroffen und in einem zweiten Schritt die potentiellen Anrissflächen mittels objektbasierter Bildanalyse in einzelne Anrissgebiete unterteilt.

Aus einem DHM werden, analog zum Algorithmus von Bühler et al. (2013), Raster der Geländeparameter Hangneigung, Plankurvatur und Rauigkeit berechnet, gemäss den eingegebenen Schwellenwerten binär kodiert und zu einer Vorauswahl potentieller Anrissflächen kombiniert. Optional können Anrissflächen im Wald aus der Vorauswahl entfernt werden. Im Unterschied zum Algorithmus von Bühler et al. (2013) werden die Rauigkeitswerte mit dem Faktor 100 multipliziert, was einen Wertebereich für die Rauigkeit von 0 bis 100 ergibt. Zusätzlich können Anrissflächen kleiner als eine gewisse Minimalfläche ausgeschlossen werden. Ebenfalls wurde erkannt, dass das Sommer-DHM mit 5x5m die

0.625	0.625	0.625	0.625	0.625
0.625	1.5	1.5	1.5	0.625
0.625	1.5	3	1.5	0.625
0.625	1.5	1.5	1.5	0.625
0.625	0.625	0.625	0.625	0.625

Abb. 3.4: 5x5 pixelgrosses, gleitendes Zellfenster mit dem entsprechenden Gewicht pro Pixel

winterliche Schneeoberfläche suboptimal repräsentiert. Darum wurde das Sommer-DTM leicht geglättet, um den Glättungseffekt der Geländeoberfläche durch die winterliche Schneebedeckung zu simulieren. Dafür wird mit Hilfe eines 5x5 pixelgrossen, gleitenden Zellfensters (Abb. 3.4) einen gewichteten Mittelwert für das jeweilige Zentralpixel berechnet.

Für die objektbasierte Bildanalyse werden die bei der Vorauswahl bestimmten Flächen in PAG- und nPAG-Gruppen segmentiert. Als nächstes werden die PAG-Gruppen zu Objektkandidaten segmentiert und zu homogenen Objekten vereinigt. Dabei wird ein multiskaliges Segmentationsverfahren (Kapitel 2.4.1) angewendet. Zur Segmentierung werden die Attribute Exposition (klassiert in die Sektoren Nord, Nordost, Ost, Südost, Süd, Südwest, West, Nordwest), Plankurvatur und Profilkurvatur benutzt. Die Exposition und die Plankurvatur dienen vor allem der seitlichen und die Profilkurvatur der Abgrenzung nach unten und oben. Die resultierenden Objekte stellen die potentiellen Anrissgebiete dar. Bei der Segmentierung kann über einen *Scale*-Faktor die Grösse der resultierenden Anrissgebiete gesteuert werden. Dadurch können szenarienspezifische Gebietsabtrennungen generiert werden. Dafür wurde ein Lawinenszenario *frequent* (mit einer Wiederkehrperiode von 5-30 Jahren) und ein Lawinenszenario *extreme* (mit einer Wiederkehrperiode von 100-300 Jahren) definiert. Anschliessend werden die Anrissgebiete in ihrer Form etwas generalisiert und als Vektordaten exportiert. Dieser Algorithmus wurde erstmals in einem Vorprojekt für den Kanton Graubünden zur flächendeckenden Gefahrenhinweiskartierung für den Prozess Lawine angewendet (Bühler et al. 2017).

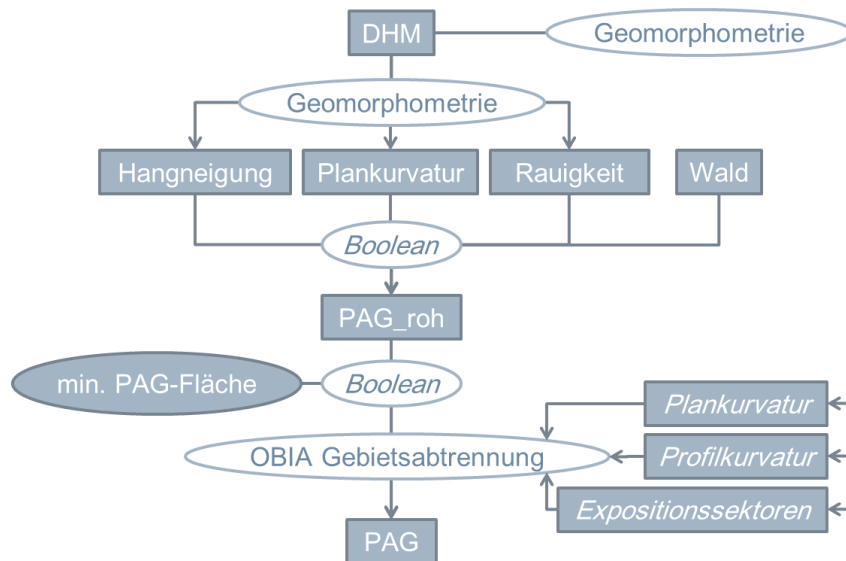


Abb. 3.5: Schematischer Prozessablauf des Algorithmus nach Bühler et al. (2018)

### 3.2 Verwendete Software

Zur Berechnung der Geländeparameter gemäss Kapitel 2.3.2 und zur Anwendung der zu vergleichenden Algorithmen bedarf es einer Software für Geodatenverarbeitung (GIS). Ausserdem wird für die Evaluierung und den Vergleich der Algorithmen gemäss Kapitel 4.1.4 ein Statistikprogramm benötigt.

### 3.2.1 Software zur Geodatenverarbeitung

Der Algorithmus von Bühler et al. (2013) wurde in Python für die Ausführung in ArcMap geschrieben. Der Algorithmus von Veitinger et al. (2016) ist eigentlich in R geschrieben, kann aber über ein Pythonskript ebenfalls in ArcMap ausgeführt werden. Diese zwei Algorithmen wurden in unveränderter Form von ihren Entwicklern (Bühler et al. 2013, Veitinger et al. 2016) übernommen. Da diese zwei Algorithmen bereits in dieser Form vorlagen, wurde der simple Ansatz von Voellmy (1955) ebenfalls in Eigenarbeit mit einem Pythonskript umgesetzt, welches im Anhang zu finden ist (Kapitel 9.1.1). Der Algorithmus von Bühler et al. (2018) bestand bereits, wurde während der Arbeit noch weiterentwickelt. Die Vorauswahl des neu entwickelten Algorithmus ist auch in Python für die Ausführung in ArcMap programmiert. Zur Kalibrierung und Validierung wurden eigene Pythonskripte zur Ausführung in ArcMap geschrieben. Diese sind ebenfalls im Anhang zu finden (Kapitel 9.1). Zur Kalibrierung diente eine Tabelle im csv-Format als Eingabedateien, welche die Parameterwerte, über die iteriert werden sollte, enthält (Anhang Kapitel 9.1.2). Dies automatisierte den ganzen Arbeitsprozess erheblich, da pro Algorithmus und Gebiet bis zu 100 Durchläufe nötig waren, und erlaubte so den beabsichtigten Algorithmenvergleich. ArcMap ist die primäre Anwendung in ArcGIS Desktop der Firma ESRI zur Kartierung, Verarbeitung und Analyse sowie zum Management von Geodaten (ESRI 2018b). Hierbei wurden ArcMap in den Versionen 10.4 und 10.5 verwendet. Python ist eine Programmiersprache mit frei verfügbarer Softwareumgebung (Stross-Radschinski 2018). Die Skripts sind alle in der Python Version 2.7 geschrieben. Als integrierte Entwicklungsumgebung (IDE) wurde PyScripter verwendet.

Die eigentliche Gebietsabtrennung des neu entwickelten Algorithmus Bühler et al. (2018), auf Grundlage der Vorauswahl, wird mit eCognition Developer der Firma Trimble prozessiert. Die Software eCognition Developer ist eine leistungsfähige Entwicklungsumgebung für die objektbasierte Bildanalyse. In dieser Arbeit kam eCognition Developer 9 zur Anwendung (Trimble 2018).

### 3.2.2 Statistikprogramm

Für die Kalibrierung und Validierung wurden die Ausgabedaten der Geodatenverarbeitung mit Hilfe eines R-Skripts ausgewertet und graphisch dargestellt. Wegen der grossen Anzahl an auszuwertenden Durchläufen war wiederum ein möglichst automatisierter Arbeitsprozess von Nöten. Das Skript dazu ist im Anhang zu finden (Kapitel 9.2). R ist eine Programmiersprache mit frei verfügbarer Softwareumgebung für statistische Berechnungen und Grafiken (R-Project 2018). Für die Anwendung von R wurde R Studio beigezogen. R Studio ist eine integrierte Entwicklungsumgebung (IDE) für R (Studio 2018).

## 3.3 Verwendetes DHM

Elementare Geländeparameter lassen sich mittels Geomorphometrie aus einem DHM ableiten (Kapitel 2.3.2). Daher bildet das DHM die Grundlage für die Ausscheidung potentieller Anrissgebiete. So kommt der Auflösung und der Qualität des DHMs eine entscheidende Bedeutung zu. In der Schweiz ist das DHM swissALTI<sup>3D</sup> flächendeckend mit einer räumlichen Auflösung von 2 m verfügbar. Dieses qualitativ sehr hochstehende DHM basiert auf flugzeuggestützte Laserdaten (unterhalb von ca. 2000 m ü. M.) sowie auf flugzeuggestützten photogrammetrischen Daten (oberhalb von ca. 2000 m

ü. M.). Zudem wird diese DHM manuell überprüft und korrigiert. Es wurde dabei auf eine möglichst laub- und schneefreie Erfassung der Laserdaten geachtet (swisstopo 2018b). Die Luftbilder zur Photogrammetrie wurden in den Monaten Juli, August und September aufgenommen, das heisst, wenn die Schneedecke in hohen Lagen ihr Minimum erreicht (Bovet 2013). Die Genauigkeit in allen drei Dimension beträgt  $\pm 0.5\text{m}$  für Gebiete unterhalb von ca. 2000 m.ü.M. und  $\pm 1$  bis 3m für Gebiete oberhalb von ca. 2000 m.ü.M (swisstopo 2018a).

### 3.3.1 Wahl der räumlichen Auflösung

Generell ermöglicht eine feinere räumliche Auflösung eine genauere Abbildung des Geländes und ihrer Parameter (Hengl und Evans 2009). Ist die räumliche Variation der Topographie hoch, was auf das teils sehr raue Gelände des Untersuchungsgebietes dieser Arbeit zutrifft (Kapitel 4.1.3), ist eine feinere räumliche Auflösung nötig (Burrough et al. 2015). Veitinger (2015) nimmt an, dass eine räumliche Auflösung von  $\geq 20\text{m}/\text{Pixel}$  nicht ausreicht, um feinskalige Geländemerkmale wie Felsblöcke, kleine Geländekante oder Rinnen zu adäquat zu erfassen. Die räumliche Auflösung sollte jedoch nicht die Genauigkeit der Höhenwerte des Geländemodells unterschreiten (Hengl und Evans 2009).

In der vorliegenden Arbeit ist die Wahl der räumlichen Auflösung allerdings durch eine weitere Faktor beeinflusst: Es wird nämlich das DHM swissALTI<sup>3D</sup> verwendet, welches das Sommergelände repräsentiert (Kapitel 3.3). Doch im alpinen Gelände wandelt sich das Sommergelände aufgrund der Glättung durch die darüber liegende Schneebedeckung in eine spezifische Winteroberfläche. Wie in Kapitel 2.1.2 beschrieben, kommt es zur Auslösung eines Schneebrett auf Grund eines initialen Bruches der Schwachschicht, meist in den oberen Schichten der Schneedecke. Für die Identifizierung potentieller Anrissgebiete von Lawinen auf Grundlage des Geländemodells ist deshalb die Winteroberfläche mit einer Schneedecke massgebend. Veitinger (2015) untersuchte den Zusammenhang zwischen der Geländerauigkeit im Sommer und der Schneeoberflächenrauigkeit im Winter: Die Korrelation zwischen der Geländerauigkeit und der Schneeoberflächenrauigkeit wird mit zunehmend größerer räumlicher Auflösung immer stärker. Eine starke Korrelation bedeutet einen starken Zusammenhang zwischen Sommer- und Winteroberfläche, was für diese Arbeit wünschenswert ist. Bei Schneetiefen von 1 bis 3 m wird eine räumliche Auflösung von mindestens 5 bis 10m pro Pixel empfohlen. Unter Berücksichtigung dieser Empfehlung kann ein Sommer-DHM für Modellierungsanwendungen in einem Wintergelände verwendet werden (Veitinger 2015).

Eine räumliche Auflösung der Sommeroberfläche von 5 Metern hat sich als guter Kompromiss erwiesen, welcher feinskalige Geländeobjekte wie Rinnen oder Kanten noch gut abbildet und gemäss Veitinger (2015) trotzdem bereits mit einer winterlichen Schneeoberfläche korreliert (Bühler et al. 2017). Infolgedessen wird für die Anwendung der Algorithmen aus Kapitel 3.1.2 das swissALTI<sup>3D</sup> mit der *Resampling*-Methode *bilinear* (ESRI 2018a) auf eine räumliche Auflösung von 5m skaliert.

## 4 Methodik

Jeder der in Kapitel 3.1.2 beschriebenen Algorithmen versucht, basierend auf dem Gelände, zu prognostizieren, wo sich potentielle Lawinenanrissgebiete befinden. Der Begriff der Prognose, gebräuchlich in den Atmosphärenwissenschaften (Wilks 2011b), wird hier aber ebenfalls für das Endprodukt der Algorithmen verwendet. Das Hauptziel dieser Arbeit (Kapitel 1.1) ist die Evaluierung der in Kapitel 3.1.2 beschriebenen Algorithmen. Der Evaluierungsprozess beinhaltet die Kalibrierung, Validierung und den Vergleich der Algorithmen (Rykiel 1996): Um eine Prognose generieren zu können, verlangen Algorithmen nach Werten für deren Eingabeparameter. Die Kalibrierung umfasst die Schätzung und Anpassung der optimalen Werte der Parameter, um die Übereinstimmung zwischen der Prognose und einem Referenzdatensatz zu verbessern. Dies verlangt nach einem quantitativen Mass zur Beurteilung der Übereinstimmung zwischen Prognose und Referenz. Die Validierung soll zeigen, wie gut ein Algorithmus seinen beabsichtigten Zweck erfüllt. Dafür wird die Prognose ebenfalls mit einem Referenzdatensatz verglichen. Zur Beurteilung der Übereinstimmung der Prognose und der Referenz ist wiederum eine Masszahl nötig. Anschliessend können mit Hilfe einer oder mehrerer Masszahlen die Algorithmen miteinander verglichen werden. Die Kalibrierung, Validierung und der Vergleich der Algorithmen verlangen deshalb ein Verfahren, um Prognosen mit Hilfe einer Masszahl beurteilen zu können (Prognosebeurteilung).

### 4.1 Referenzdatensatz

Für die Evaluierung und den Vergleich der in Kapitel 3.1.2 vorgestellten Algorithmen wird ein Referenzdatensatz benötigt. Auf Grundlage dieses Referenzdatensatzes sollten während einer möglichst langen Beobachtungsperiode möglichst alle vorkommenden Lawinenanrissgebiete räumlich bestimmt werden können. In der Schweiz führen diverse Organisationen und Institutionen, allen voran die Kantone, wie beispielsweise der Kanton Graubünden (2010), Ereigniskataster diverser Naturgefahrenprozesse. Im Allgemeinen ist jedoch zu sagen, dass wenige flächendeckende Aufzeichnungen mit genügender Kartierungsqualität vergangener Lawinenereignisse existieren. Besonders schwierig sind Referenzdaten mit explizit auskartierten Lawinenanrissgebieten zu finden. Dies ist wohl zum einen damit zu erklären, dass sich Lawinen und vor allem deren Anrissgebiete in alpinen, schwer zugänglichen Gebieten befinden. Zum anderen sind die räumlichen Ausmasse einer Lawine, besonders das Anrissgebiet, beispielsweise wegen Neu- oder Triebsschnee, in relativ kurzer Zeit nach dem Ereignis nur noch schwer erkenn- und damit kartierbar (Bühler et al. 2009).

Obschon anhand hochaufgelöster optischer Fernerkundungsdaten ( $<1.5\text{m}/\text{Pixel}$ ) versucht wurde, Schneelawinen automatisch zu kartieren (Bühler et al. 2009, Lato et al. 2012, Frauenfelder et al. 2015, Eckerstorfer et al. 2016, Korzeniowska et al. 2017), sind bisher die dafür notwendigen Datensätze nicht frei zugänglich und deren Erwerb mit hohen Kosten verbunden. Damit gemeint sind beispielsweise Produkte der Satelliten WorldView, Pléiades oder Spot (Landinfo Worldwide Mapping 2018). Deshalb sind solche Datensätze meist nur für einen bestimmten Perimeter und Zeitpunkt verfügbar und taugen nicht als Referenzdatensatz. Kürzlich konnten aber mit Hilfe von frei verfügbaren Sentinel-1 Radardaten Lawinenablagerungen teilweise kartiert werden (Eckerstorfer et al. 2017). Jedoch war es nicht möglich, aus den Radardaten Anrissgebiete abzulei-

ten. Folglich stellen die im Feld kartierten Lawinenumrisse immer noch die geeignetste Grundlage für einen Referenzdatensatz dar.

#### 4.1.1 Auswahl der Lawinenumrisse

Für die Region Davos (Abb. 4.6) führt das SLF seit dem Winter 1949/50 einen Ereigniskataster, für welchen die Lawinenumrisse durch erfahrene Personen im Feld kartiert werden (Zweifel 2017). Im Grossraum Davos stehen 11798 (Stand 2016) kartierte Lawinenumrisse zur Verfügung. Dieser Kataster gilt gegenwärtig als einer der besten verfügbaren Referenzdatensätze. Mit den Jahren hat die Kartierungsaktivität und -qualität zugenommen (Zweifel 2017). Doch nicht alle dieser Lawinenumrisse sind geeignet, um daraus die für diese Arbeit benötigten Lawinenanrissgebiete abzuleiten. Ein massgeschneidertes Werkzeug (*AvalancheOutlineFilterTool*) wurde von Harvey et al. (2018a) entwickelt und ist in der Lage, geeignete Lawinenumrisse auszuwählen (Harvey et al. 2018b). Ziel ist es, möglichst viele Lawinenumrisse auszuwählen, aber dennoch zu gewährleisten, dass wenig aussagekräftige und zu vernachlässigende Lawinen, beispielsweise Gleitschneelawinen, eliminiert werden. Folgende Filter werden durch das Tool unterstützt (Tab. 4.1):

Tab. 4.1: *AvalancheOutlineFilterTool* mit einer Beschreibung und den für diese Arbeit verwendeten Filtereinstellungen

Filter	Beschreibung	Einstellung
<i>minYear</i>	Erforderliches Mindestjahr des Lawinenumrisses	1970
<i>minSize</i>	Erforderliche Mindestfläche des Lawinenumrisses	1000 m <sup>2</sup>
<i>maxSize</i>	Erlaubte Maximalfläche des Lawinenumrisses	Keine Obergrenze
<i>minDigitQuality</i>	Erforderliche Digitalisierungsqualität: Die Tatsache, dass viele der Lawinenumrisse mit sehr wenigen Stützpunkten digitalisiert wurden, deutet darauf hin, dass die Digitalisierungsqualität ungenügend sein kann. Die Digitalisierungsqualität wurde mit folgender Formel abgeschätzt und muss mindestens den Wert 40 erreichen: $\text{Digitalisierungsqualität (DQ)} = \frac{100n}{\sqrt{l}}$ <i>n</i> : Anzahl Stützpunkte des Lawinenumrisses <i>l</i> : Umfang des Lawinenumrisses	40
<i>maxIntersection</i>	Maximal erlaubte Flächenüberlappung: Etliche der Lawinenumrisse überlappen sich bzw. sind deckungsgleich. Um Redundanz zu vermeiden, kann es sinnvoll sein, von stark überlappenden Lawinenumrisse nur denjenigen mit der besten Digitalisierungsqualität auszuwählen. Um möglichst wenig Daten zu verlieren, wird im vorliegenden Fall jedoch auf diese Möglichkeit verzichtet, d.h. auch eine maximale Flächenüberlappung von 1 wird zugelassen	1
<i>avalancheTriggeringType</i>	Typ der Auslösung: Die für die Schneebrettauslösung notwendige Zusatzlast kann verschiedene Ursachen haben	Natürlich Menschlich Sprengung Unbekannt
<i>reportType</i>	Art des Reports, wobei zwischen Einzellawinen- und Mehrfachlawinenerfassungen unterschieden wird. Im vorliegenden Fall werden alle Lawinen zugelassen. Bei Mehrfachlawinenerfassungen wird das entsprechende Feature für jedes erfasste Polygon dupliziert	Alle
<i>avalanche-TypeExcluded</i>	Eliminierung der Umrise bezüglich Lawinengleitfläche, Lawinenbruch, Lawinenart	Staublawine, Bodenlawine, Gleitschneelawine, Lockerschneelawine



Von den total rund 12000 kartierten Lawinenumrissen blieben nach der Filterung noch deren 5830 kartierte Lawinenumrisse als Referenzdatensatz übrig. Der nächste Schritt besteht darin, aus den Lawinenumrissen deren Anrissgebiete abzuleiten.

#### 4.1.2 Ableitung der Anrissgebiete aus den Lawinenumrissen

In einem internen Projekt des SLFs wurde ein Algorithmus von Harvey et al. (2018a) entwickelt, um aus einem Lawinenumriss dessen Anrissgebiet zu berechnen. Dieser Algorithmus wird auf die oben beschriebenen 5830 Lawinenumrisse angewendet. Die Anrissgebietsbestimmung erfolgt in fünf Schritten gemäss (Harvey et al. 2018a):

##### 1. Umrissunterteilung

Der Lawinenumriss wird mit Hilfe von Höhenlinien in einzelne Sektionen unterteilt. Jede Sektion umfasst einen Höhenunterschied von 10 m (Abb. 4.1).

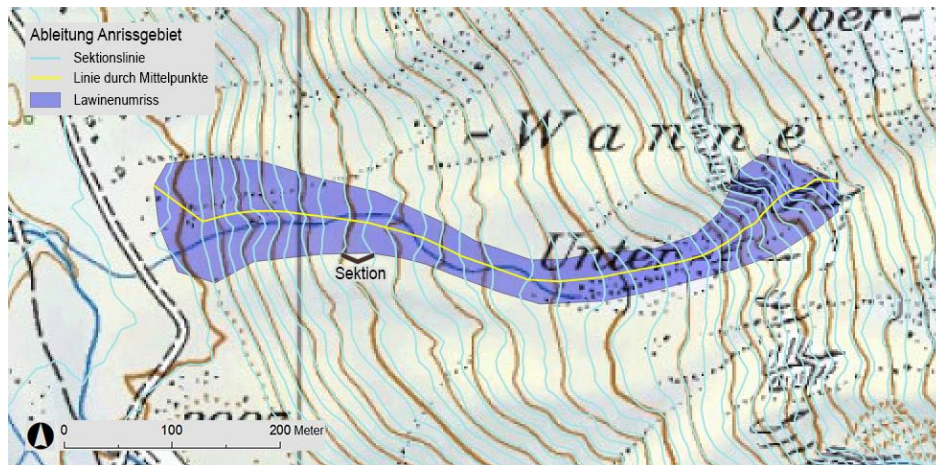


Abb. 4.1: Unterteilung des Lawinenumrisses in einzelne Sektionen (Harvey et al. 2018a)

##### 2. Bestimmung der Lawinenlänge und Lawinenbreite

Die Lawinenrohlänge entspricht der Länge der gelben Mittellinie (Abb. 4.1). Die Mittellinie führt jeweils durch den Mittelpunkt aller einzelnen Höhenlinien. Die Lawinenbreite ergibt sich durch die durchschnittliche Länge der Höhenlinie innerhalb des Lawinenumrisses. Nun wird eine alternative Lawinenlänge bestimmt, indem die Fläche des Lawinenumrisses durch dessen Breite dividiert wird. Das arithmetische Mittel zwischen der alternativen Lawinenlänge und der Rohlawinenlänge wird als Lawinenlänge bezeichnet.

##### 3. Bestimmung der relativen Anrisslänge bezogen auf den Lawinenumriss

Die relative Anrisslänge (Gl. 4.2) wird mit Hilfe eines Grössenverhältnisses (Gl. 4.1) bestimmt:

$$\text{Grössenverhältnis} = \frac{\text{Lawinenbreite}}{\text{Lawinenlänge}} \quad \text{Gl. 4.1}$$

Dabei werden die folgenden drei Fälle unterschieden:

- Ist das Grössenverhältnis (Gl. 4.1) kleiner als 0,2, so wird die gesuchte relative Anrisslänge bei 25% der gesamten Lawinenlänge (0,25) festgelegt. Die Form des Lawinenumrisses ist schmal und langezogen (Abb. 4.2 (a)).
- Ist das Grössenverhältnis (Gl. 4.1) grösser als 2,0, so wird die gesuchte relative Anrisslänge bei 75% der gesamten Lawinenlänge (0,75) festgelegt. Die Form des Lawinenumrisses ist breit, aber kurz (Abb. 4.2 (b)).
- Liegt das Grössenverhältnis (Gl. 4.1) zwischen 0,2 und 2,0, so wird die gesuchte relative Anrisslänge gemäss Gl. 4.2 linear interpoliert.

$$\text{relative Anrisslänge} = \frac{5}{18} \times \text{Grössenverhältnis} + \frac{7}{36} \quad \text{Gl. 4.2}$$

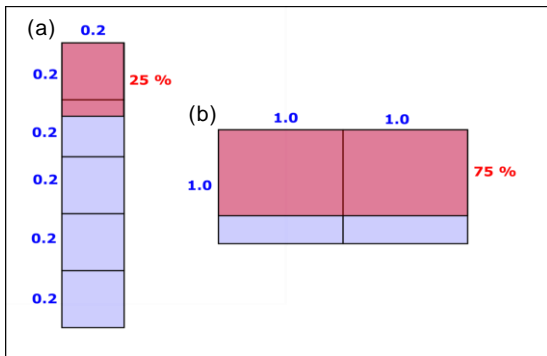


Abb. 4.2: Ableitung des relativen Längenverhältnisses einer schmalen, langgezogenen (a) und einer breiten, kurzen (b) Lawine (Harvey et al. 2018a)

Mit der relativen Anrisslänge kann nun die absolute Anrisslänge bestimmt werden (Gl. 4.3):

$$\text{absolute Anrisslänge} = \text{Lawinenlänge} \times \text{relative Anrisslänge} \quad \text{Gl. 4.3}$$

Mit Hilfe der absoluten Anrisslänge lässt sich ein primäres Lawinenanrissgebiet ableiten (Abb. 4.3).

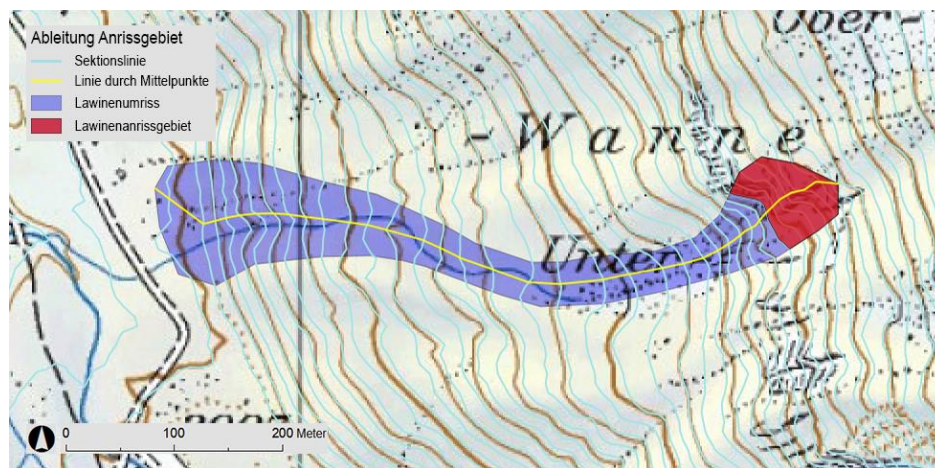


Abb. 4.3: Primäres Anrissgebiet (Harvey et al. 2018a)

#### 4. Bestimmung des sekundären Anrissgebietes

Die kritische Neigung, ob ein Schneebrett nach dem Bruch der Schwachschicht abrutscht oder nicht, liegt bei rund  $30^\circ$  Hangneigung (Kapitel 2.1.2). Deswegen wird das primäre Anrissgebiet mit derjenigen Fläche verschnitten, die eine Neigung von mindestens  $30^\circ$  aufweist. Daraus entsteht das sogenannte sekundäre Lawinenanrissgebiet. In Abb. 4.4 wird sichtbar, wie Bereiche mit einer Neigung kleiner als  $30^\circ$  wegfallen.

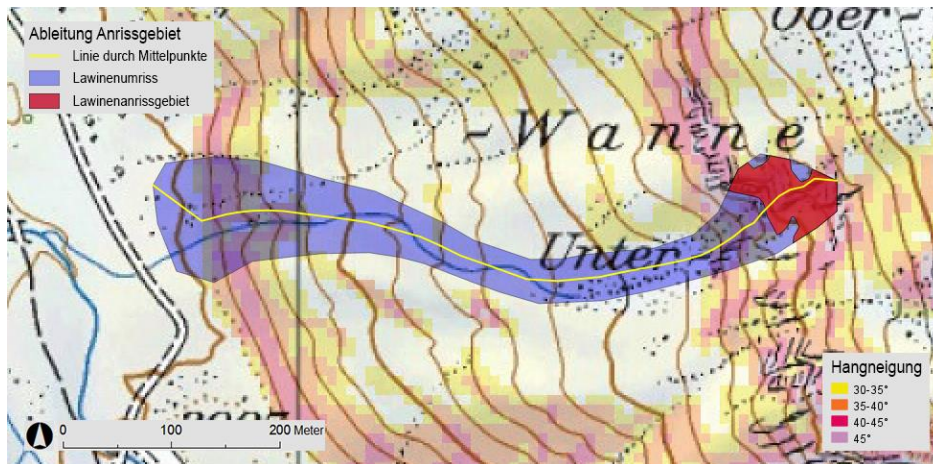


Abb. 4.4: Sekundäres Anrissgebiet (Harvey et al. 2018a)

#### 5. Bereinigung von Digitalisierungsfehlern

Digitalisierungsfehler betreffen vorwiegend die Ränder von Lawinenumrissen. Aus diesem Grund wird ein Puffer von 10% der Lawinenbreite vom sekundären Anrissgebiet abgezogen. Eine maximale Pufferbreite wird für 30 m begrenzt. Abb. 4.5 zeigt die Wirkung des Puffers.

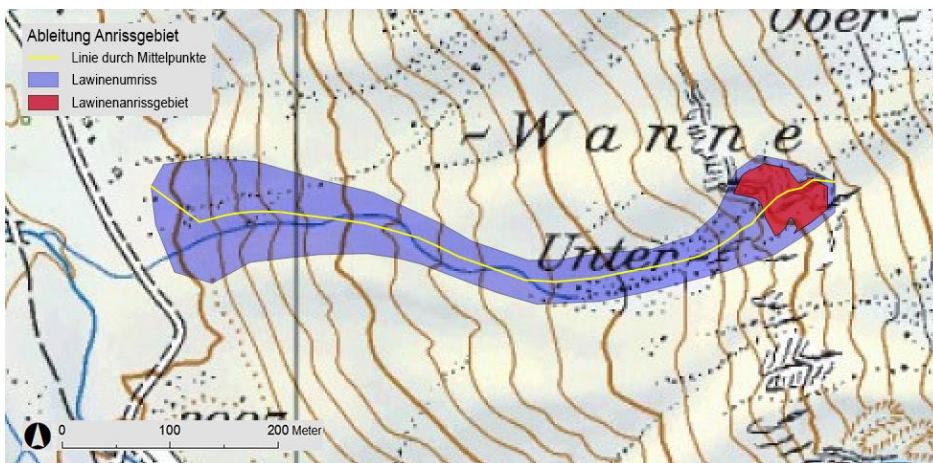


Abb. 4.5: Tertiäres Anrissgebiet (Harvey et al. 2018a)

Von den 5830 kartierten Lawinenumrissen blieben nach diesem Verfahren noch 5785 abgeleitete Anrissgebiete übrig. Für 45 Lawinenumrisse wurde kein Anrissgebiet abgeleitet, weil diese Gebiete

eine Hangneigung unter  $30^\circ$  aufweisen und/oder durch den Puffer noch zusätzlich weiter beschnitten wurden.

Aus diesem Referenzdatensatz mit den Anrissgebieten wurden für die Region Davos drei Untersuchungsgebiete ausgewählt: Parsenn, Jakobshorn und Rinerhorn. Diese drei Gebiete befinden sich in den drei grössten Skigebieten von Davos. Deshalb stehen diese Gebiete im Winterbetrieb mehr oder weniger ständig unter Beobachtung des zuständigen Pisten- und Rettungsdienstes. Lawineneignisse, natürlich oder künstlich ausgelöst, werden von Skipatrouillen kartiert und in die SLF-Datenbank aufgenommen. Der Perimeter der Testgebiete beschränkt sich auf Geländeabschnitte, die vom Skigebiet aus gut beobachtet werden können und schliesst Gebiete aus, in denen Lawinenebeobachtungen schwierig sind. Es ist daher davon auszugehen, dass mit grosser Wahrscheinlichkeit alle potenziellen Lawinen mindestens einmal kartiert wurden. Eine Überprüfung des Referenzdatensatzes mit den lokalen Pisten- und Rettungsdiensten zeigte jedoch, dass bestimmte Anrissgebiete noch nicht im Referenzdatensatz enthalten sind. Darum wurden diese in den Datensatz aufgenommen, um eine möglichst vollständige Referenz zu erhalten. Ebenfalls wurden teilweise Bereiche, die wegen des  $30^\circ$ -Schwellenwertes wegfielen, wieder ergänzt.

#### 4.1.3 Beschreibung der Untersuchungsgebiete

Das Untersuchungsgebiet Parsenn ist mit einer Fläche von  $7.3 \text{ km}^2$  das grösste und erstreckt sich mit einer mittleren Höhe von  $2464 \text{ m ü.M.}$  von  $2197 \text{ m ü.M.}$  bis  $2830 \text{ m ü.M.}$  (Abb. 4.7(a)). Für dieses Gebiet beträgt die mittlere Hangneigung  $25.2^\circ$  (Abb. 4.7(b)) und es wurden 1382 einzelne Anrissgebiete kartiert. Im südlichen Teil wird ein Geländeabschnitt weggelassen, da jener vom Skigebiet aus nicht gut sichtbar und daher nicht genügend gut dokumentiert ist.

Das Untersuchungsgebiet Jakobshorn hat eine Fläche von  $2.0 \text{ km}^2$  und reicht von  $2309 \text{ m ü.M.}$  bis  $2678 \text{ m ü.M.}$  mit einer mittleren Höhe von  $2448 \text{ m ü.M.}$  (Abb. 4.7(a)). Die mittlere Hangneigung beträgt  $27.5^\circ$  und ist damit leicht steiler als Parsenn (Abb. 4.7(b)). Das Gelände ist mehrheitlich sanft und es fehlen steile Felswände, was sich in einer etwas tieferen Geländerauigkeit zeigt (Abb. 4.7(d)). Total enthält der Referenzdatensatz 309 einzelne Anrissgebiete für das Gebiet Jakobshorn.

Das Untersuchungsgebiet Rinerhorn reicht von  $2183 \text{ m ü.M.}$  bis  $2906 \text{ m ü.M.}$  und weist eine

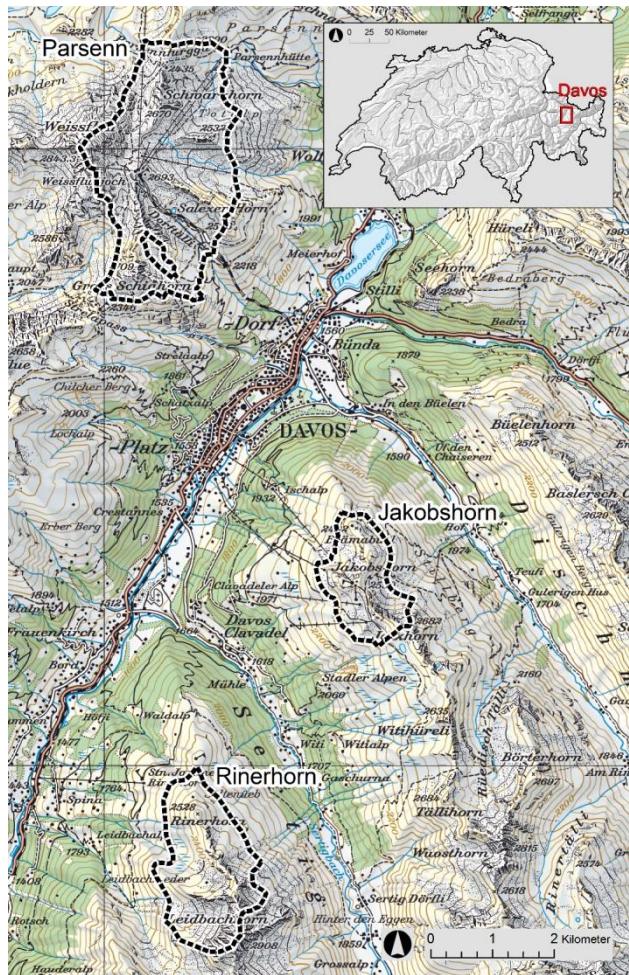


Abb. 4.6: Region Davos mit den drei Gebieten Parsenn, Jakobs- und Rinerhorn [#]

mittlere Höhe von 2443 m ü.M. (Abb. 4.7(a)) sowie eine mittlere Hangneigung von  $28.9^\circ$  auf (Abb. 4.7(b)). Es ist das steilste Testgebiet mit vielen rauen Felsabschnitten, was sich in einer höheren Geländeraugigkeit ausdrückt (Abb. 4.7(d)). Es zählt 438 einzelne Anrissgebiete.

Alle drei Untersuchungsgebiete weisen durchschnittliche Plankurvaturwerte von  $0.06 / 100\text{m}$  auf (Abb. 4.7(c)). Total umfassen alle drei Gebiete insgesamt 2129 einzelne PAG. Im Perimeter der drei Gebiete befindet sich kein Wald, weshalb die Rolle des Waldes im Zusammenhang mit der Auslösung von Lawinen nicht weiter berücksichtigt wird.

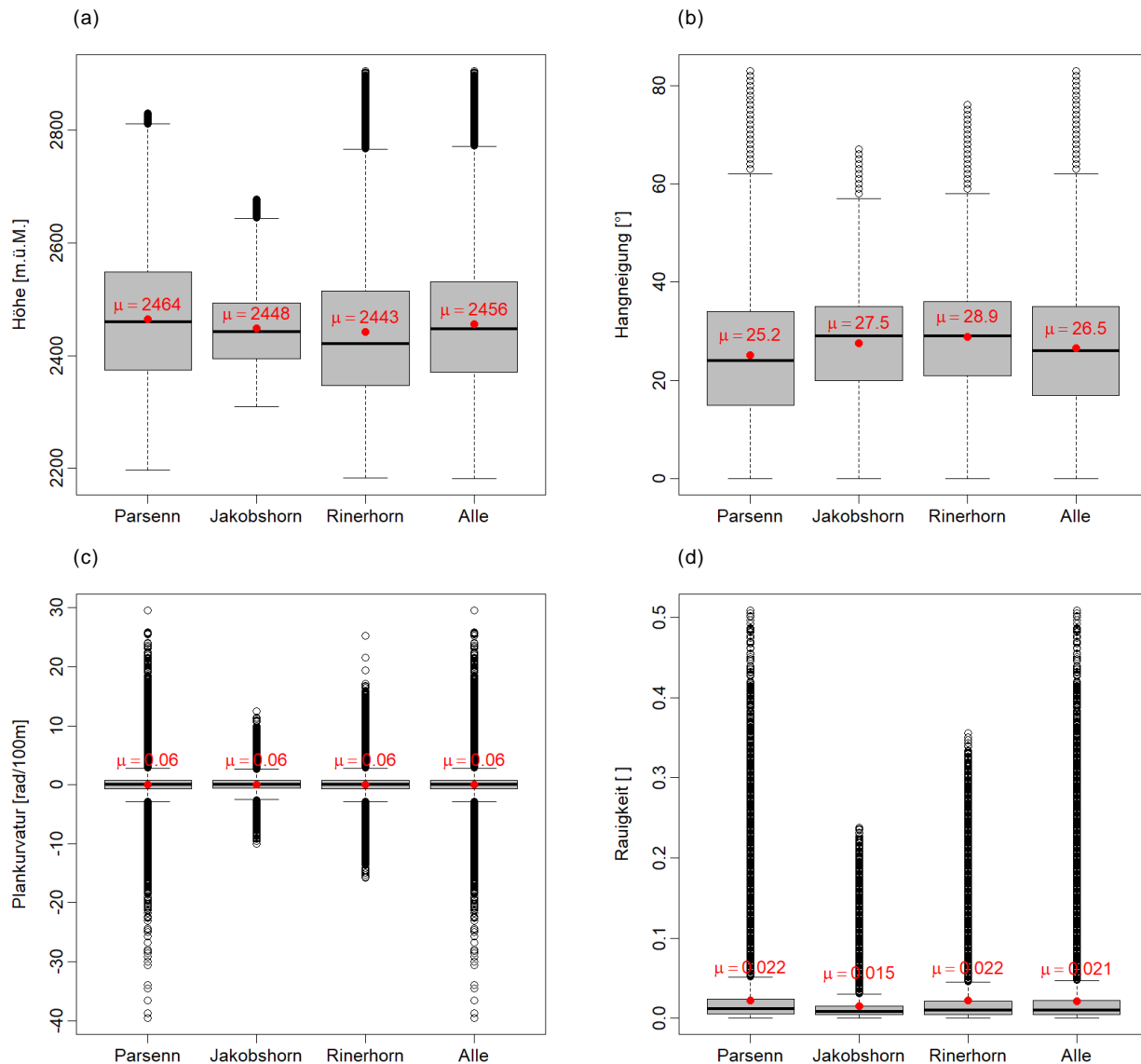
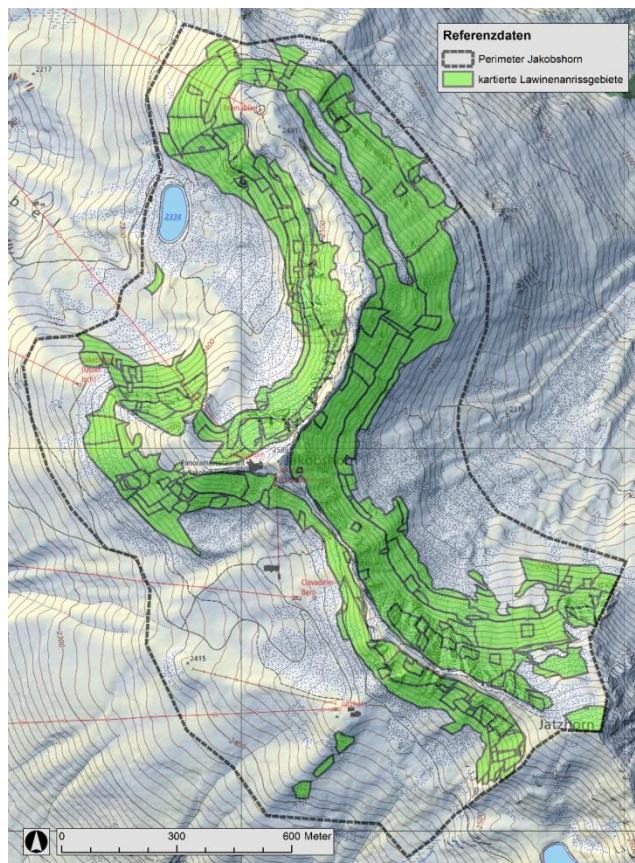


Abb. 4.7: Verteilungen der geländespezifischen Eigenschaften (Höhe über Meer (a), Hangneigung (b), Plankurvatur (c) und Rauigkeit (d)) der drei Untersuchungsgebiete Parsenn, Jakobs- und Rinerhorn [\*]

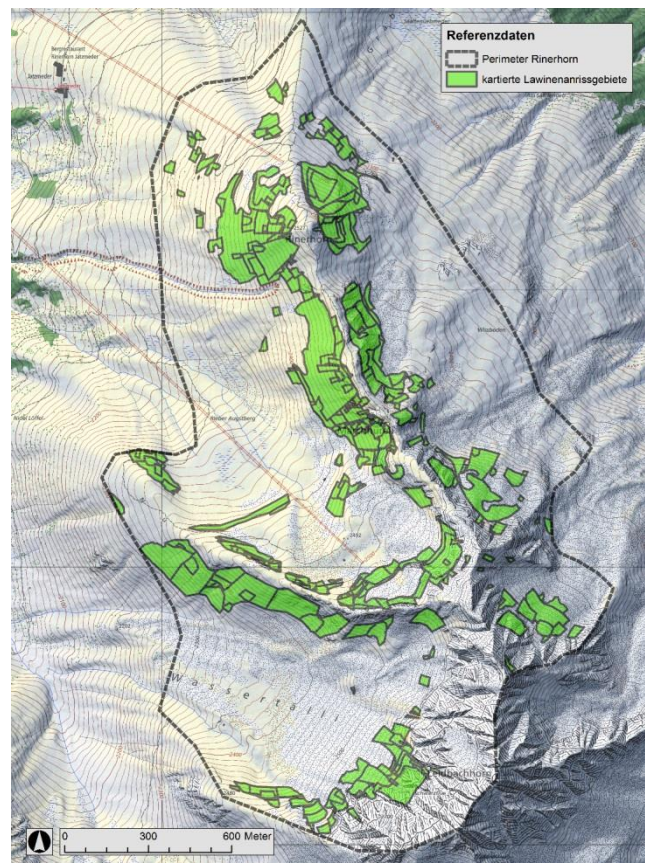
#### 4.1.4 Beschreibung der Anrissgebiete des Referenzdatensatzes

Im Vergleich zu den durchschnittlichen Höhen der Untersuchungsgebiete von 2456m.ü.M. (Abb. 4.7(a)) ist die durchschnittliche Höhe der Anrissgebiete mit 2487m.ü.M. leicht höher (Abb. 4.9(a)). Bezogen auf die Anrissgebiete verfügt das Gebiet Parsenn (Abb. 4.8(c)) mit  $37.6^\circ$  über die höchste durchschnittliche Hangneigung (Abb. 4.9(b)). Ausserdem weisen die Anrissgebiete von Parsenn im Mittel eine leicht konvexe Form auf. Dagegen sind die Anrissgebiete des Jakobshorns (Abb. 4.8(a)) und Rinerhorns (Abb. 4.8(b)) leicht konkav. Über alle drei Gebiete sind sie im Durchschnitt jedoch planar (Abb. 4.9(c)). Ebenfalls zeigen die Anrissgebiete Parsenn im Vergleich zum Jakobs- und Rinerhorn im Mittel höhere Rauigkeitswerte (Abb. 4.9(d)). Die Unterschiede zwischen den verschiedenen Gebieten sind bezüglich Hangneigungs-, Plankurvatur- und Geländerauigkeitswerten nur minim. Generell bestätigen also die Verteilungen der Hangneigungs-, Plankurvatur- und Geländerauigkeitswerte die Annahmen zu den geländespezifischen, lawinenbildenden Faktoren in Kapitel 2.1.2. Des Weiteren fällt auf, dass die Anrissgebiete Hangneigungen unter  $30^\circ$  aufweisen. Das ist mit den manuellen Ergänzungen durch die lokalen Pisten- und Rettungsdiensten zu erklären.

(a)



(b)



(c)

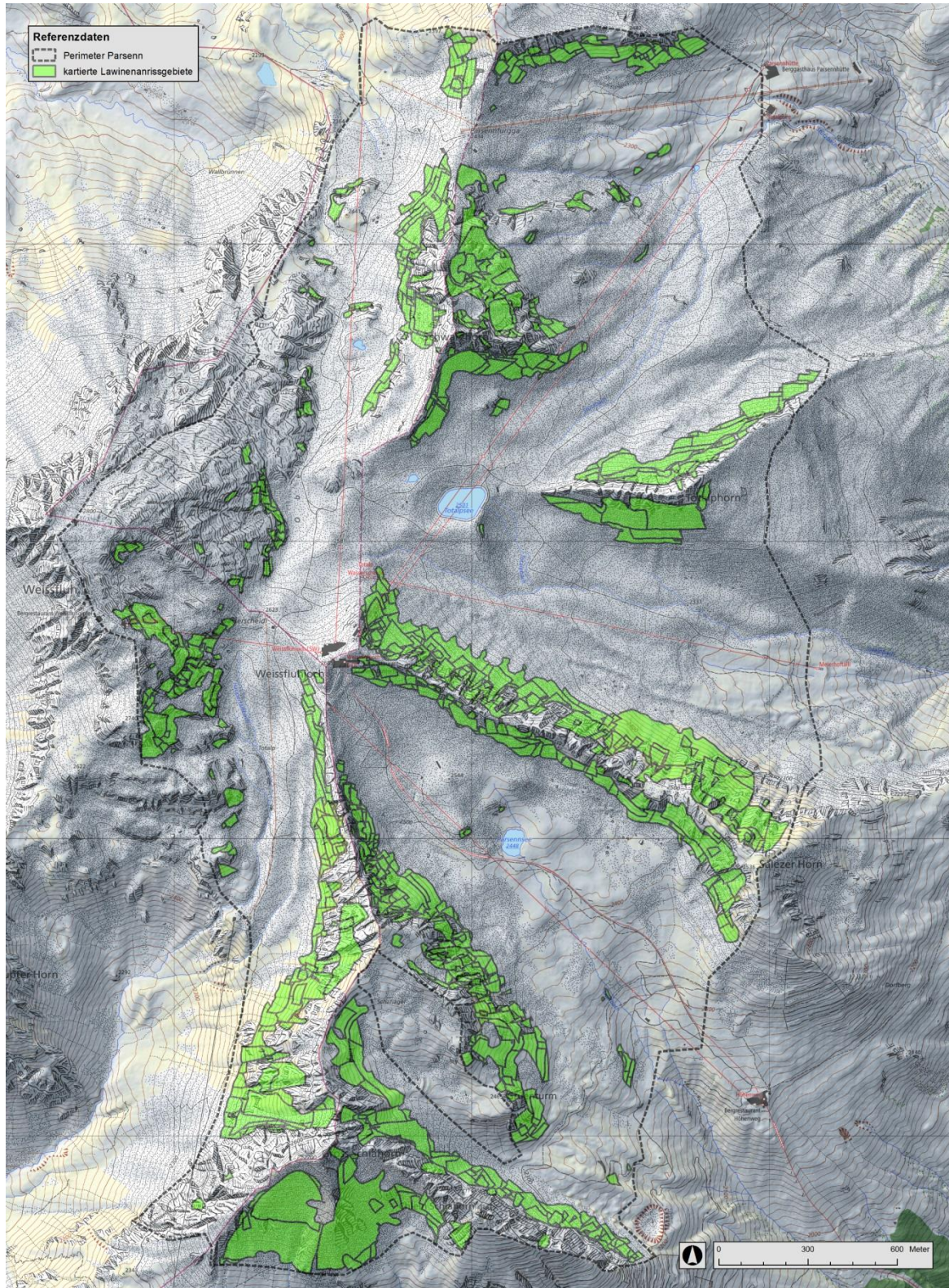


Abb. 4.8: Lawinenanrissgebiete des Referenzdatensatzes für das Gebiet Jakobshorn (a), Rinerhorn (b) und Parsenn (c) [#]

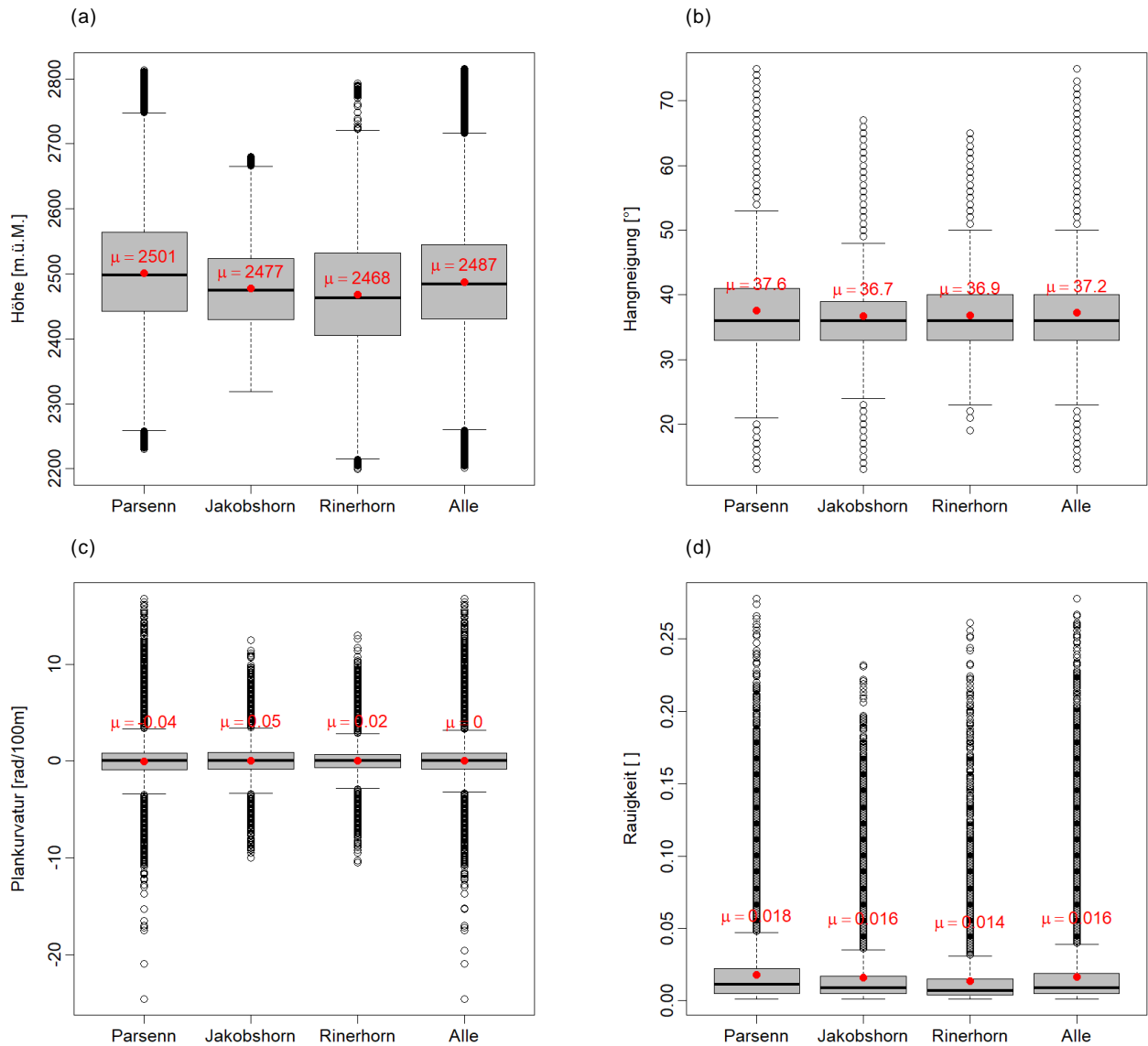


Abb. 4.9: Verteilungen der geländespezifischen Eigenschaften (Höhe über Meer (a), Hangneigung (b), Plankurvatur (c) und Rauigkeit (d)) der Anrissgebiete der drei Untersuchungsgebiete Jakobshorn, Parsenn und Rinerhorn mit 5m räumlicher Auflösung [°]

## 4.2 Prognosebeurteilung

In ganz verschiedenen Disziplinen wurden Verfahren entwickelt, um die Qualität einer Prognose zu beurteilen. Deshalb existieren auch unzählige Masszahlen, wobei für dieselbe Masszahl manchmal unterschiedliche Begrifflichkeiten verwendet werden. Allen Verfahren ist jedoch gemeinsam, dass die Beurteilung auf dem Verhältnis von Prognose und Referenz basiert. (Wilks 2011b). Für die Evaluation muss die Prognose als auch die Referenz in demselben Datentyp vorliegen. Der vorliegende Datentyp begrenzt dabei die anwendbaren Evaluationsverfahren (Veitinger 2015).



### 4.2.1 Prognosebeurteilung für binäre Klassifikationen (*Boolean*)

Die vier in Kapitel 3.1.2 ausgewählten Algorithmen unterteilen das Gebiet innerhalb eines bestimmten Perimeters in die beiden Klassen Potentielles Anrissgebiet (PAG) und nicht Potentielles Anrissgebiet (nPAG). Die Prognose der Algorithmen kann als thematische Karte mit den beiden diskreten Klassen PAG und nPAG aufgefasst werden. Deswegen wird für die Prognoseverifikation eine Konfusionsmatrix berechnet, wie sie auch für thematische Kartierungen in der Fernerkundung benutzt wird (Stehman 1997, Congalton und Green 2009). Als Referenzdatensatz werden die Anrissgebiete aus Kapitel 4.1 genutzt. Zur Erstellung der Konfusionsmatrix werden dabei die PAG- und nPAG-Flächen der Prognose des Algorithmus mit den PAG- und nPAG-Flächen des Referenzdatensatzes verschnitten. Diese Teilflächen werden in der Konfusionsmatrix (Abb. 4.10) durch die Werte a, b, c und d (Gl. 4.4 bis Gl. 4.7) repräsentiert. Das Gesamttotal aller Teilflächen wird mit n (Gl. 4.8) bezeichnet.

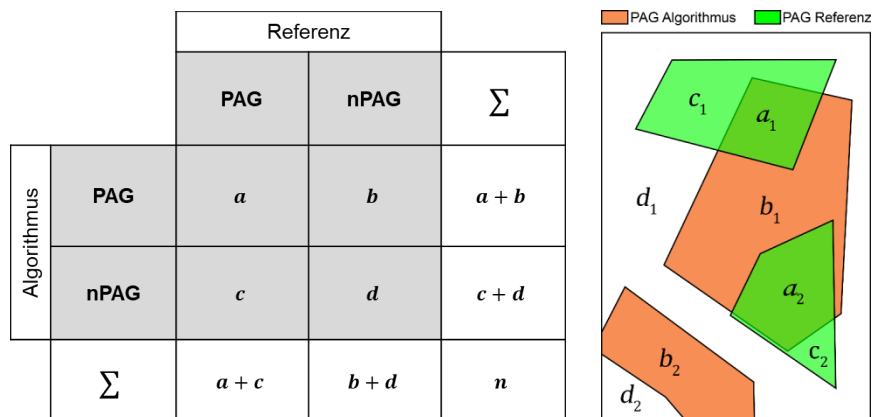


Abb. 4.10: Schematische Darstellung einer Konfusionsmatrix nach Congalton und Green (2009)

$$a = \sum_{i=1}^n a_i = a_1 + a_2 + \dots + a_n \quad \text{Gl. 4.4}$$

$$b = \sum_{i=1}^n b_i = b_1 + b_2 + \dots + b_n \quad \text{Gl. 4.5}$$

$$c = \sum_{i=1}^n c_i = c_1 + c_2 + \dots + c_n \quad \text{Gl. 4.6}$$

$$d = \sum_{i=1}^n d_i = d_1 + d_2 + \dots + d_n \quad \text{Gl. 4.7}$$

$$n = (a + c) + (b + d) = (a + b) + (c + d) = a + b + c + d \quad \text{Gl. 4.8}$$

Wie in Kapitel 1.1 beschrieben, sollen die Algorithmen zur automatisierten PAG-Identifizierung auf die Verwendung von dynamischer Lawinensimulationssoftware ausgelegt sein. Für die meisten modernen, dynamischen Lawinensimulationsprogramme werden als Eingangsdaten Anrissgebiete als diskrete Vektorobjekte benötigt. Zudem liegt der Referenzdatensatz ebenfalls als Vektorobjekt vor. Deswegen geben auch die Algorithmen die Anrissgebiete als Vektorobjekte aus. Folglich sind die für die Konfusionsmatrix berechneten Flächenwerte Quadratmeter und nicht die Anzahl der Pixel, wie es für die Konfusionsmatrix eher üblich wäre (Stehman und Wickham 2011). Mit Hilfe der Konfusionsmatrix können verschiedene Masszahlen abgeleitet werden, um eine Prognose auf verschiedene Aspekte hin zu beurteilen.

### Aspekt der Genauigkeit

Der Aspekt der Genauigkeit bezieht sich auf die durchschnittliche Übereinstimmung zwischen einer Prognose, simuliert von einem Modell, und einem Referenzdatensatz. Genauigkeitsmasse versuchen die Gesamtqualität einer Prognose in einer einzigen Zahl zusammenzufassen (Wilks 2011b).

Das vielleicht unmittelbarste und intuitivste Mass für die Genauigkeit für diskrete Klassierungen ist der Anteil korrekt prognostizierter Flächen  $a$  und  $d$  am Gesamttotal  $n$  (Gl. 4.9). Dieses Genauigkeitsmass erfüllt das Prinzip der Gleichwertigkeit, da korrekt prognostizierte PAG- und nPAG-Flächen gleichermaßen gewichtet werden. Im Falle von sehr wenigen beobachteten Ereignissen im Referenzdatensatz  $a+c$  (Abb. 4.10) ist dieses Mass aber mit Vorsicht zu bewerten, weil dann die Prognose von nPAG-Flächen  $d$  (Abb. 4.10) mit grosser Wahrscheinlichkeit korrekt ist und deshalb leicht zu prognostizieren sind. Der Wertebereich dieses Masses reicht von 0 bis 1, wobei 0 das schlecht- und 1 das bestmögliche Resultat darstellt. (Wilks 2011b). Als Synonym für das Mass *Proportion Correct* wird auch der Begriff *Overall Accuracy* verwendet (Allouche et al. 2006). Durch das Multiplizieren mit dem Faktor 100% erhält man einen Prozentsatz, welcher im Englischen als *Percent Correct* oder *Percentage of Forecasts Correct* bezeichnet wird (Wilks 2011b).

$$\text{Percent Correct (PC)} = \frac{a + d}{n} \quad \text{Gl. 4.9}$$

### Aspekt der Unterscheidungsfähigkeit

Der Aspekt der Unterscheidungsfähigkeit beschreibt die Fähigkeit, wie gut ein Modell, Untersuchungsobjekte, in diesem Fall Flächen, in die richtige Klasse einteilen kann. Hierbei werden die folgenden zwei Masszahlen näher beschrieben (Beguería 2006, Wilks 2011b):

POD (Erkennungsrate) ist gleich dem Anteil, der durch den Algorithmus korrekt identifizierten PAG Flächen  $a$  an der PAG-Fläche des Referenzdatensatzes  $a+c$  (Gl. 4.10). Dies entspricht der Wahrscheinlichkeit, dass, gegeben eine bestimmte Fläche ist im Referenzdatensatz als PAG vermerkt, diese durch den Algorithmus korrekt als PAG-Fläche klassiert wird (Wilks 2011b). Im Englischen wird neben dem Begriff *Probability of Detection* auch der Begriff *Hit Rate* verwendet. Der Wertebereich

reich dieses Masses reicht von 0 bis 1, wobei 0 das schlechteste- und 1 das bestmögliche Resultat darstellen (Gariano et al. 2015).

$$\text{Probability of Detection (POD)} = \frac{a}{a + c} \quad \text{Gl. 4.10}$$

POFD (Falscherkennungsrate) ist gleich dem Anteil der durch den Algorithmus falsch identifizierten PAG-Flächen  $b$  an der nPAG-Fläche des Referenzdatensatzes  $b+d$  (Gl. 4.11). Dies entspricht der Wahrscheinlichkeit, dass, gegeben eine bestimmte Fläche ist im Referenzdatensatz als nPAG vermerkt, diese durch den Algorithmus fälschlicherweise trotzdem als PAG-Fläche klassiert wird (Wilks 2011b). Der Wertebereich dieses Masses reicht von 0 bis 1, wobei 1 das schlecht- und 0 das bestmögliche Resultat darstellen (Gariano et al. 2015).

$$\text{Probability of False Detection (POFD)} = \frac{b}{b + d} \quad \text{Gl. 4.11}$$

### Aspekt der Prognosefähigkeit

Der Aspekt der Prognosefähigkeit bezieht sich auf die relative Genauigkeit einer bestimmten Prognose im Verhältnis zur Genauigkeit einer Referenzprognose. Zur Erstellung von Referenzprognosen können verschiedene Methoden, beispielsweise ein Zufallsklassifikator, verwendet werden. Masse, welche die Prognosefähigkeit quantifizieren, werden im Englischen als *Skill Scores* bezeichnet (Wilks 2011b). Im Allgemeinen wird zur Berechnung des *Skill Score* das erzielte Genauigkeitsmass einer bestimmten Prognose in Bezug zum Genauigkeitsmass einer Referenzprognose gesetzt (Gl. 4.12). Als ein mögliches Mass für die Genauigkeit dient dabei beispielsweise *Percent Correct* (Gl. 4.9) und als Referenzprognose kann beispielsweise ein *Random Forecast* (Gl. 4.13) verwendet werden. Zusätzlich wird  $A_{\text{perf}} = 1$  (Gl. 4.12) benötigt, definiert als Genauigkeitsmass, welches durch eine perfekte Prognose erzielt wird (Wilks 2011b).

*Skill Scores* werden ebenfalls in Prozent angegeben und anschliessend als Prozentsatz interpretiert, um wie viel besser oder schlechter der Algorithmus im Vergleich zu einer Referenzprognose ist. Falls  $A = A_{\text{perf}}$  erreicht der *Skill Score* seinen maximalen Wert von 100%. Falls  $A = A_{\text{ref}}$  beträgt der *Skill Score* 0%, was bedeutet, dass der Algorithmus keine Verbesserung gegenüber der Referenzprognose erzielen konnte. Wenn das Genauigkeitsmass einer bestimmten Prognose schlechter ist als die Referenzprognose, ist der *Skill Score* kleiner als 0%.

$$\text{Skill Score (SS)} = \frac{A - A_{\text{ref}}}{A_{\text{perf}} - A_{\text{ref}}} \quad \text{Gl. 4.12}$$

$A$ : erzieltes Genauigkeitsmass eines bestimmten Prognose

$A_{\text{ref}}$ : erzieltes Genauigkeitsmass einer Referenzprognose

$A_{\text{perf}}$ : erzieltes Genauigkeitsmass einer perfekten Prognose

$$\text{Random Forecast (RF)} = \frac{(a + b \times a + c) + (c + d \times b + d)}{n^2} \quad \text{Gl. 4.13}$$

$$\text{unbiased Random Forecast (RF}_{\text{unbiased}}) = \frac{(a + c)^2 + (b + d)^2}{n^2} \quad \text{Gl. 4.14}$$

Einer der am häufigsten verwendeten *Skill Scores* wurde ursprünglich von Doolittle (1888) vorgeschlagen, ist aber allgemein bekannt als *Heidke Skill Score* nach Heidke (1926). Dieser Ansatz wurde von Cohen (1960) wieder aufgegriffen und seitdem wird *Heidke Skill Score* auch als Kappa bezeichnet. Dieses Mass bezüglich der Prognosefähigkeit wurde später durch Congalton et al. (1983) in die Fernerkundung eingeführt. Der *Heidke Skill Score* (Gl. 4.15) wendet als Genauigkeitsmass *Proportion Correct* (Gl. 4.9) an. Für die Referenzprognose wird eine Zufallsprognose gemäss Gl. 4.13 berechnet (Wilks 2011b).

$$\text{Heidke Skill Score (HSS)} = \frac{PC - RF}{1 - RF} \quad \text{Gl. 4.15}$$

Ein weiteres adäquates Mass zur Quantifizierung der Prognosefähigkeit ist der *Peirce Skill Score* (Peirce 1884). Dieses Mass wird ebenfalls als *Hanssen-Kuipers Discriminant* (Hanssen und Kuipers 1965) oder *True Skill Statistics* (Allouche et al. 2006) bezeichnet. Der *Peirce Skill Score* (Gl. 4.16) ist identisch mit dem *Heidke Skill Score* (Gl. 4.15), mit der Ausnahme, dass im Nenner eine andere Referenzprognose (Gl. 4.14) verwendet wird. Umgeformt lässt sich der *Peirce Skill Score* als Differenz zwischen den zwei bedingten Wahrscheinlichkeiten POD (Gl. 4.10) und POFD (Gl. 4.11) definieren. Im Vergleich zum HSS ist der PSS unabhängig von der Prävalenz (Allouche et al. 2006). Die Prävalenz steht für einen unterschiedlichen Anteil der beiden Klassen (hier PAG und nPAG) im Referenzdatensatz (Beguería 2006).

$$\text{Peirce Skill Score (PSS)} = \frac{PC - RF}{1 - RF_{\text{unbiased}}} = \frac{ad - bc}{(a + c)(b + d)} = \text{POD} - \text{POFD} \quad \text{Gl. 4.16}$$

### 4.3 Ablauf der Evaluierung

Am Ende ist das Ziel, die verschiedenen Algorithmen miteinander vergleichen zu können. Für den Vergleich wird pro Algorithmus eine bestimmte Prognose benötigt. Die Algorithmen verlangen für eine Prognose bestimmte Werte für ihre Eingabeparameter. Wie sollen aber die Werte für diese eine Prognose pro Algorithmus gewählt werden? Ziel ist es, die bestmögliche Performance für jeden Algorithmus zu erreichen und sie der bestmöglichen Performance der anderen Algorithmen gegenüberzustellen. Dafür müssen für jeden Algorithmus die optimalen Werte für seine Eingangsparameter eruiert werden.

Wie in Kapitel 4.2 dargelegt, gibt es eine Vielzahl verschiedener Masse, um eine Prognose auf ihre unterschiedlichen Aspekte hin zu bewerten. Zur Ermittlung der optimalen Inputparameter bedarf es jedoch der Maximierung einer einzelnen Masszahl. Das bedeutet, dass die Beurteilung einer Prognose auf ein einzelnes Mass reduziert werden muss und dabei nicht alle Aspekte der Prognosebeurteilung abgebildet werden (Wilks 2006). Dementsprechend ist die Wahl des zu verwendenden Masses herausfordernd.

### 4.3.1 Gütemasse zur Prognosebeurteilung

Wie bereits in Kapitel 1.1 erwähnt, wird in dieser Arbeit die Identifizierung potentieller Anrissgebiete im Hinblick auf automatische Gefahrenhinweiskartierung vorgenommen. Gefahrenhinweiskarten halten flächendeckend fest, wo überall möglicherweise mit welcher Gefahr (in diesem Fall Schneebrettlawinen) gerechnet werden muss (Kapitel 2.2.1). Ein Algorithmus sollte darum einerseits möglichst alle Flächen als PAG klassifizieren, welche im Referenzdatensatz als Anrissgebiete erfasst sind. Andererseits soll er aber auch zuverlässig in der Lage sein, möglichst wenig Fläche als PAG zu klassieren, die im Referenzdatensatz nicht als Anrissgebiete vermerkt ist. Das erste Kriterium wird durch die Erkennungsrate (POD, Gl. 4.10) und das zweite durch die Falscherkennungsrate (POFD, Gl. 4.11) gemessen. Je besser der Algorithmus, desto höher sollte der POD und desto niedriger der POFD sein. Diese zwei Kriterien werden durch den *Peirce Skill Score* (PSS) gleich stark berücksichtigt. Hinsichtlich der Ermittlung der optimalen Parametereinstellung verfügt der PSS über eine weitere entscheidende Eigenschaft. Der PSS gewichtet die korrekte Prognose von weniger häufigen Ereignissen stärker als die korrekte Prognose von häufigeren Ereignissen. Folglich werden die Algorithmen daraufhin trainiert, seltene Ereignisse, obgleich ihres verhältnismässig kleinen Anteils im Referenzdatensatzes, trotzdem zu prognostizieren (Wilks 2006).

Der PSS wird häufig für die Evaluierung von Modellen in der Gefahrenbeurteilung und im Risikomanagement eingesetzt (Beguería 2006), zum Beispiel zur Kalibrierung von Überschwemmungsmodellen (Pappenberger et al. 2007) für die Validierung von Erdbebenanfälligkeitsmodellen (Frattini et al. 2010, Gariano et al. 2015, Leonarduzzi et al. 2017) und Vorhersage der Standorte zukünftiger Erdbeben (Shcherbakov et al. 2010) oder im Zusammenhang mit der Lawinengefahr (Purves et al. 2003).

Zudem wird ein Mass eingeführt, um die jeweils prognostizierte PAG-Fläche in Relation zu einem Flächenrichtwert zu setzen (Gl. 4.17). Die jeweils prognostizierte PAG-Fläche resultiert aus einer Simulation eines bestimmten Algorithmus mit bestimmten Parametereinstellungen ( $Fläche_{Algorithmus}$ ). Als Richtwert dient die Fläche, welche Hangneigungswerte zwischen 28 und 60° aufweist. Dies ist der simple Ansatz nach Voellmy (1955), welcher nur die Hangneigung berücksichtigt ( $Fläche_{Voellmy28-60^\circ}$ ). Im Hinblick auf die regionale Gefahrenhinweiskartierung ist ein Mass für die als PAG ausgeschiedene Fläche eine wichtige Zusatzinformation. Denn je kleiner die als PAG ausgeschiedene Fläche ist, desto besser ist die Eingrenzung, und umso weniger Rechenzeit würde in einem nächsten Schritt für die Lawinensimulation zur Gefahrenhinweiskartierung benötigt.

$$FA = \frac{Fläche_{Algorithmus}}{Fläche_{Voellmy\ 28-60^\circ}} \times 100\%$$

Gl. 4.17

Wie beschrieben in Kapitel 3.1.2 setzt sich der Algorithmus von Bühler et al. (2018) aus den zwei Prozessierungsschritten Vorauswahl und szenarienspezifischer Gebietsabtrennung zusammen. Deswegen werden Gütemasse zur Prognosebeurteilung jeweils separat für die Vorauswahl und die Szenarien aufgelistet. Des Weiteren werden alle aufgelisteten Gütemasse auf die zweite Nachkommastelle gerundet.

### 4.3.2 Trainings- und Testgebiete

Zur Kalibrierung als auch für die Validierung eines Modells werden die in Kapitel 4.1 beschriebenen Referenzdaten benötigt. Das Trainingsgebiet zur Kalibrierung sollte jedoch nicht dasselbe sein wie das Testgebiet zur Validierung (Wilks 2011a). Durch die Kalibrierung werden die optimalen Parametereinstellungen des jeweiligen Algorithmus eruiert und mit diesen Einstellungen sollten die Algorithmen universal einsetzbar sein. Deswegen wird das Gebiet Parsenn als Trainingsgebiet und das Jakobs- und Rinerhorn als Testgebiete dienen. So kann die Prognosefähigkeit der Algorithmen untereinander verglichen und für zukünftige Anwendungen abgeschätzt werden (Wilks 2011a).

### 4.3.3 Kalibrierung

Zur Ermittlung der optimalen Parametereinstellungen für den jeweiligen Algorithmus wird das folgende Vorgehen gewählt: Für den jeweiligen Algorithmus werden die Werte eines Parameters systematisch geändert (beispielsweise 20°, 21°, 22° bis 40° als Untergrenze für die Hangneigung), während alle anderen Eingabeparameter dieses Algorithmus auf einem bestimmten Wert (beispielsweise 60° als Obergrenze für die Hangneigung) fixiert bleiben. Nach der Iteration über diesen Parameter (Untergrenze Hangneigung) wird ein anderer Eingabeparameter (beispielsweise Obergrenze Hangneigung) systematisch geändert, während die Untergrenze für die Hangneigung wieder auf einem Standardwert (beispielsweise 30°) fixiert wird. Die Parametereinstellung der einzelnen Algorithmen wird auf einen möglichst hohen PSS hin optimiert (Heierli et al. 2004). Um die Iteration über alle Eingabeparameter automatisiert vollziehen zu können, wurde pro Algorithmus ein Skript in Python geschrieben (Kapitel 3.2.1). Über eigens erstellte Toolboxes in ArcMap konnten die Skripte mit den nötigen Eingabedaten gestartet werden. Die Standardwerte waren direkt im Tool hinterlegt, während dem die Parameterwerte, über welche iteriert werden sollte, aus einer entsprechend verlinkten Tabelle entnommen wurden. Zur Auswertung und Darstellung der Resultate wurde ein R-Skript geschrieben (Kapitel 3.2.2). Bei Darstellung der Resultate in Kapitel 5 wird für jeden Algorithmus pro Parameter, über welchen iteriert wird, ein x/y-Diagramm erstellt. Abhängig vom Parameterwert (x-Wert) wird der POD, POFD und der PSS (y-Werte) angegeben. Wie in Kapitel 4.2.1 erklärt, werden die Masse bezüglich der Prognosefähigkeit jeweils mit dem Faktor 100% skaliert. Konsequenterweise wird deshalb auch der POD und POFD skaliert. Ebenfalls abhängig vom Parameterwert wird zusätzlich der Flächenanteil Gl. 4.3 angegeben. Die Eingabewerte pro Parameter mit dem maximalen PSS-Wert gelten als optimale Parametereinstellungen.

#### 4.3.4 Validierung und Vergleich

Lässt man nun einen Algorithmus mit den entsprechenden, optimalen Parametereinstellungen laufen, so ergibt dies die bestmögliche Performance, welche mit diesem Algorithmus erreichbar ist. Diese kann dann mit der bestmöglichen Performance der anderen Algorithmen verglichen werden. Das ermöglicht einen objektiven Vergleich der verschiedenen Algorithmen, obwohl sie unterschiedlich aufgebaut sind und verschiedene Eingangsparameter benötigen. Die Resultate werden dabei in Tabellenform zusammengefasst. In der Tabelle werden die Masse POD (Gl. 4.10), POFD (Gl. 4.11), PSS (Gl. 4.16), HSS (Gl. 4.15) und FA (Gl. 4.17) aufgeführt, wobei der PSS als entscheidend angesehen wird. Je höher POD, PSS und HSS, desto besser ist der Algorithmus zu bewerten. Je höher der POFD und FA, desto schlechter ist der Algorithmus einzustufen.

## 5 Resultate

Im Folgenden werden die Algorithmen anhand des Referenzdatensatzes gemäss Kapitel 4.3 kalibriert, validiert und miteinander verglichen. Hierbei dient das Gebiet Parsenn als Trainingsgebiet zur Kalibrierung und das Jakobs- und Rinerhorn als Testgebiet für die Validierung.

### 5.1 Kalibrierung

Die Algorithmen, beschrieben in Kapitel 3.1.2, werden anhand des Trainingsgebiets Parsenn gemäss Kapitel 4.3.3 kalibriert. Der Algorithmus nach Voellmy (1955) musste nicht kalibriert werden, da deren Schwellenwerte von vornherein festgesetzt wurden.

#### 5.1.1 Bühler et al. (2013)

Wie in Abb. 5.1(a) zu erkennen ist, reagiert der Algorithmus von Bühler et al. (2013) ziemlich sensitiv auf eine Änderung des unteren Schwellenwertes für die Hangneigung. Wird er tief gesetzt, resultieren hohe POD-, gleichzeitig aber auch hohe POFD-Werte. Bei hohen Schwellenwerten resultieren tiefe POD-, gleichzeitig aber auch tiefere POFD-Werte. Ein Schwellenwert von  $30^\circ$  weist den maximalen PSS-Wert auf, was für das Optimum zwischen POD und POFD steht. Die y-Achse, auf welcher der Flächenanteil aufgetragen wird, reicht bis 120%. Dies ist damit zu erklären, dass für ganz tiefe Hangneigungswerte ( $<28^\circ$ ) der Algorithmus Bühler et al. (2013) mehr Fläche als PAG ausweist als Voellmy (1955)  $28-60^\circ$ .

Anders verhält sich der obere Schwellenwert für die Hangneigung (Abb. 5.1(b)): Bis rund  $60^\circ$  ist eine Änderung des POD-, POFD- und der FA-Wertes noch erkennbar, ab rund  $60^\circ$  jedoch fast nicht mehr.

Für die Grösse des Zellfensters zur Berechnung der Rauigkeit zeigt sich bezüglich des PSS-Wertes ein schwaches Maximum bei einem Zellfenster von  $9 \times 9$  Pixeln (Abb. 5.1(c)).

Bezüglich der Rauigkeit (Abb. 5.1(d)) und der Plankurvatur (Abb. 5.1 (e)) ist folgendes erkennbar: Eine Änderung des Schwellenwertes im tiefen Bereich hat eine starke Änderung der Validierungswerte zur Folge. Eine Änderung des Schwellenwertes im hohen Bereich bewirkt kaum eine Änderung der Validierungswerte. Zwei Werte (8.5 und 9) erzielen einen maximalen PSS-Wert hinsichtlich der Plankurvatur (Abb. 5.1 (e)).

Der PSS-Wert für den Parameter Minimalfläche steigt leicht bis zu einem Wert von  $4000 \text{ m}^2$  und fällt danach ab (Abb. 5.1(f)). Das heisst, der höchstmögliche PSS-Wert wird erreicht, wenn man alle ausgeschiedenen Anrissgebiete kleiner als  $4000 \text{ m}^2$  weglässt.



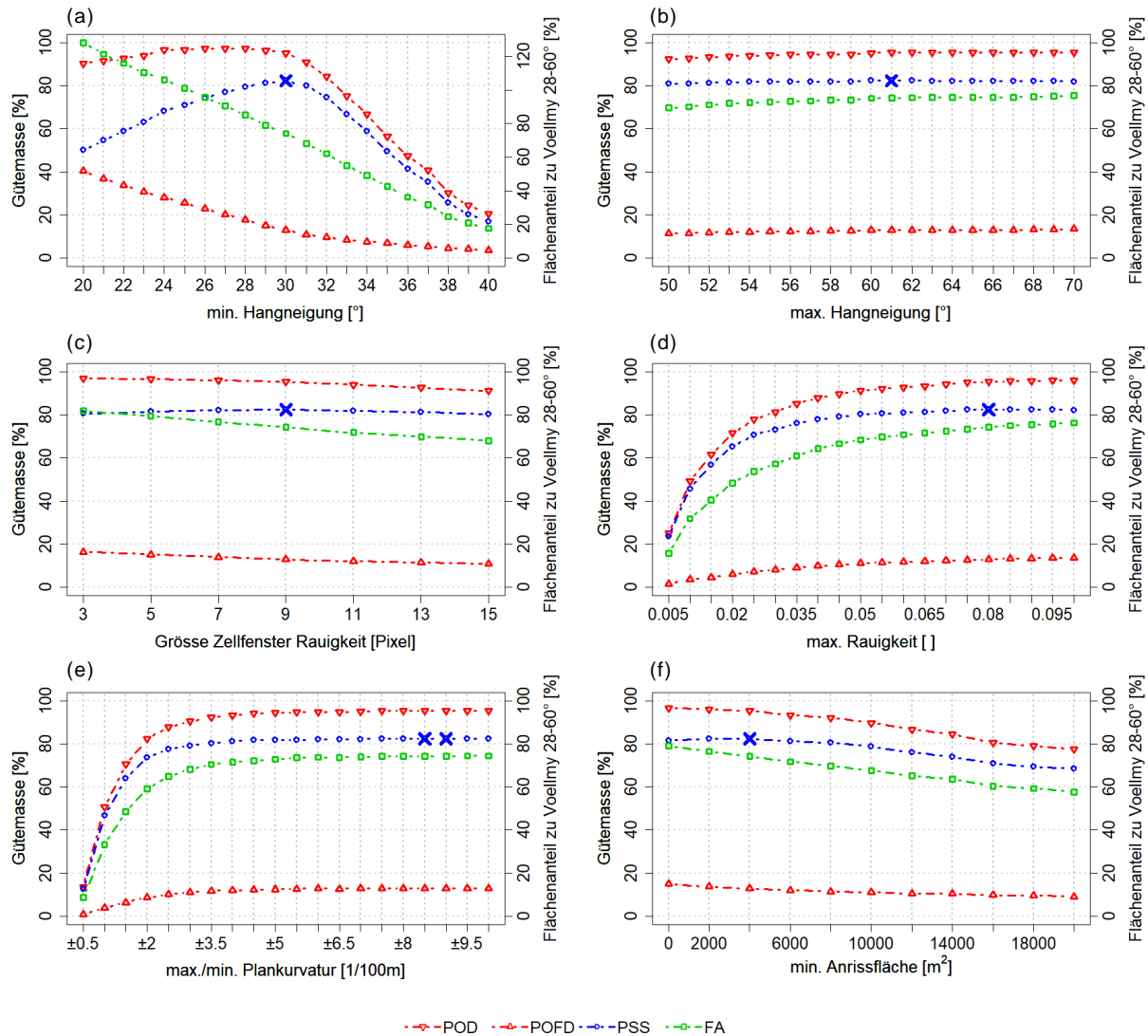


Abb. 5.1: Kalibrierung der Parameter Hangneigung (a) (b), Rauigkeit (c) (d), Plankurvatur (e) sowie Anrissfläche (f) des Algorithmus von Bühler et al. (2013) für das Trainingsgebiet Parsenn; Die blauen Kreuze markieren die optimalen Parameterwerte [\*]

### 5.1.2 Veitinger et al. (2016)

Der Algorithmus von Veitinger et al. (2016) verlangt nach einer Schneehöhe als Eingangsparameter. Für Gefahrenhinweiskarten sind keine Wiederkehrperioden explizit vermerkt (Camenzind-Wildi et al. 2000). Zur Anwendung des Algorithmus wird für das Trainingsgebiet eine Schneehöhe mit einer Wiederkehrperiode von 100 Jahren berechnet. Als Grundlage dafür wird auf die Wetterstationsdaten von EVA+ (2018) zurückgegriffen. Für diese Wetterstationen können mittels Extremwertstatistik mit EVA+ (2018) die 100-jährlichen Schneehöhen berechnet werden. Diese Schneehöhenwerte müssen aber von den Stationshöhen auf die mittlere Höhe über Meer des Trainingsgebietes korrigiert werden. Für die Korrektur wird eine Abhängigkeit der 100-jährlichen Schneehöhe mit der Höhe über Meer von Auer (2005) auf Basis von Martinec (1990) angenommen. Das Trainingsgebiet

Parsenn (Stationen Weissfluhjoch, Davos) befindet sich betreffend der Höhenabhängigkeit der Schneehöhen in der Zone 2 (Abb. 5.2).

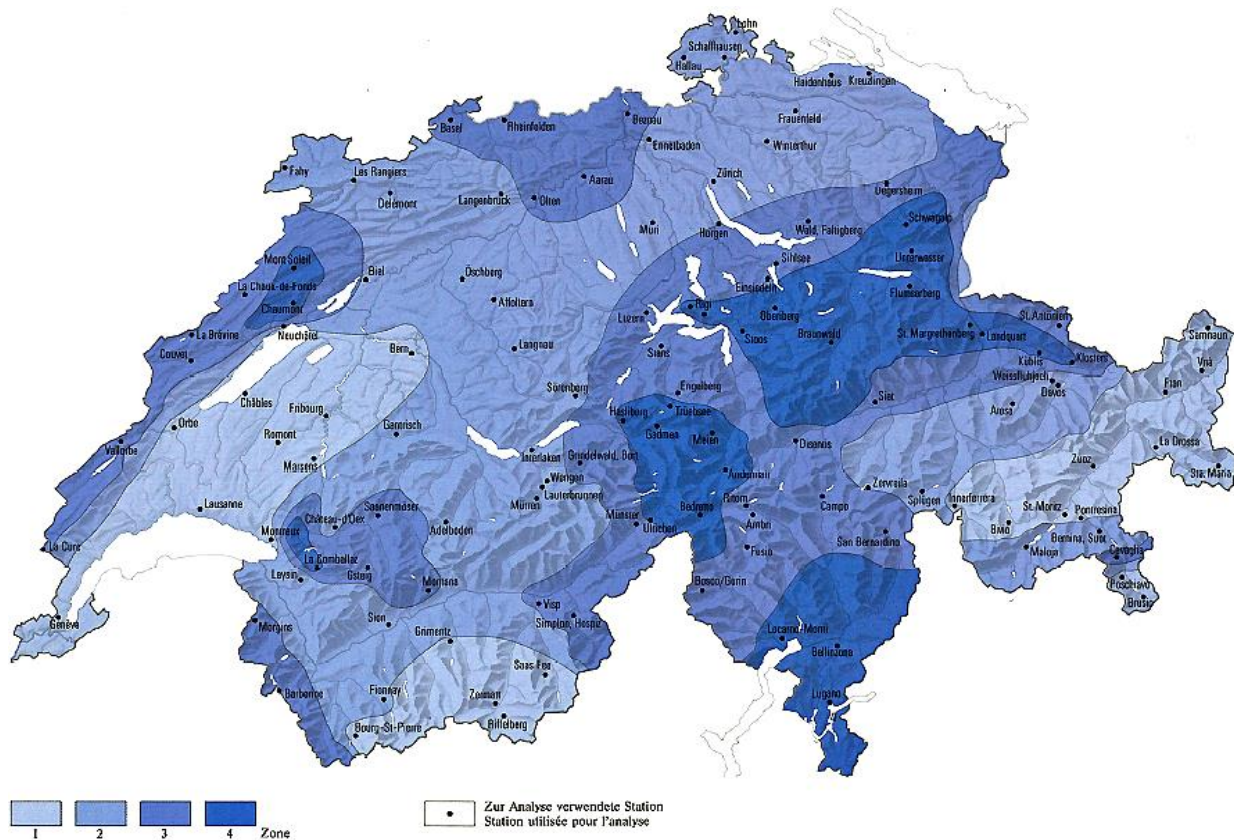


Abb. 5.2: Zonierung der Abhängigkeit zwischen der Höhe über Meer und der 100-jährlichen Schneehöhe (Martinec 1990)

Für Zone 2 lässt sich gemäss Auer (2005) die Höhenabhängigkeit mathematisch folgendermassen formulieren (Gl. 5.1):

$$HS_{100} = 1.35(0.12h + 2) \quad \text{Gl. 5.1}$$

$HS_{100}$ : Schneehöhe [cm], Wiederkehrperiode 100 Jahre

$h$ : Höhe über Meer [m]

Daraus folgt ein Höhengradient der 100-jährlichen Schneehöhe von (Gl. 5.2):

$$\frac{\Delta HS_{100}}{\Delta h} = 0.162 \frac{\text{cm}}{\text{m}} \quad \text{Gl. 5.2}$$

Basierend auf dem Höhengradient (Gl. 5.2) kann eine 100-jährliche Stationsschneehöhe auf die mittlere Höhe des Trainingsgebiets korrigiert und somit eine 100-jährliche Schneehöhe für das Trainingsgebiet berechnet werden (Gl. 5.3):

$$HS_{100}^{TG} = \frac{\Delta HS_{100}}{\Delta h} \times (\bar{h}_{TG} - h_{Station}) + HS_{100}^{Station} \quad \text{Gl. 5.3}$$

$HS_{100}^{TG}$ : Schneehöhe [cm] Untersuchungsgebiet, Wiederkehrperiode 100 Jahre

$\bar{h}_{TG}$ : mittlere Höhe über Meer [m] des Untersuchungsgebiets

$h_{Station}$ : Höhe über Meer [m] der Messstation

$HS_{100}^{Station}$ : Schneehöhe [cm] Messstation, Wiederkehrperiode 100 Jahre

Für die zwei Messstationen Davos Flüelastrasse und Davos Weissfluhjoch werden die 100-jährlichen Schneehöhen auf die mittlere Höhe des Untersuchungsgebiets korrigiert und daraus der Durchschnitt für eine mittlere 100-jährliche Schneehöhe für das Trainingsgebiet berechnet (Tab. 5.1). Die 100-jährlichen Schneehöhen der Messstationen stammen von EVA+ (2018). Daraus folgt eine mittlere 100-jährliche Schneehöhe von 363.1cm für das Trainingsgebiet. Als Eingabewert für die mittlere Schneehöhe werden näherungsweise 360cm angenommen.

Tab. 5.1: Berechnung der mittleren 100-jährlichen Schneehöhe für das Gebietes Parsenn

Stationsname	Messreihe	$h_{Station}$ [m.ü.M.]	$\bar{h}_{TG}$ [m.ü.M.]	$\bar{h}_{TG} - h_{Station}$ [m]	$HS_{100}^{Station}$ [cm]	$\frac{\Delta HS_{100}}{\Delta h}$ [cm/m]	$HS_{100}^{TG}$ [cm]
Davos Flüelastrasse	1946-2018	1560	2464	904	234.2	0.162	380.6
Davos Weissfluhjoch	1934-2018	2540	2464	-76	357.8	0.162	345.5
$\overline{HS}_{100}^{TG}$ [cm]							363.1

In Abbildung Abb. 5.3(a) ist zu erkennen, dass mittlere Schneehöhen von 3 bis 3.75m alle einen maximalen PSS-Wert erreichen. Das ist so, weil aus dem Algorithmus von Veitinger et al. (2016) für diese Schneehöhen der gleiche Glättungsfaktor für das DHM und infolgedessen dieselben Hangneigungs- und Rauigkeitsraster resultieren. Folglich werden für mittlere Schneehöhen von 3 bis 3.75m dieselben Anrissgebiete ausgeschieden. In diesem Bereich liegt auch die exakte berechnete Schneehöhe von 363.1m (Tab. 5.1).

Wie in Kapitel 3.1.2 beschrieben, wird für die Anrissanfälligkeit, gegeben in kontinuierlichen Werten von 0 bis 1, ein Schwellenwert gesetzt, um die binären Klassen PAG und nPAG zu erhalten. Für eine minimale Anrissanfälligkeit von 0.03 als Schwellenwert resultiert für das Gebiet Parsenn der

maximale PSS-Wert (Abb. 5.3(b)). Im Bereich von 0 bis 0.1 minimaler Anrissanfälligkeit wurde mit einem Intervall von 0.01 iteriert. Bei der graphischen Darstellung wurden jedoch die 0.01-Intervalle zur besseren Lesbarkeit weggelassen. Ein Wert von 0 für die minimale Anrissanfälligkeit scheidet die ganze Fläche des Perimeters als PAG aus, wodurch sich ein POD- und POFD-Wert von 100%, ein PSS-Wert von 0% und ein FA-Wert von 254.88% ergibt. Ein Wert von 254.88% würde den Wertebereich der sekundären y-Achse massiv aufspannen. Deshalb wurde zur besseren Darstellung anstatt 0 ein Wert nahe 0 ( $1e-9 = 0.000000001$ ) gewählt.

Eine Änderung der minimalen Anrissfläche hat nur geringfügig eine Änderung des PSS-Wertes zur Folge (Abb. 5.3(c)). Wie in Kapitel 3.1.2 beschrieben, unterteilt der Algorithmus von Veitinger et al. (2016) die Anrissflächen nicht weiter in individuelle Anrissgebiete. Dadurch ergeben sich viele kleine, isolierte Flächen und wenige sehr grosse ( $>20000\text{m}^2$ ). Eine Veränderung des Schwellenwertes im Bereich unter  $20000\text{m}^2$ , betrifft nur kleine Flächen (der Flächenanteil ändert sich nur geringfügig). Dies hat kaum eine Änderung der Gütemasse zu Folge.

Es zeigte sich während der Kalibrierung, dass dieser Algorithmus, wohl verursacht durch die multiskalige Berechnung der Geländeparameter, längere Rechenzeiten im Vergleich zu den anderen Algorithmen benötigt.

### 5.1.3 Bühler et al. (2018)

Der erste Prozessierungsschritt des Algorithmus von Bühler et al. (2018) ist die Generierung einer Vorauswahl mit potentiell möglichen Anrissflächen. Die Generierung der Vorauswahl basiert stark auf dem Algorithmus von Bühler et al. (2013) mit dem Hauptunterschied, dass das DHM vor dessen Weiterverwendung geglättet wird (Kapitel 3.1.2).

Infolgedessen kann bei der Iteration für die meisten Parameter dasselbe Verhalten wie in Abb. 5.1 beobachtet werden: Der Schwellenwert bezüglich der minimalen Hangneigung liegt bei  $30^\circ$  (Abb. 5.4(a)). Hinsichtlich der maximalen Hangneigung weisen  $58^\circ$  und  $59^\circ$  denselben maximalen PSS-Wert auf (Abb. 5.4(b)). Für die Grösse des Zellfensters zur Berechnung der Rauigkeit beträgt der optimale Parameterwert 9 Pixel (Abb. 5.4(c)), jener betreffend der maximalen Rauigkeit 5.5 (Abb. 5.4(d)). Bezüglich der Plankurvatur ergeben sich die Schwellenwerte von  $\pm 6/100\text{m}$  (Abb. 5.4(e)). Abb. 5.4(f) zeigt die Iteration über die minimale Anrissfläche bezogen auf die Vorauswahl. Das heisst isolierte Flächen kleiner als  $4000\text{m}^2$  werden weggelassen.

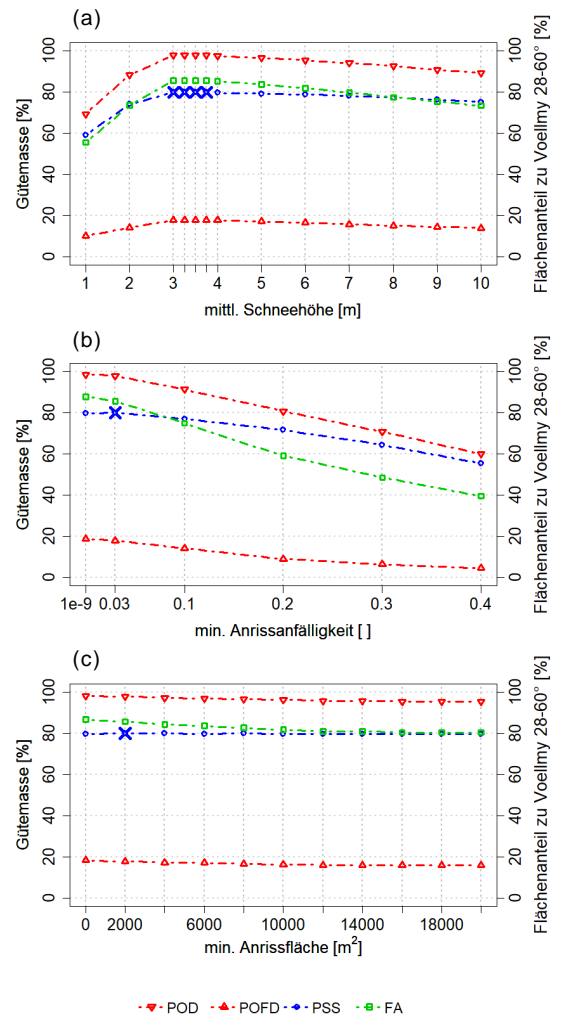


Abb. 5.3: Kalibrierung der Parameter mittlere Schneehöhe (a), Anrissanfälligkeit (b) sowie Anrissfläche (c) des Algorithmus von Veitinger et al. (2016) für das Trainingsgebiet Parsenn; Die blauen Kreuze markieren die optimalen Parameterwerte [\*]

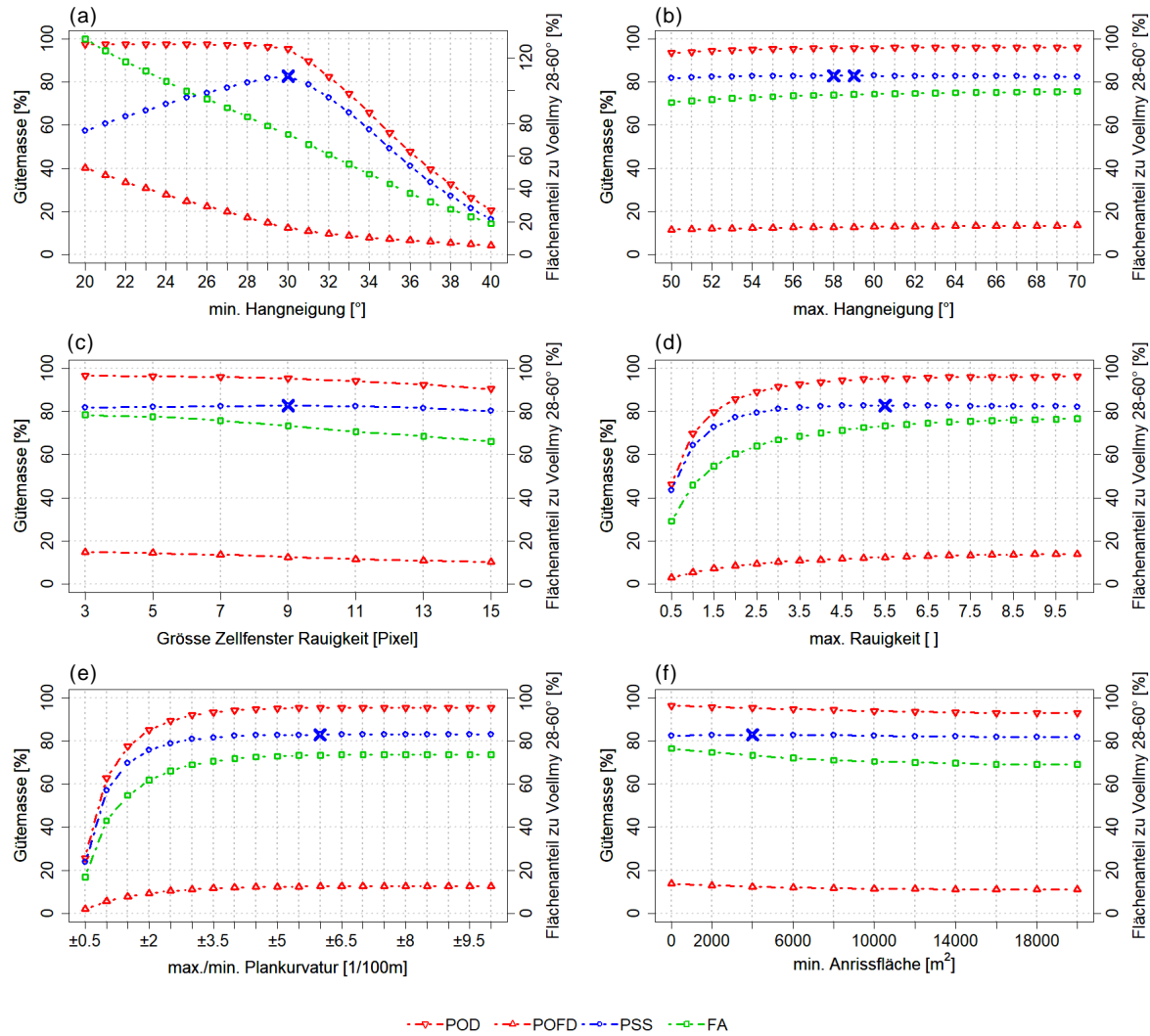


Abb. 5.4: Kalibrierung der Parameter Hangneigung (a) (b), Rauigkeit (c) (d), Plankurvatur (e) sowie Anrissfläche (f) des Algorithmus von Bühler et al. (2018) für das Trainingsgebiet Parsenn; Die blauen Kreuze markieren die optimalen Parameterwerte [\*]

## 5.2 Validierung und Vergleich

Die Algorithmen, beschrieben in Kapitel 3.1.2, werden gemäss Kapitel 4.3.4 für die Gebiete Parsenn, Jakobs- und Rinerhorn validiert und miteinander verglichen. Der Algorithmus Voellmy (1955) 28-60° erreicht hinsichtlich des Flächenanteils jeweils einen Wert von 100%, weil dessen absolut unterschiedene Fläche als Flächenrichtwert dient (Kapitel 4.3.1).

### 5.2.1 Parsenn

Die Validierungsergebnisse für das Trainingsgebiet Parsenn sind in Tab. 5.2 aufgelistet: Den höchsten POD-Wert erreicht Voellmy (1955) 28-60° und den tiefsten Bühler et al. (2013). Den tiefsten POFD-Wert erreicht Bühler et al. (2018) Vorauswahl und den höchsten Voellmy (1955) 28-60°. Bezüglich des decisiven PSS zeigt Voellmy (1955) 28-60° den tiefsten Wert. Veitinger et al. (2016) und Bühler et al. (2018) *extreme* erreichen bereits deutlich höhere PSS-Werte. Bühler et al. (2018) Vorauswahl

zeigt den höchsten und Bühler et al. (2018) *frequent*, noch vor Bühler et al. (2013), den zweithöchsten PSS-Wert. Diese drei letztgenannten Algorithmen erreichen ebenfalls die höchsten HSS-Werte und bezüglich des Flächenanteils die tiefsten Werte.

Tab. 5.2: Validierungsergebnisse aller getesteten Algorithmen für das Trainingsgebiet Parsenn

Algorithmus	POD [%]	POFD [%]	PSS [%]	HSS [%]	FA [%]	Kartographische Darstellung
Voellmy (1955) 28-60°	98.75	24.70	74.05	53.75	100.00	Abb. 5.5
Bühler et al. (2013)	95.43	12.96	82.47	69.70	74.33	Abb. 5.6
Veitinger et al. (2016)	97.85	17.90	79.94	63.00	85.62	Abb. 5.7
Bühler et al. (2018) Vorauswahl	95.93	12.85	83.08	70.20	74.31	Abb. 5.8
Bühler et al. (2018) <i>frequent</i>	95.82	13.17	82.65	69.56	74.92	Abb. 5.9
Bühler et al. (2018) <i>extreme</i>	97.98	16.78	81.20	64.85	83.38	Abb. 5.10

Die Unterschiede der Algorithmen sind in den kartographischen Darstellungen vor allem für die Bereiche rund um Geländekanten und Grate sowie felsige Partien erkennbar. Der Algorithmus nach Voellmy (1955) 28-60° (Abb. 5.5) scheidet beispielsweise den felsigen Bereich östlich des Felsenweges, im südwestlichen Spitz des Perimeters, fast komplett als PAG aus. Die anderen Algorithmen, vor allem Bühler et al. (2013) (Abb. 5.6) und Bühler et al. (2018) (Abb. 5.8), klassieren in diesem Bereich weniger Fläche als PAG. Aus Gründen der Kartenlesbarkeit werden ausschliesslich für den Algorithmus Voellmy (1955) 28-60° die Referenzdaten in aggregierter Form graphisch hinterlegt.

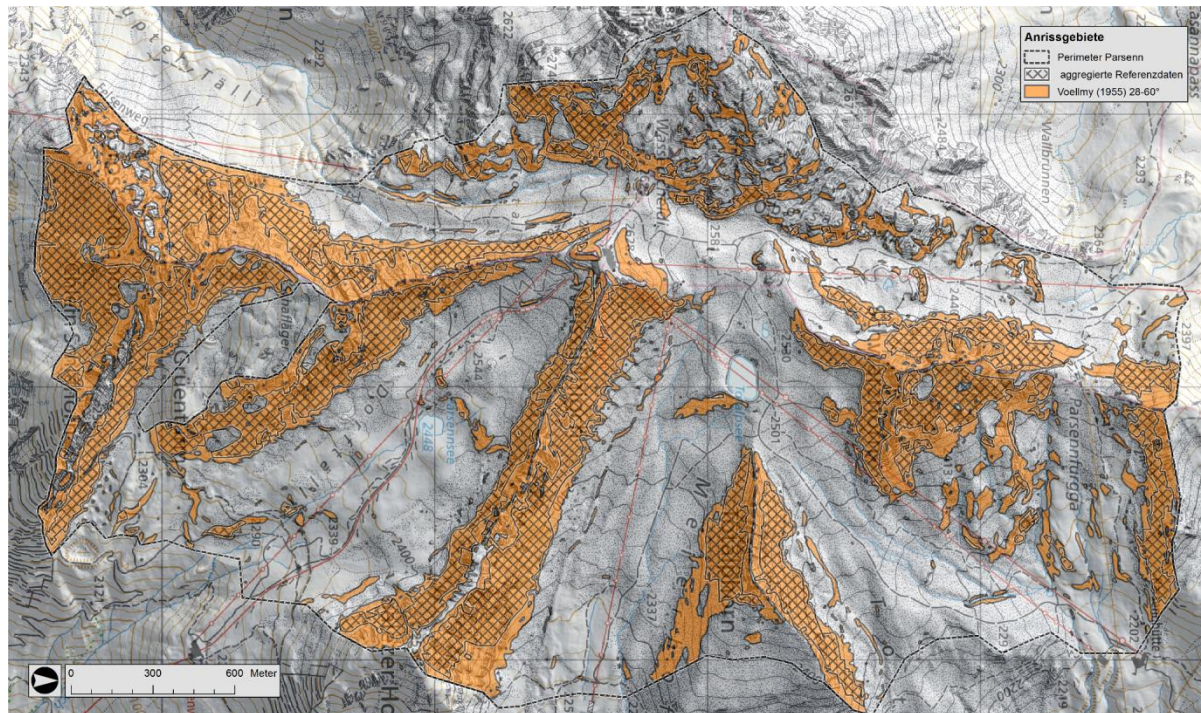


Abb. 5.5: Lawinenanrissgebiete des Algorithmus Voellmy (1955) 28-60° für das Gebiet Parsenn [#\*]

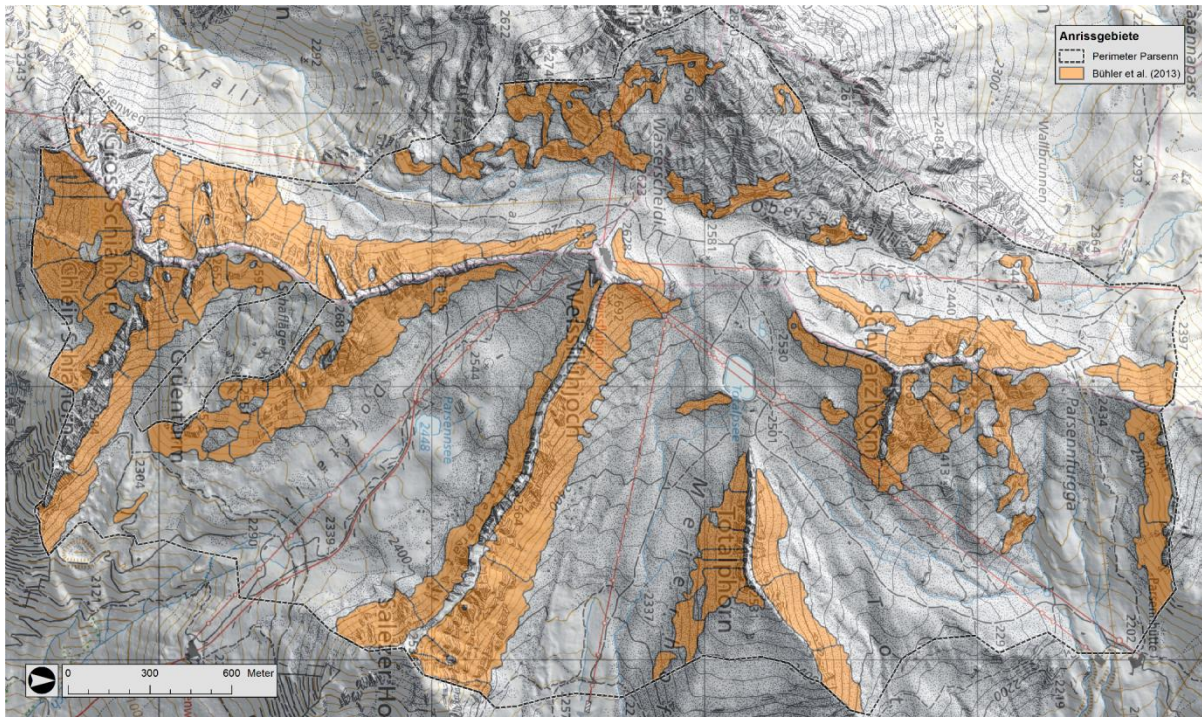


Abb. 5.6: Lawinenanrissgebiete des Algorithmus Bühler et al. (2013) das Gebiet Parsenn [#\*]

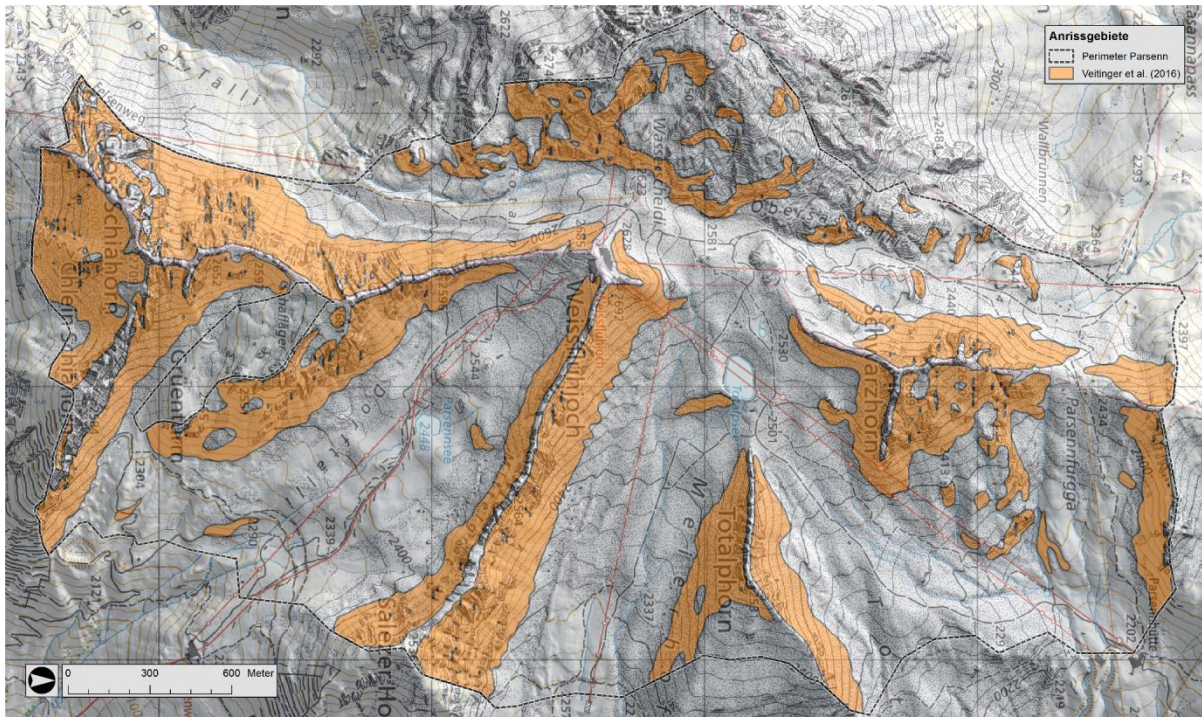


Abb. 5.7: Lawinenanrissgebiete des Algorithmus Veitinger et al. (2016) für das Gebiet Parsenn [#\*]

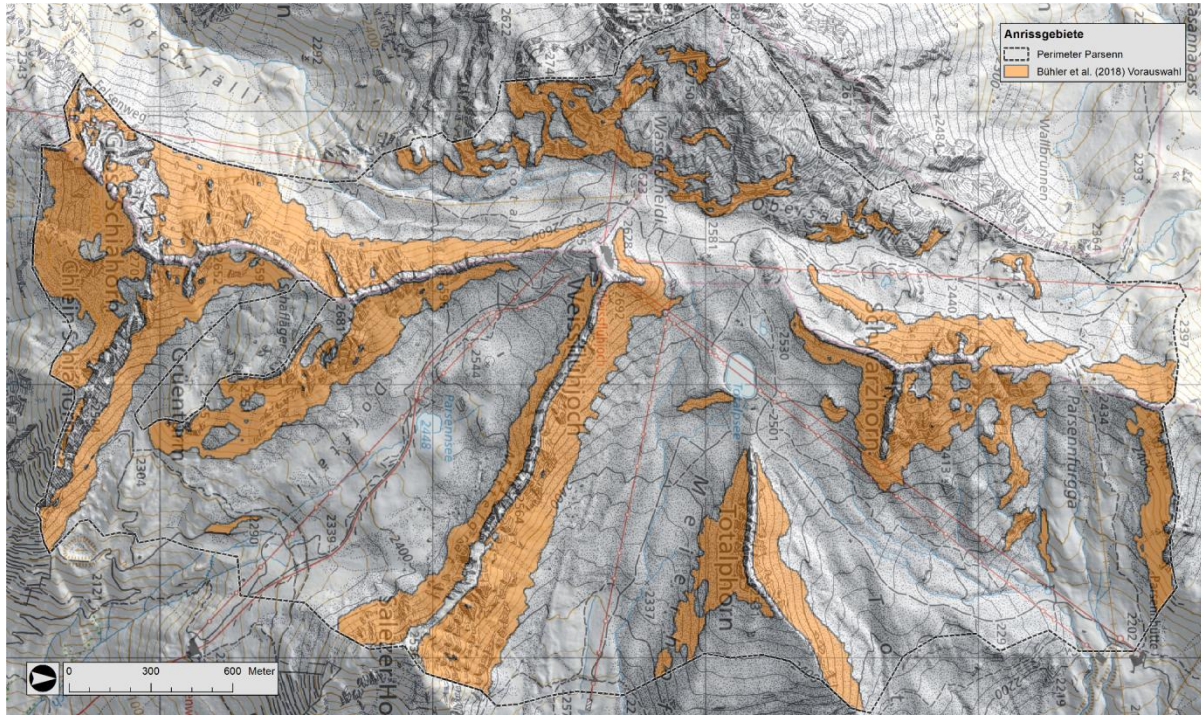


Abb. 5.8: Lawinanrissgebiete des Algorithmus Bühler et al. (2018) Vorauswahl für das Gebiet Parsenn [#\*]

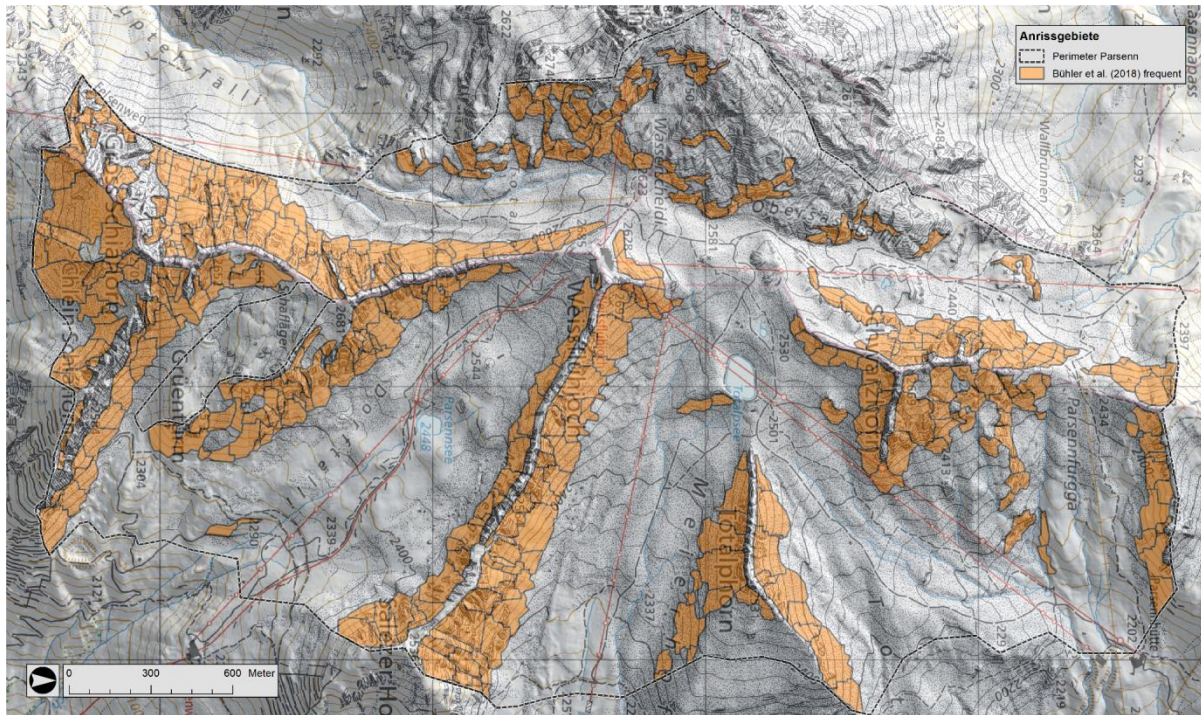


Abb. 5.9: Lawinanrissgebiete des Algorithmus Bühler et al. (2018) frequent für das Gebiet Parsenn [#\*]



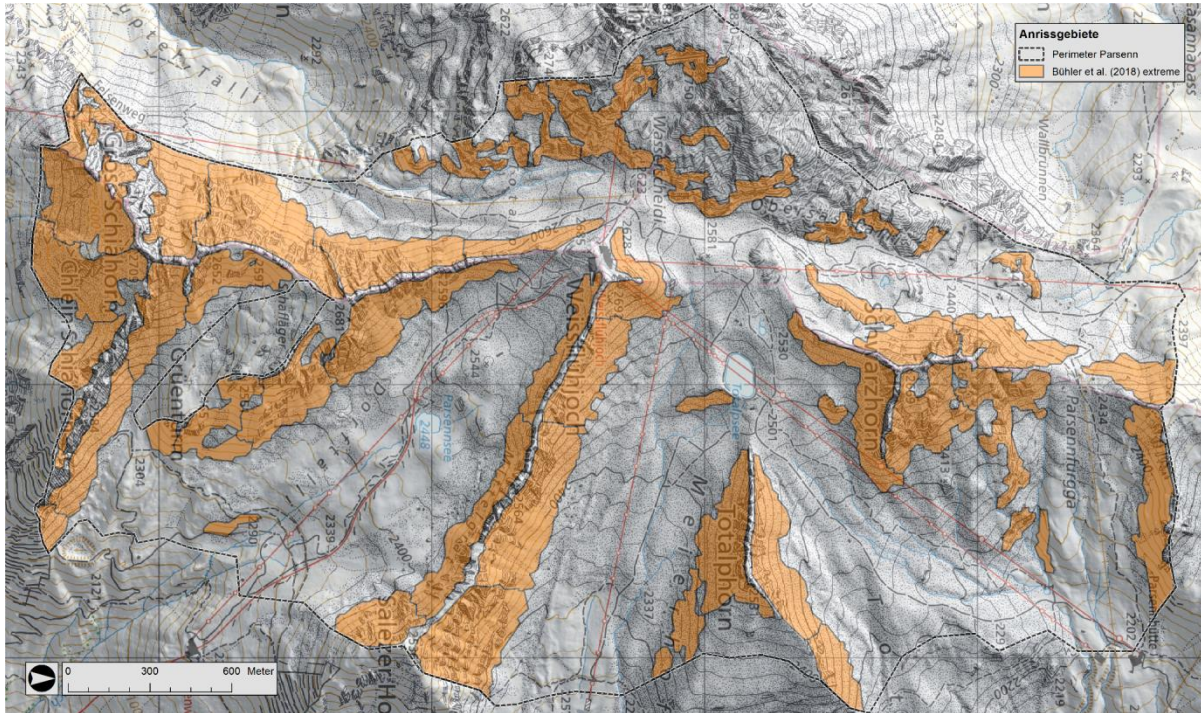


Abb. 5.10: Lawinenanrissgebiete des Algorithmus Bühler et al. (2018) *extreme* für das Gebiet Parsenn [#\*]

## 5.2.2 Jakobs- und Rinerhorn

Die Validierungsergebnisse für das Testgebiet Jakobs- und Rinerhorn sind in Tab. 5.3 aufgelistet. Es präsentiert sich ein ähnliches Bild wie für das Trainingsgebiet Parsenn: Den höchsten POD-Wert erreicht Voellmy (1955) 28-60° und den tiefsten Bühler et al. (2013). Den tiefsten POFD-Wert erreicht Bühler et al. (2018) Vorauswahl und den höchsten Voellmy (1955) 28-60°. Bezüglich des decisiven PSS zeigt Voellmy (1955) 28-60° den tiefsten Wert. Veitinger et al. (2016) und Bühler et al. (2018) *extreme* erreichen bereits deutlich höhere PSS-Werte. Bühler et al. (2018) Vorauswahl zeigt den höchsten und Bühler et al. (2018) *frequent*, noch vor Bühler et al. (2013), den zweithöchsten PSS-Wert. Diese drei Algorithmen erreichen ebenfalls die höchsten HSS-Werte und bezüglich des Flächenanteils die tiefsten Werte. Speziell zu erwähnen gilt, dass Bühler et al. (2018) *frequent* hinsichtlich aller Validierungswerte besser abschneidet als Bühler et al. (2013).

Tab. 5.3: Validierungsergebnisse aller getesteten Algorithmen für das Testgebiet Jakobs- und Rinerhorn

Algorithmus	POD [%]	POFD [%]	PSS [%]	HSS [%]	FA [%]	Kartographische Darstellung
Voellmy (1955) 28-60°	98.58	35.23	63.35	48.32	100.00	Abb. 5.11, Abb. 5.12
Bühler et al. (2013)	95.75	21.45	74.30	63.05	79.10	Abb. 5.13, Abb. 5.14
Veitinger et al. (2016)	98.18	29.56	68.62	54.51	91.76	Abb. 5.15, Abb. 5.16
Bühler et al. (2018) Vorauswahl	95.84	20.80	75.04	63.98	78.15	Abb. 5.17, Abb. 5.18
Bühler et al. (2018) <i>frequent</i>	95.91	21.35	74.57	63.30	78.96	Abb. 5.19, Abb. 5.20
Bühler et al. (2018) <i>extreme</i>	98.44	26.50	71.93	58.38	87.55	Abb. 5.21, Abb. 5.22

Die Unterschiede der Algorithmen sind in den kartographischen Darstellungen ebenfalls vor allem für die Bereiche rund um Geländekanten und Grate sowie felsige Partien erkennbar. Da das Gebiet Jakobshorn wenig felsige Partien aufweist, sind die graphischen Unterschiede zwischen Algorithmen weniger offensichtlich. Dennoch sind Unterschiede bei markanten Graten im Zentrum des Perimeters Jakobshorn erkennbar: Der Algorithmus Voellmy (1955) 28-60° klassiert die Bereiche über Grate hinweg als PAG (Abb. 5.11). Bei anderen Algorithmen wird weniger Fläche unmittelbar um und auf dem Grat als PAG klassiert (Abb. 5.13, Abb. 5.15 und Abb. 5.17). Offensichtlicher sind die graphischen Unterschiede für das Gebiet Rinerhorn, weil dieses grössere Felspartien aufweist: Der Algorithmus nach Voellmy (1955) 28-60° (Abb. 5.12) scheidet beispielsweise den felsigen Bereich im südlichen Teil des Perimeters, fast komplett als PAG aus. Die anderen Algorithmen, vor allem Bühler et al. (2013) (Abb. 5.14) und Bühler et al. (2018) (Abb. 5.18), klassieren in diesem Bereich weniger Fläche als PAG. Aus Gründen der Kartenlesbarkeit werden ebenfalls ausschliesslich für den Algorithmus Voellmy (1955) 28-60° die Referenzdaten in aggregierter Form bei der graphischen Darstellung hinterlegt.

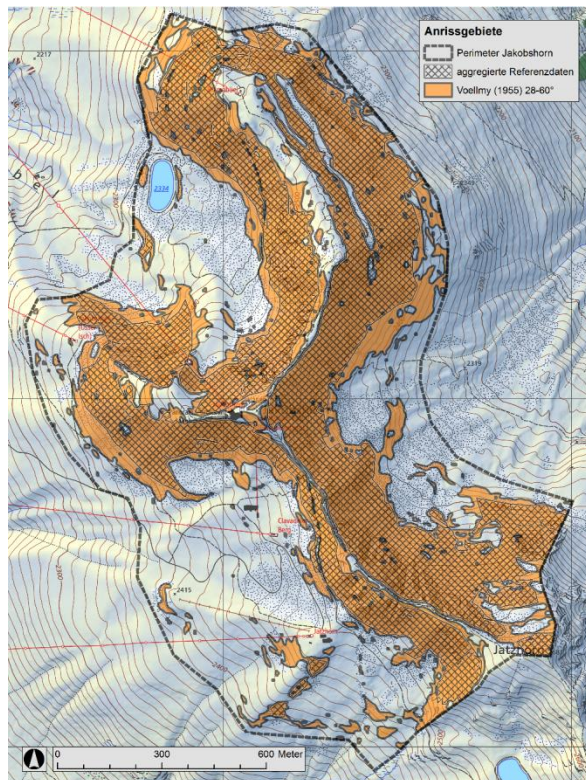


Abb. 5.11: Lawinenanrissgebiete des Algorithmus Voellmy (1955) 28-60° für das Gebiet Jakobshorn [#\*]

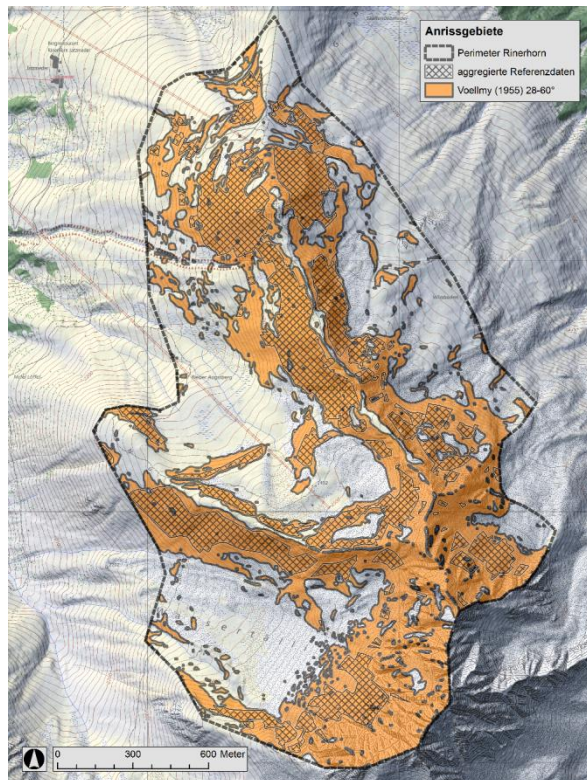


Abb. 5.12: Lawinenanrissgebiete des Algorithmus Voellmy (1955) 28-60° für das Gebiet Rinerhorn [#\*]

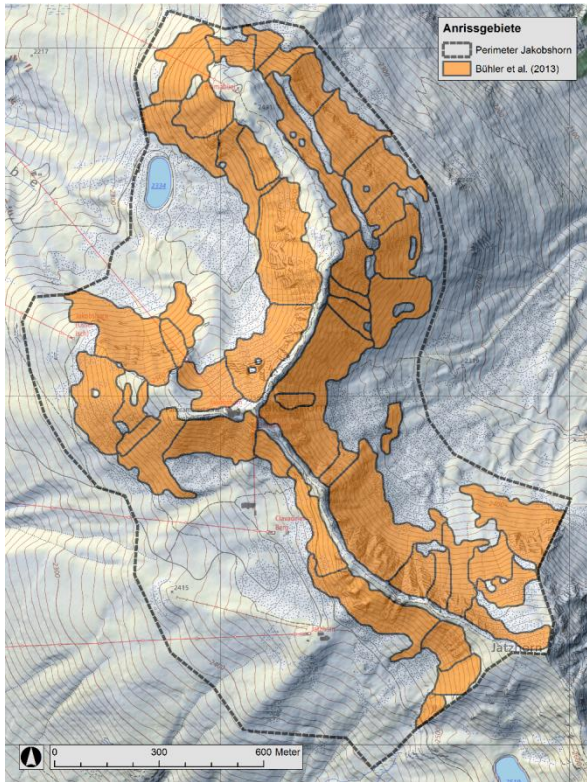


Abb. 5.13: Lawinenanrissgebiete des Algorithmus Bühler et al. (2013) für das Gebiet Jakobshorn [#\*]

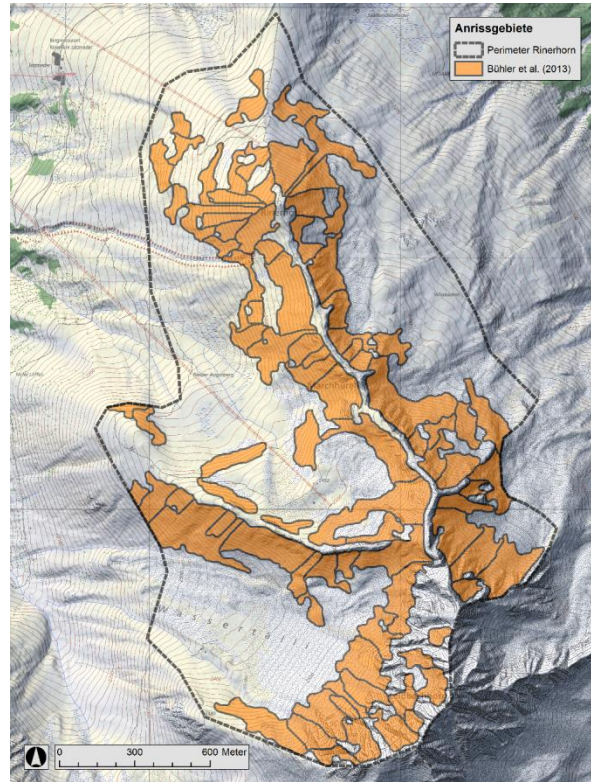


Abb. 5.14: Lawinenanrissgebiete des Algorithmus Bühler et al. (2013) für das Gebiet Rinerhorn [#\*]

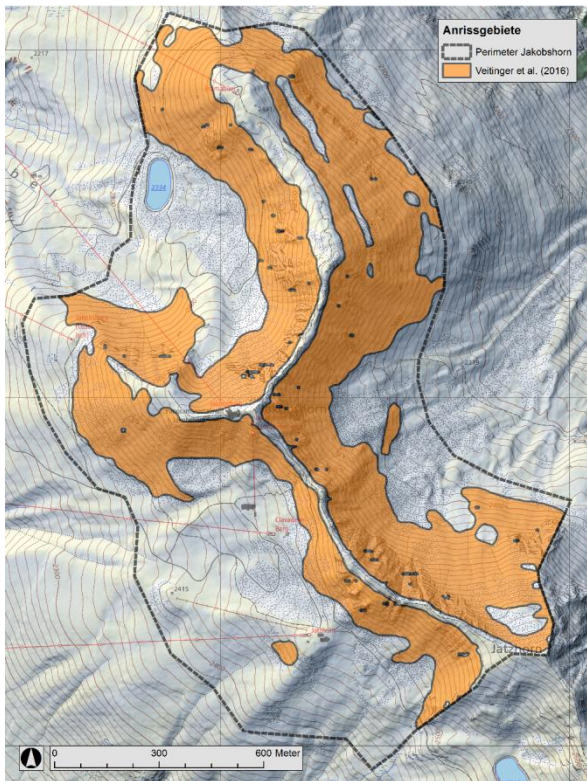


Abb. 5.15: Lawinenanrissgebiete des Algorithmus Veitinger et al. (2016) für das Gebiet Jakobshorn [#\*]

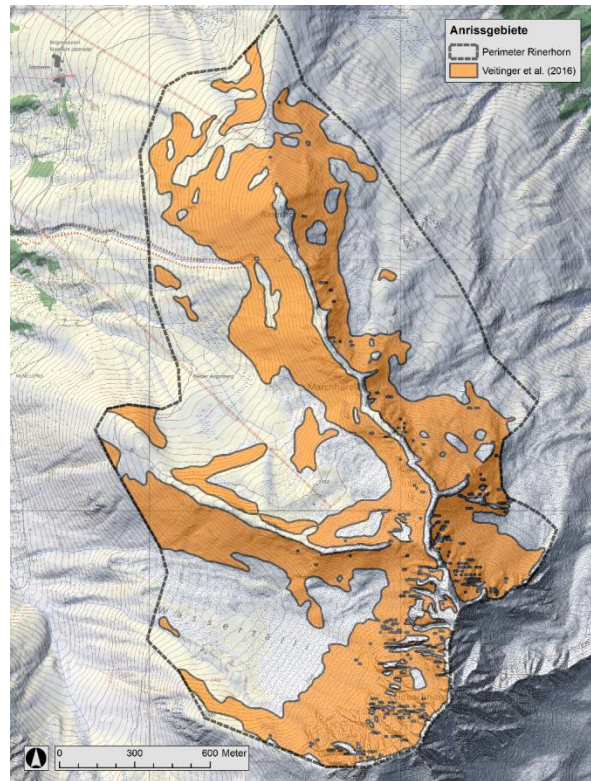


Abb. 5.16: Lawinenanrissgebiete des Algorithmus Veitinger et al. (2016) für das Gebiet Rinerhorn [#\*]

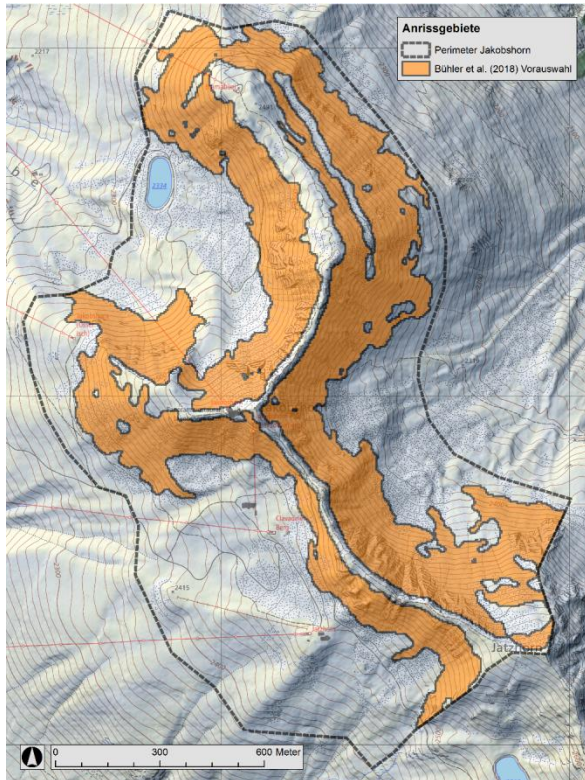


Abb. 5.17: Lawinenanrissgebiete des Algorithmus Bühler et al. (2018) Vorauswahl für das Gebiet Jakobshorn [#\*]

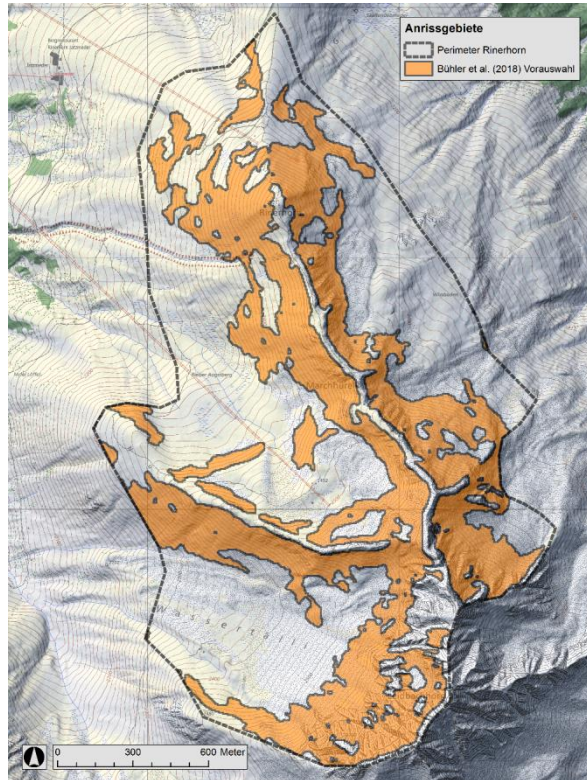


Abb. 5.18: Lawinenanrissgebiete des Algorithmus Bühler et al. (2018) Vorauswahl für das Gebiet Rinerhorn [#\*]

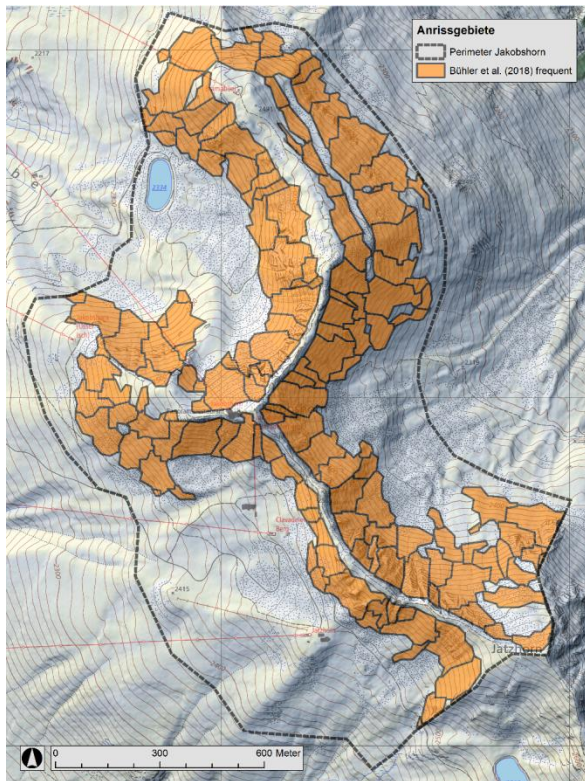


Abb. 5.19: Lawinenanrissgebiete des Algorithmus Bühler et al. (2018) frequent für das Gebiet Jakobshorn [#\*]

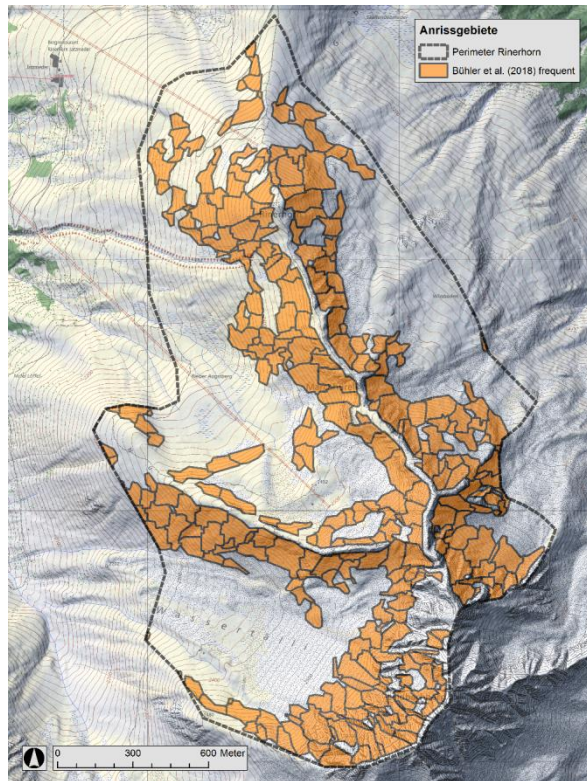


Abb. 5.20: Lawinenanrissgebiete des Algorithmus Bühler et al. (2018) frequent für das Gebiet Rinerhorn [#\*]

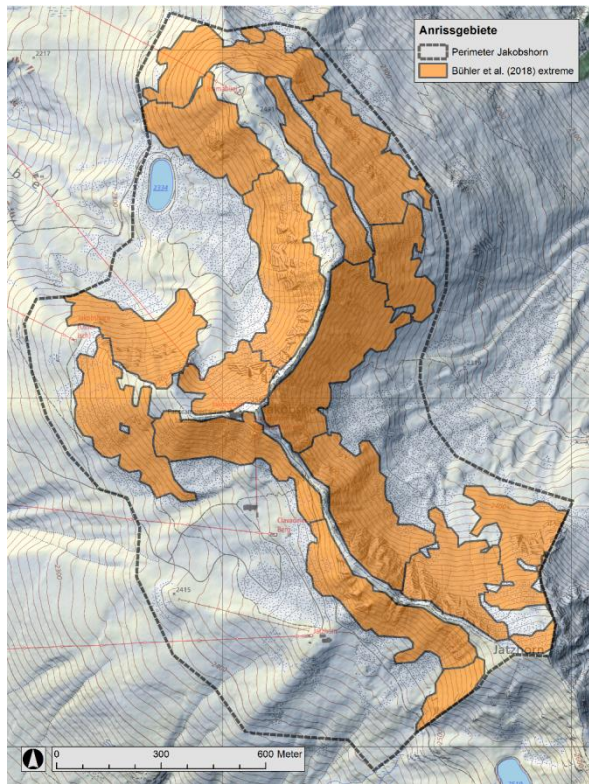


Abb. 5.21: Lawinenanrissgebiete des Algorithmus Bühler et al. (2018) *extreme* für das Gebiet Jakobshorn [#\*]

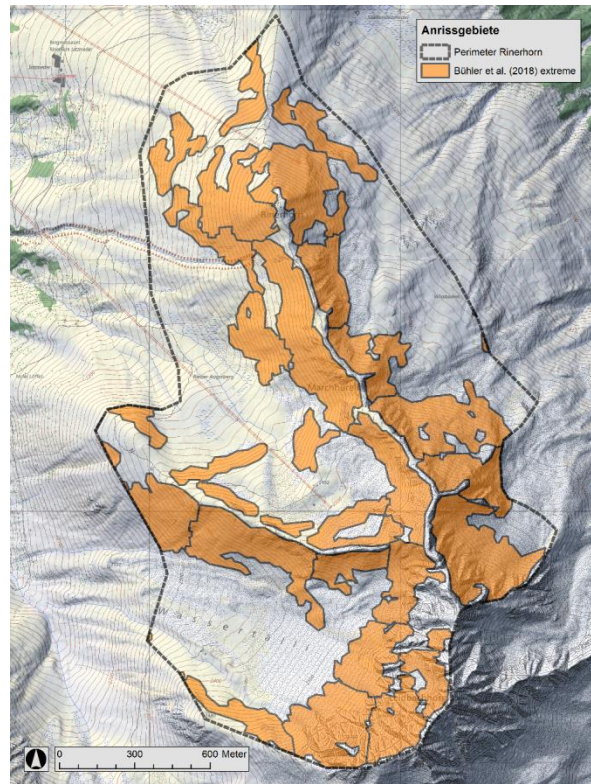


Abb. 5.22: Lawinenanrissgebiete des Algorithmus Bühler et al. (2018) *extreme* für das Gebiet Rinerhorn [#\*]

## 6 Diskussion

### 6.1 Algorithmen zur automatischen Identifizierung potentieller Anrissgebiete

Die Auslösung einer Schneebrettlawine ist, direkt oder indirekt, unter anderem durch bestimmte Geländeeigenschaften beeinflusst oder sogar bedingt (Kapitel 2.1). Diese Geländeeigenschaften sind mehrheitlich bekannt und deren Wirkungsweisen zunehmend besser erforscht. Auf Basis eines digitalen Höhenmodells lassen sich durch die Geomorphometrie Geländeeigenschaften für digitale Höhenmodelle als Parameter beschreiben (Kapitel 2.3). Die Verknüpfung der lawinenbildenden Faktoren und der Geomorphometrie ermöglicht die automatische Identifizierung potentieller Anrissgebiete auf Basis des digitalen Geländemodells.

In der Vergangenheit wurden mehrere Ansätze entwickelt, um automatisch potentielle Lawinenanrissgebiete auf Basis des digitalen Geländemodells zu identifizieren. Das erste Ziel dieser Arbeit bestand darin, die bisherigen Forschungsentwicklungen hinsichtlich dieser Thematik zu erfassen. Ein Überblick über die bisher entwickelten Algorithmen ist in Kapitel 3.1 zu finden. Durch den anhaltenden technologischen Fortschritt in der Fernerkundung, wodurch räumlich höher aufgelöste digitale Modelle der Erdoberfläche verfügbar wurden, konnten weitere Geländeparameter, wie beispielsweise die Geländerauigkeit, in die Anrissgebietsidentifizierung miteinbezogen werden.

In dieser Arbeit wurde besonders die automatische Identifizierung potentieller Lawinenanrissgebiete im Hinblick auf die Gefahrenhinweiskartierung betrachtet. Bei Gefahrenhinweiskartierungen wird eine allgemeine Gefährdungsbeurteilung, unabhängig von den aktuellen schneedecken- und wetterspezifischen Verhältnissen, vorgenommen. Als Algorithmen zur Identifizierung potentieller Anrissgebiete wurden deshalb bewusst nur Algorithmen gewählt, die primär den statischen lawinenbildenden Faktor des Geländes beachten und die zeitlich variablen, lawinenbildenden Faktoren Schneedecke und Wetter vernachlässigen. Dadurch ist, mit Ausnahme von Veitinger et al. (2016), die einzig nötige Datengrundlage für diese Algorithmen ein DHM, woraus die nötigen Geländeparameter abgeleitet werden können. Zwei Algorithmen können optional räumliche Informationen zur Waldbedeckung mitberücksichtigen. Der Vergleich ist auf vier unterschiedliche Ansätze begrenzt. Die weiteren Algorithmen konnten beim Vergleich nicht miteinbezogen werden, wurden aber in Kapitel 3.1.1 beschrieben.

### 6.2 Methodik

Das Hauptziel der Arbeit war die Evaluierung und der Vergleich der vier ausgewählten Algorithmen. Um jedoch ausgewählte Algorithmen evaluieren und vergleichen zu können, ist ein Referenzdatensatz nötig. Darum war eine der Herausforderungen die Erstellung eines bestmöglichen Referenzdatensatzes: Das Anrissgebiet von Schneelawinen befindet sich oft in schwer zugänglichem Gelände und ist aufgrund von Neuschnee oder Schneeverwehungen meist nur für kurze Zeit als solches erkennbar (Kapitel 4.1). Darüber hinaus ist die genaue Kartierung der beobachteten Anrissgebiete in komplexem und steilem Gelände sehr anspruchsvoll. Genaue Lawinenkartierungen auf Ba-

sis von optischen oder Radar-Daten, mit ausreichender räumlicher Auflösung, können aufgrund hoher Datenerfassungskosten nur gelegentlich generiert werden. In der Region Davos gibt es eine Vielzahl von Lawinenkartierungen, die vom SLF und dem lokalen Skipatrouillenpersonal im Feld durchgeführt wurden. In den meisten Fällen wird jedoch nur der Lawinenumriss und nicht separat das Anrissgebiet kartiert. Um einen Referenzdatensatz der Anrissgebiete zu erhalten, wurden daher die Anrissgebiete aus den kartierten Lawinenumrissen abgeleitet. Die Anrissgebiete des Referenzdatensatzes zeigen die geländetypischen Eigenschaften, welche auch bei bisherigen Forschungsarbeiten eruiert wurden (Kapitel 4.1.4). Dies bestätigt die Annahme, dass Anrissgebiete von Schneebrettlawinen durch typische Geländeeigenschaften gekennzeichnet sind. Obwohl das Verfahren zur Ableitung der Anrissgebiete aus den Lawinenumrissen auf empirischen Wissen beruht, so ist es trotzdem eine Ableitung aus Primärdaten und deshalb mit gewissen Unsicherheiten behaftet. Deswegen wurde dieser provisorische Referenzdatensatz der Anrissgebiete durch lokale Experten begutachtet, angepasst und ergänzt. Trotzdem besteht eine Ungewissheit über aufgetretene, aber nicht kartierte Lawinen und deren Anrissgebiete. Es ist wohl jedoch der bestmögliche verfügbare Referenzdatensatz.

Die nächste Herausforderung im Prozess dieser Arbeit bestand darin, ein geeignetes Evaluierungsverfahren zu definieren. Die Anrissgebietsidentifizierung soll als Grundlage für die automatische Gefahrenhinweiskartierung dienen. Deswegen wurden die Algorithmen auf einen möglichst hohen PSS-Wert hin kalibriert, validiert und miteinander verglichen. Wie in Kapitel 4.2.1 beschrieben, können Prognosen auf unterschiedliche Aspekte hin beurteilt werden. Des Weiteren gibt es diverse Masszahlen um einen Aspekt bewerten zu können. Deshalb wird die Fokussierung auf den PSS-Wert zur Evaluierung und den Vergleich kurz diskutiert: Der PSS-Wert ist ein Mass zur Beurteilung der Prognosefähigkeit. Dabei berücksichtigt der PSS die Aspekte der Genauigkeit als auch der Unterscheidungsfähigkeit. Ausserdem ist er unabhängig von der Prävalenz. In Kapitel 4.1.4 wurden die geländespezifischen Eigenschaften der Anrissgebiete des Referenzdatensatzes beschrieben. Damit kann abgeschätzt werden, ob eine Optimierung auf einen möglichst hohen PSS-Wert als Kalibrierungsverfahren plausible Schwellenwerte liefert. Beispielsweise resultiert für den Algorithmus von Bühler et al. (2018) ein unterer Schwellenwert von  $30^\circ$  und oberen Schwellenwert von  $58^\circ/59^\circ$  (Abb. 5.4) und diese liegen damit im Wertebereich (Hangneigung:  $13-75^\circ$ ) der Anrissgebiete des Referenzdatensatzes (Abb. 4.9). Demnach kann angenommen werden, dass eine Evaluierung und Vergleich auf Basis des PSS-Wertes den beabsichtigten Zweck im Hinblick auf eine Gefahrenhinweiskartierung erfüllt. Werden die Algorithmen zur Identifizierung potentieller Anrissgebiete für einen anderen Zweck genutzt, stellt sich von neuem die Frage, für welches Gütemass die Algorithmen optimiert werden sollen.

Bei dem gewählten Evaluierungsverfahren werden die Algorithmen zuerst durch Iteration kalibriert (Kapitel 4.3.3). Dadurch resultierten pro Algorithmus und Gebiet bis zu 100 Durchläufe. Diese grosse Anzahl an auszuwertenden Durchläufen verlangte nach einem möglichst automatisierten Auswertungsprozess. Dies konnte dank der Programmierung von Python- und R-Skripten (im Anhang Kapitel 9) erreicht werden. Mit wenig Erfahrung in der Python-Programmierung war dies eine nicht zu unterschätzende Herausforderung und hat dementsprechend viel Zeit in Anspruch genommen.

## 6.3 Resultate

Zur Kalibrierung, Validierung und Vergleich der Algorithmen diente das Gebiet Parsenn als Trainingsgebiet und Jakobs- und Rinerhorn als Testgebiet. Damit kann die Prognosefähigkeit der Algorithmen untereinander verglichen und für zukünftige Anwendungen abgeschätzt werden.

### 6.3.1 Kalibrierung

Die Algorithmen, beschrieben in Kapitel 3.1.2, wurden anhand des Trainingsgebietes Parsenn in Kapitel 5.1 kalibriert. Der Algorithmus nach Voellmy (1955) musste nicht kalibriert werden, da deren Schwellenwerte von vornherein festgesetzt wurden. Die Algorithmen Bühler et al. (2013) und Bühler et al. (2018) werden im folgendem zusammen besprochen, da sie die gleichen Geländeparameter für die Anrissgebietsidentifizierung miteinbeziehen.

#### Bühler et al. (2013) und Bühler et al. (2018)

Je nach Algorithmus und Parameter zeigt der Verlauf des PSS-Wertes ein mehr oder weniger ausgeprägtes Maximum. Der Parameter minimale Hangneigung weist beispielsweise ausgeprägte Maxima auf. Die Parameter maximale Hangneigung, maximale Rauigkeit, maximale, beziehungsweise minimale Plankurvatur sowie minimale Anrissfläche zeigen wenig ausgeprägte PSS-Maxima. Dies ist zum einen damit zu erklären, dass eine Änderung des Parameterwertes nur wenige Pixel betrifft und folglich nur eine geringfügige Änderung des PSS-Wertes zur Folge hat. Zum anderen kann eine Änderung des Parameters eine proportionale Änderung sowohl des POD- als des POFD-Wertes nach sich ziehen und folglich kaum eine Änderung des PSS-Wertes resultieren.

Der Algorithmus von Bühler et al. (2013) als auch jener von Bühler et al. (2018) weisen einen optimalen Parameterwert von  $30^\circ$  bezüglich der minimalen Hangneigung auf. Diese  $30^\circ$  stellen die kritische Neigung zur Auslösung einer Schneebrettlawine dar (Kapitel 2.1.2). Dass bei der Kalibrierung anhand der Referenzdaten ein Schwellenwert von  $30^\circ$  resultierte, ist wohl hauptsächlich damit zu erklären, dass bei der Ableitung der provisorischen Referenzanrissgebiete aus den Lawinenumrissen eine  $30^\circ$ -Untergrenze verwendet wurde (Kapitel 4.1.2).

Bezüglich der Grösse des Zellfensters zur Berechnung der Rauigkeit ist für den Algorithmus von Bühler et al. (2013) und Bühler et al. (2018) folgendes beobachten: Mit Zunahme der Zellfenstergrösse bei gleich bleibendem Schwellenwert für die Rauigkeit wird am Übergang von rauem zu weniger rauem Gelände dem Zentralpixel häufig ein höherer Rauigkeitswert zugewiesen, da in solchen Fällen ein grösseres Fenster mehr raue Werte miteinbezieht. Damit ist die Abnahme des FA-Wertes auf Grund der Zunahme der Zellfenstergrösse zu erklären.

Betreffend der minimalen Anrissflächen ergibt die Kalibrierung optimale Schwellenwerte von  $4000 \text{ m}^2$  für den Algorithmus von Bühler et al. (2013) als auch für den Algorithmus von Bühler et al. (2018). Der Algorithmus Bühler et al. (2013) zeigt eine Änderung des PSS-Wertes in Abhängigkeit der minimalen Anrissfläche. Da dieser Algorithmus eine Unterteilung der potentiellen Anrissflächen in individuelle Anrissgebiete vornimmt, ergibt sich eine Änderung des PSS-Wertes in Abhängigkeit des Schwellenwertes für die minimale Anrissfläche. Für grosse Schwellenwerte nimmt der PSS-Wert deutlich ab, weil zu viele individuelle Anrissgebiete wegfallen. Der Algorithmus von Bühler et al.



(2018) nimmt auf Basis der Vorauswahl ebenfalls eine Abtrennung in individuelle Anrissgebiete vor (Kapitel 3.1.2). In diesem Fall wurde jedoch über die minimale Anrissfläche der Vorauswahl iteriert. Die Vorauswahl besteht aus vielen, kleinen und wenigen, sehr grossen Anrissflächen ( $>20000\text{m}^2$ ). Bei der Iteration über die minimale Anrissfläche von 0 bis  $20000\text{m}^2$  fallen nur die kleineren, isolierten Anrissflächen weg, was kaum eine Änderung des PSS-Wertes zur Folge hat.

### **Veitinger et al. (2016)**

Der Algorithmus von Veitinger et al. (2016) verlangt nach einer Schneehöhe als Eingabeparameter. Für das Trainingsgebiet waren in diesem Fall Daten von Messstationen zur Abschätzung einer Schneehöhe vorhanden. Dies ist wohl aber global gesehen nur für wenige alpine Regionen der Fall. Zudem birgt die Berechnung der Schneehöhe eine weitere Fehlerquelle. Dadurch ist die Anwendung dieses Algorithmus limitiert.

Betreffend der minimalen Anrissanfälligkeit ergibt sich für den Algorithmus von Veitinger et al. (2016) ein ausgesprochen kleiner Parameterwert. Das bedeutet, dass, sobald der Algorithmus einem Pixel eine bestimmte Anrissanfälligkeit wenig grösser Null (0.03) zuweist, dieses gemäss der Kalibrierung als potentiell Anrissgebiet in Frage kommt.

Hinsichtlich der minimalen Anrissfläche ist der gleiche Effekt wie für den Algorithmus von Bühler et al. (2018) zu erkennen. Die Modellausgabe weist viele, kleine und wenige, sehr grosse Anrissflächen auf. Bei der Iteration über die minimale Anrissfläche von 0 bis  $20000\text{m}^2$  fallen nur die kleineren, isolierten Anrissflächen weg, was kaum eine Änderung des PSS-Wertes zur Folge hat.

### **Weitere Kalibrierungsmöglichkeiten**

Um eine bestmögliche, breit abgestützte Kalibrierung für zukünftige Anwendungen zu erreichen, wurden die Algorithmen in einem Testlauf für alle drei Gebiete (Parsenn, Jakobs- und Rinerhorn) kalibriert, validiert und verglichen. Da in diesem Fall die Algorithmen anhand aller drei Gebiete, Parsenn, Jakobs- und Rinerhorn, kalibriert wurden, ergaben sich teilweise leichte Unterschiede bezüglich den optimalen Parameterwerten, verglichen mit den Parameterwerten, welche aus der Kalibrierung für das Gebiet Parsenn resultierten. Die Unterschiede fielen jedoch nicht gross aus, weil sich die Anrissgebiete für Parsenn und für das Jakobshorn bezüglich der Geländeparameter sehr ähnlich sind.

Des Weiteren wurde für den Algorithmus von Bühler et al. (2018) zur Kalibrierung die Häufigkeit des Auftretens einer Fläche als Anrissgebiet berücksichtigt. Dies hatte leicht tiefere Schwellenwerte für die maximale Hangneigung und die maximale Rauigkeit zur Folge. Dadurch werden aber Anrissgebiete in etwas steileren und raueren Gebieten durch den Algorithmus nicht als mögliche Anrissgebiete ausgeschieden. Für die Gefahrenhinweiskartierung sollten aber möglichst alle möglichen Anrissgebiete miteinbezogen werden, weshalb die häufigkeitsunabhängige Kalibrierung gemäss Kapitel 4.3.3 für die Gefahrenhinweiskartierung bevorzugt wird.

### 6.3.2 Validierung und Vergleich

Die Algorithmen, beschrieben in Kapitel 3.1.2, wurden in Kapitel 5.2 für die Gebiete Parsenn, Jakobs- und Rinerhorn validiert und miteinander verglichen.

#### Parsenn, Jakobs- und Rinerhorn

Der Algorithmus von Voellmy (1955) 28-60°, welcher einzig den Geländeparameter Hangneigung nutzt, erreicht jeweils einen sehr hohen POD- und tiefen POFD-Wert (Tab. 5.2 und Tab. 5.3). Das heisst, dass relativ viele Flächen durch den Algorithmus als PAG klassifiziert werden, welche im Referenzdatensatz nicht als Anrissflächen vermerkt sind. Das ist der grosse Schwachpunkt dieses simplen Ansatzes. Dies ist auch in den graphischen Darstellungen zu erkennen (Abb. 5.5, Abb. 5.11 und Abb. 5.12). Dadurch resultieren auch die schlechtesten PSS- sowie HSS-Werte im Vergleich mit den anderen Algorithmen.

Die anderen Algorithmen, welche mehr Geländeparameter als nur die Hangneigung beachten, können bei ebenfalls hohem POD-Wert den POFD-Wert merklich verbessern (Tab. 5.2 und Tab. 5.3). Das heisst, die möglichen Anrissflächen können mit Hilfe weiterer Geländeparameter effektiv eingegrenzt werden. Dabei werden Flächen eliminiert, welche aufgrund ihrer Geländeeigenschaften, wie beispielsweise eine hohe Rauigkeit, die Lawinenanrissanfälligkeit hemmen. Dies ist vor allem für Geländekanten, Grate oder für die felsigen Partien der Untersuchungsgebiete zu beobachten. In den graphischen Darstellungen zeigt sich dies beispielsweise für den Bereich östlich des Felsenweges in Gebiet Parsenn in Abb. 5.8 oder für das Leidbachhorn im Gebiet Rinerhorn in Abb. 5.18 für den Algorithmus von Bühler et al. (2018). Dadurch ergeben sich bessere PSS- und HSS-Werte und tiefere FA-Werte im Vergleich zu Voellmy (1955) 28-60°. Für das Trainingsgebiet Parsenn erreicht Bühler et al. (2018) Vorauswahl immer den besten und Bühler et al. (2018) *frequent* den zweitbesten PSS-Wert (Tab. 5.2). Dasselbe gilt für das Testgebiet Jakobs- und Rinerhorn: Bühler et al. (2018) Vorauswahl ergibt immer die besten und Bühler et al. (2018) *frequent* die zweitbesten Validierungswerte (Tab. 5.3). Damit weist der Algorithmus von Bühler et al. (2018) die beste Prognosefähigkeit hinsichtlich zukünftiger Anwendungen auf.

An dieser Stelle werden die Validierungsergebnisse des Algorithmus von Bühler et al. (2018) detaillierter diskutiert: Während des Prozessierungsschrittes der Gebietsabtrennung des Algorithmus von Bühler et al. (2018) werden die einzelnen PAG-Polygone generalisiert und dadurch deren Umrisse, verglichen mit der Vorauswahl, leicht vergrössert (Kapitel 3.1.2). Beim Szenario *extreme* wird stärker generalisiert als beim Szenario *frequent* und darum wird auch der FA-Wert höher (Tab. 5.2 und Tab. 5.3). Deshalb unterscheiden sich die Validierungswerte des Szenarios *extreme* mehr von der Vorauswahl als das Szenario *frequent*. Der Flächenzuwachs für das Szenario *extreme* ist aber beabsichtigt, weil für Extremereignisse mit einer erhöhten Intensität der Lawinenwirkung gerechnet werden muss, was unter anderem durch grössere Anrissflächen bedingt ist (Nairz et al. 2011). Dabei ist zu beobachten, dass das Szenario *extreme* immer etwas schlechtere Validierungswerte als das Szenario *frequent* erreicht. Dies ist folgendermassen zu erklären: Der zur Evaluierung verwendete Referenzdatensatz beinhaltet beobachtete Lawinenabgänge in der Region Davos seit 1970. In Anbetracht dieser begrenzten Zeitspanne muss angenommen werden, dass die darin enthaltenen Lawinen mehrheitlich dem Szenario *frequent* und nur wenige dem Szenario *extreme* zuzuweisen

wären. Demgemäss kann gesagt werden, dass für eine Validierung des Szenarios *extreme* dieser Referenzdatensatz suboptimal ist. Trotzdem ist es fraglich, ob überhaupt ein besserer Referenzdatensatz als der verwendete zur Verfügung steht.

Wie in Kapitel 4.3.2 beschrieben, wurden die Algorithmen anhand des Gebietes Parsenn kalibriert und für das Gebiet Jakobs- und Rinerhorn getestet. Deswegen sollte angenommen werden, dass grundsätzlich bessere Validierungswerte für das Trainingsgebiet Parsenn (Tab. 5.2) als für das Testgebiet Jakobs- und Rinerhorn (Tab. 5.3) resultieren. Diese Annahme wird im Folgendem diskutiert: Bisher wurden die Validierungsergebnisse für das Testgebiet Jakobs- und Rinerhorn (Kapitel 5.2.2) zusammen betrachtet. Um die oben formulierte Annahme detailliert diskutieren zu können, werden anschliessend die Validierungsergebnisse für das Jakobshorn und das Rinerhorn separat aufgelistet.

### Jakobshorn

Die Validierungsergebnisse für das Testgebiet Jakobshorn sind in Tab. 6.1 aufgelistet: Den höchsten POD-Wert erreicht Voellmy (1955) 28-60° und den tiefsten Bühler et al. (2018) Vorauswahl. Gerade umgekehrt ist es hinsichtlich der POFD-Werten: Den tiefsten POFD-Wert erreicht Bühler et al. (2018) Vorauswahl und den höchsten Voellmy (1955) 28-60°. Bezüglich des entscheidenden PSS- und HSS-Wertes zeigt wiederum Bühler et al. (2018) Vorauswahl den höchsten und Voellmy (1955) 28-60° den tiefsten Wert. Bühler et al. (2018) *frequent* weist ebenfalls den zweithöchsten PSS- und HSS-Wert auf. Betreffend des Flächenanteils erzielt Bühler et al. (2018) Vorauswahl den tiefsten Wert.

Tab. 6.1: Validierungsergebnisse aller getesteten Algorithmen für das Testgebiet Jakobshorn

Algorithmus	POD [%]	POFD [%]	PSS [%]	HSS [%]	FA [%]	Kartographische Darstellung
Voellmy (1955) 28-60°	98.27	24.21	74.07	69.17	100.00	Abb. 5.11
Bühler et al. (2013)	95.00	12.52	82.48	80.17	83.97	Abb. 5.13
Veitinger et al. (2016)	97.11	18.49	78.62	74.80	92.45	Abb. 5.15
Bühler et al. (2018) Vorauswahl	94.60	10.04	84.56	82.87	80.70	Abb. 5.17
Bühler et al. (2018) <i>frequent</i>	94.69	10.39	84.31	82.52	81.17	Abb. 5.19
Bühler et al. (2018) <i>extreme</i>	97.55	15.42	82.13	78.78	89.17	Abb. 5.21

### Rinerhorn

Die Validierungsergebnisse für das Testgebiet Rinerhorn sind in Tab. 6.2 aufgelistet: Den höchsten POD-Wert erzielt Bühler et al. (2018) *extreme* und den tiefsten Wert Bühler et al. (2013). Den tiefsten POFD-Wert erreicht Bühler et al. (2013) und den höchsten Wert Voellmy (1955) 28-60°. Bezüglich des dezisiven PSS und HSS zeigt jeweils Bühler et al. (2018) Vorauswahl den höchsten Wert und Voellmy (1955) 28-60° den tiefsten Wert. Hohe PSS- und HSS-Werte erzielen ebenfalls Bühler et al. (2018) *frequent* und Bühler et al. (2013). Hinsichtlich des Flächenanteils erzielt Bühler et al. (2013) den tiefsten Wert.

Tab. 6.2: Validierungsergebnisse aller getesteten Algorithmen für das Testgebiet Rinerhorn

Algorithmus	POD [%]	POFD [%]	PSS [%]	HSS [%]	FA [%]	Kartographische Darstellung
Voellmy (1955) 28-60°	99.02	40.21	58.81	36.00	100.00	Abb. 5.12
Bühler et al. (2013)	96.13	25.46	70.67	50.89	75.93	Abb. 5.14
Veitinger et al. (2016)	98.63	33.95	64.68	42.20	90.06	Abb. 5.16
Bühler et al. (2018) Vorauswahl	97.23	25.63	71.60	51.30	76.54	Abb. 5.18
Bühler et al. (2018) <i>frequent</i>	97.38	26.14	71.24	50.73	77.38	Abb. 5.20
Bühler et al. (2018) <i>extreme</i>	99.34	31.31	68.03	45.55	86.21	Abb. 5.22

Bemerkenswert ist, dass der Algorithmus von Voellmy (1955), Bühler et al. (2013) und vor allem Bühler et al. (2018) bessere PSS- als auch HSS-Werte für das Testgebiet Jakobshorn im Vergleich zum Trainingsgebiet Parsenn erreichen. Dies ist erstaunlich, weil mit den optimalen Parametereinstellungen, kalibriert für Parsenn, im Gebiet Jakobshorn bessere Validierungswerte erreicht werden. Obwohl die Algorithmen für ein bestimmtes Trainingsgebiet kalibriert wurden, konnten sie also auch für ein Testgebiet die Anrissgebiete des Referenzdatensatzes zuverlässig identifizieren. Es ist auffallend, dass die POFD-Werte dieser Algorithmen für das Testgebiet Jakobshorn mit nahezu gleichhohen POD-Werten vergleichsweise tiefer sind als für das Trainingsgebiet Parsenn. Dies könnte folgendermassen erklärt werden: Die Fläche des Gebietes Jakobshorn ist relativ klein und der Perimeter dieses Gebiets ist eng um das Skigebiet gewählt (Abb. 5.11). Die lokalen Skipatrouillen bestätigten, dass die Referenzdaten für dieses Gebiet mit sehr grosser Wahrscheinlichkeit alle potentiellen Anrissgebiete beinhalten. Für das Gebiet Parsenn und vor allem für das Rinerhorn konnten sie dies nicht mit der gleichen Überzeugung behaupten. Es kann daher angenommen werden, dass die tieferen POFD-Werte und die höheren PSS- und HSS-Werte für das Testgebiet Jakobshorn auf einen vollständigeren Referenzdatensatz zurückzuführen sind. Im Vergleich dazu könnten die bei allen Algorithmen höheren POFD-Werte für das Gebiet Rinerhorn durch einen weniger vollständigeren Referenzdatensatz begründbar sein. Eine weitere Erklärung für die relativ hohen POFD-Werte für das Gebiet Rinerhorn sei noch angefügt: Betreffend den geländespezifischen Eigenschaften, vor allem hinsichtlich der mittleren Krümmung und der Rauigkeit (Abb. 4.9), sind sich die Gebiete Parsenn und Jakobshorn ähnlicher als das Rinerhorn. Darum passt wohl die Kalibrierung von Parsenn besser auf das Jakobshorn als auf das Rinerhorn, was im Falle des Rinerhorns zu höheren POFD-Werten führt.

Alle Algorithmen weisen für das Rinerhorn jeweils höhere POFD-Werte als für das Jakobshorn auf. Zum einen ist dies damit zu begründen, dass die Kalibrierung besser für das Jakobshorn als für das Rinerhorn passt. Damit stellt sich die Frage, wie gut sich diese Parametereinstellungen für weitere zukünftige Gebiete eignen. Es wird angenommen, dass, trotz unterschiedlichen Schneesverhältnissen, Anrissgebiete generell durch ähnliche geländespezifische Eigenschaften gekennzeichnet sind (McClung 2001, Schweizer et al. 2003). Die ausgewählten Algorithmen nutzen hauptsächlich das Gelände zur Anrissgebietsidentifizierung und vernachlässigen schneedeckenspezifische Eigenschaften. Darum ist anzunehmen, dass sich diese Parametereinstellungen auch für weitere Anwendungsgebiete eignen und eine universelle Anwendung der Algorithmen möglich ist. Diese Annahme

ist jedoch durch die Anwendung auf weitere Gebiete zu evaluieren. Dafür sind aber geeignete Referenzdatensätze nötig. Dieser Punkt leitet über zum zweiten Aspekt, warum die POFD-Werte für alle vier Algorithmen für das Rinerhorn schlechter als für das Jakobshorn sind: Für Naturgefahren, wie Lawinen, ist aufgrund ihrer Eigenart die Vollständigkeit von Referenzdaten schwierig zu beurteilen. Falscherkennungsfehler sind darum nicht zwingend als Klassifizierungsfehler dieser Flächen zu betrachten. Sie können auch Anrissgebiete darstellen, für welche in der Vergangenheit keine Lawinen beobachtet wurden, folglich nicht im Referenzdatensatz enthalten sind, aber in der Zukunft auftreten könnten. Solche Besonderheiten sind bei der Interpretation der Validierungsergebnisse mit dem verwendeten Evaluierungsverfahren zu berücksichtigen.

## 6.4 Erkenntnisse aus dem Algorithmenvergleich

Bisher wurden die bestehenden Algorithmen zur Anrissgebietsidentifizierung in den jeweiligen Studien zum Teil separat evaluiert, wobei die Idee für einen Algorithmenvergleich bereits geäußert wurde (Ghinoi und Chung 2005). Die langjährigen und detaillierten Lawinenkartierungen des SLFs für die Region Davos eignen sich bestens als Referenzdatengrundlage für einen Algorithmenvergleich (Kapitel 4.1). Diese Arbeit hat erstmals mehrere Algorithmen (Kapitel 3.1.2) mit demselben Referenzdatensatz im Hinblick auf die Gefahrenhinweiskartierung evaluiert und miteinander verglichen. Am besten schnitt der Algorithmus von Bühler et al. (2018) ab. Beim Vergleich diente der simple Ansatz, welcher nur den Geländeparameter der Hangneigung verwendet, als Richtwert für die anderen Algorithmen (Bühler et al. 2013, Veitinger et al. 2016), Bühler et al. (2018).

Aus dem Vergleich lassen sich die folgenden Erkenntnisse formulieren: Grundsätzlich weisen alle untersuchten Algorithmen hohe Erkennungsraten auf. Doch je mehr Geländeparameter ein Algorithmus berücksichtigt, desto tiefer sind die Falscherkennungsraten und desto besser die Validierungsergebnisse der untersuchten Algorithmen. Diese Beobachtungen machten bereits Ghinoi und Chung (2005) sowie Pistocchi und Notarnicola (2013), jedoch nur bei Tests ihres eignen Algorithmus und nicht im Vergleich mehrerer Algorithmen. Zu den verschiedenen Ansätzen der räumlichen Modellierung lässt sich folgendes sagen: Mit Veitinger et al. (2016) war auch ein Algorithmus im Vergleich, welcher *Fuzzy*-Zugehörigkeitsfunktionen anwendet. Dies entspricht eher dem Charakter von natürlichen Phänomenen, in diesem Fall Anrissgebieten, weil damit keine scharfen Abtrennungen, sondern Anrissanfälligkeiten modelliert werden (Kapitel 2.5). Diesen Ansatz verfolgten auch Ghinoi und Chung (2005) (*Fuzzy*) und Pistocchi und Notarnicola (2013) (logistischer Regressionsanalyse) bei der Entwicklung ihrer Algorithmen. Die Anrissgebiete im Referenzdatensatz liegen jedoch als klar abgegrenzte Objekte (Polygone) vor (Kapitel 4.1). Zur Evaluierung sollten deswegen auch die modellierten Anrissgebiete als Polygone vorliegen. Deshalb wurde für den Algorithmus von Veitinger (2015) den optimalen Wert der Anrissanfälligkeit eruiert, der dann als Schwellenwert für die *Boolean*-Zugehörigkeitsfunktion zur Klassierung in PAG und nPAG diente. Obwohl also *Fuzzy* wohl eher dem Charakter eines natürlichen Phänomens entspricht, so mussten trotzdem scharfe Abgrenzungen definiert werden und im Vergleich resultierten für den Ansatz *Fuzzy* (Veitinger et al. 2016) schlechtere Validierungswerte als für zwei *Boolean*-Algorithmen (Bühler et al. 2013, Bühler et al. 2018). Eine weitere Erkenntnis ist die folgende: Mit der Kalibrierung anhand von Referenzdaten wurden die optimalen Parametereinstellungen eruiert. Mit diesen Einstellungen konnten für die

Testgebiete ähnlich gute Validierungsergebnisse wie für das Trainingsgebiet erreicht werden. Das bestätigt die bisherigen Forschungsergebnisse hinsichtlich des lawinenbildenden Faktors des Geländes (Kapitel 2.1.2), dass Anrissgebiete grösstenteils dieselben geländespezifischen Eigenschaften, unabhängig des betrachteten Gebiets, aufweisen. Folglich kann angenommen werden, dass Algorithmen zur Anrissgebietsidentifizierung, basierend auf Geländeparametern, mit kalibrierten Parametereinstellungen auch auf weitere Gebiete angewendet werden können.

Der Algorithmus von Bühler et al. (2018) glättet das DHM um den Einfluss der Schneedecke auf das Gelände zu simulieren. Das Thema der Glättung des Geländes durch die Schneedecke wurde in dieser Arbeit nicht weiter untersucht, ist jedoch bei der Identifizierung potentieller Lawinenanrissgebiete auf Grundlage von digitalen Geländemodellen von Bedeutung. In dieser Arbeit dienten digitale Geländemodelle als Datengrundlage, die das Gelände während der Sommerzeit repräsentieren. Zur Identifizierung potentieller Anrissgebiete von Schneebrettlawinen ist wohl aber zeitlich variable Oberfläche der Schwachschicht massgebend (Kapitel 2.1.2). Deshalb stellt sich die Frage, wie sensitiv sich die Verwendung von verschiedenen Oberflächenmodellen mit oder ohne Schneebedeckung auf die Anrissgebietsidentifizierung auswirkt. Für numerische Lawinensimulation wurde bereits der Einfluss der Verwendung einer Sommer- oder Winteroberflächentopographie untersucht (Maggioni et al. 2013). Im Zusammenhang mit der Anrissgebietsidentifizierung hat Veitinger (2015) erste Untersuchungen zur Geländeglättung durch eine winterliche Schneebedeckung vorgenommen. Die Verfügbarkeit von digitalen Modellen der Winteroberfläche wird durch die anhaltenden Entwicklungen in der Fernerkundung weiter zunehmen (Bühler et al. 2016). Dadurch könnte die Glättung durch die Schneebedeckung und der Einfluss der Verwendung einer Sommer- oder Winteroberflächentopographie auf für die Anrissgebietsidentifizierung in Zukunft umfassend untersucht werden.

Ausserdem bieten zwei Algorithmen (Bühler et al. 2013, Bühler et al. 2018) die Möglichkeit, räumliche Informationen zur Waldbedeckung in der Anrissgebietsidentifizierung zu berücksichtigen. In dieser Arbeit war das aber nicht nötig, weil sich innerhalb des Perimeters des Untersuchungsgebietes kein Wald befindet. Deshalb wurde die Rolle der Waldes im Zusammenhang mit der Lawinenauslösung nicht weiter thematisiert, auch deshalb, weil es bereits umfassende Studien dazu gibt (McClung 2001, Bebi et al. 2009, Teich und Bebi 2009, Viglietti et al. 2010).

In dieser Arbeit wurde ausschliesslich die von einem Algorithmus ausgeschiedene PAG-Fläche mit Lawinenanrissflächen eines Referenzdatensatzes verglichen. Möchte man die automatische Anrissgebietsidentifizierung in Kombination mit dynamischer Lawinensimulationssoftware zur automatischen Gefahrenhinweiskartierung verwenden, so ist eine Abtrennung in einzelne Polygone nötig. Gleichwohl sollte dabei beachtet werden, dass eine solch scharfe Abgrenzung in einzelne Polygone eine starke Abstrahierung der Realität ist (Kapitel 2.5). Die Abtrennung der Flächen in individuelle Gebiete wurde nicht evaluiert. Eine explizite Abtrennung in Polygone nehmen von den getesteten Algorithmen ausschliesslich Bühler et al. (2013) und Bühler et al. (2018) vor. Die vernünftige Abtrennung in einzelne Gebiete bedingt die Definition von Gefahrenszenarien mit bestimmten Eintrittswahrscheinlichkeiten (Kapitel 2.2 und Kapitel 3.1.2). Die Abtrennung von Bühler et al. (2013) ist jedoch nicht szenarienspezifisch und zudem bewerten SLF-Experten diese Abtrennungen zum

Teil als wenig realistisch (Abb. 5.6, Abb. 5.13 und Abb. 5.14). Erste Anwendungen (Bühler et al. 2017) des Algorithmus von Bühler et al. (2018) in Kombination mit dynamischer Lawinensimulationen ergaben aber vielversprechende Resultate (Abb. 6.1). Es wurde in Erwägung gezogen, ob mit dem bestehenden Referenzdatensatz auch eine szenarienspezifische Gebietsabtrennung validierbar wäre. Erste Versuche zeigten jedoch, dass der Referenzdatensatz noch zu wenig umfassend ist, um insbesondere Szenarien mit geringen Eintrittswahrscheinlichkeiten (beispielsweise das Szenario *extreme*) zu validieren.

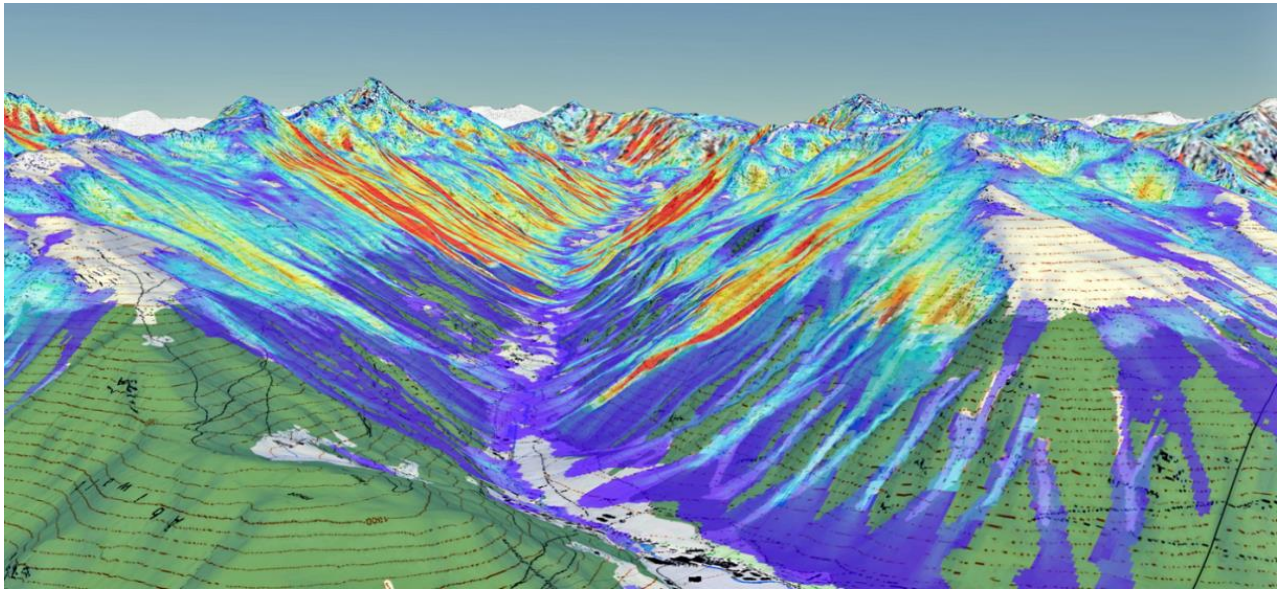


Abb. 6.1: Numerische Lawinensimulationen auf Basis automatisch identifizierter Lawinenanrissgebiete [#\*]

## 7 Schlussfolgerungen

Der Algorithmenvergleich hat gezeigt: Alle untersuchten Algorithmen erreichen sehr hohe Erkennungsraten (POD-Werte von nahezu 100%). Doch der einfache Ansatz nach Voellmy (1955), der nur die Hangneigung nutzt, weist zugleich hohe Falscherkennungsraten (POFD-Werte) auf. Fazit: Je mehr Geländeparameter einbezogen werden, desto tiefer ist tendenziell die Falscherkennungsrate und desto bessere Validierungsergebnisse weisen die verglichenen Algorithmen auf. Am besten schnitt dabei der Algorithmus von Bühler et al. (2018) ab.

Die verglichenen Algorithmen zur Identifizierung potentieller Anrissgebiete basieren hauptsächlich auf Geländeparametern und wenden einfache Zugehörigkeitsfunktion an (Kapitel 3.1.1). Dies hat den Vorteil, dass ein Geländemodell die einzig zwingend nötige Datengrundlage ist und eine Anrissgebietsidentifizierung für grosse Gebiete in kurzer Rechenzeit erlaubt. Dabei werden die Geländeparameter als statisch, das heisst über die Zeit konstant, angesehen. Es stellt sich jedoch die Frage, wie statisch diese Geländeparameter während eines Winters wirklich sind, wenn zunehmende Schneemengen in der Landschaft kleinräumige Geländemerkmale verändern, die Oberflächenrauigkeit ausgleichen und damit die Hangneigung reduzieren können. Eine Glättung des DHMs, wie es für den Algorithmus von Bühler et al. (2018) angewandt wird, versucht dem Rechnung zu tragen. Die besten Validierungsergebnisse von Bühler et al. (2018) im Vergleich zu den anderen Algorithmen bekräftigen, dass für die Anrissgebietsidentifizierung ein Sommer-DHM für eine winterliche Schneeoberfläche angepasst werden sollte. Diese einfache Glättung ist aber keinesfalls eine wissenschaftlich fundierte Anpassung des DHMs für eine winterliche Schneebedeckung. Veitinger (2015) hat für wenige Beispiele den Zusammenhang zwischen dem Sommergelände und der winterlichen Schneeoberfläche untersucht, doch ist in diesem Bereich wohl noch weitere Forschung erforderlich. Dadurch wäre eine wissenschaftlich fundierte Anpassung von Sommer-DHMs für winterliche Schneeoberflächen möglich.

Die mangelnde Verfügbarkeit und ungewisse Vollständigkeit von Referenzdaten wurde wiederholt thematisiert. Für die weitere Forschung im Bereich der automatischen Identifizierung potentieller Anrissgebiete ist deshalb die genaue und systematische Aufzeichnung sowie Kartierung von Lawineneignissen von grosser Wichtigkeit. Insbesondere die detaillierte Kartierung der Anrissgebiete wäre in diesem Zusammenhang wünschenswert. Durch noch bessere Referenzdaten könnte versucht werden, neben der Flächenausscheidung auch die Gebietsabtrennung für verschiedene Gefahrenszenarios zu evaluieren. Mit dem für diese Arbeit zur Verfügung stehenden Referenzdatensatz war das quantitativ nicht möglich.

Diese Arbeit beschränkte sich auf die Identifizierung potentieller Lawinenanrissgebiete. Das Fernziel ist jedoch die automatische Gefahrenhinweiskartierung. Da auch für alpine Regionen immer mehr räumlich hochaufgelöste DHM-Daten zur Verfügung stehen, hat der Algorithmus von Bühler et al. (2018) mit seiner Gebietsabtrennung das Potenzial mit numerischen Lawinensimulationen kombiniert zu werden (Abb. 6.1). Dies ermöglicht die automatisierte Gefahrenhinweiskartierung für grossflächige Gebiete. Dadurch kann beispielsweise die Schutzfunktion von Wäldern und damit deren Nutzen automatisch abgeschätzt werden. Für Gebiete ohne jegliche Lawinenkataster können



damit erste Gefahrenabschätzungen vorgenommen werden. Dies ist eine grosse Hilfe bei der Planung neuer Infrastrukturen und der Evaluierung von bereits bestehenden Schutzmassnahmen von Siedlungen und Verkehrswegen. Solche Simulationen können auch mit einer Geländeklassifizierung zu einem neuen Produkt für die Skitourenplanung kombiniert werden. Diese exemplarischen Anwendungen sind bereits heute möglich. Über die Qualität der automatischen Gefahrenhinweiskartierung könnten quantitative Vergleiche mit bestehenden Gefahrenkarten Aufschluss geben. Dabei sei gesagt: Diese automatischen Verfahren können nur Gefahrenhinweise liefern. Für Fälle, in denen eine detaillierte Studie nötig ist, wie beispielsweise bei der Erstellung von Gefahrenkarten, ist eine Expertenbeurteilung nötig.

Um allerdings situationsspezifische Gefahrenhinweise automatisch vornehmen zu können, könnten die Algorithmen noch erweitert werden, um neben dem statischen Gelände auch zeitlich variable Informationen zur Vegetation, Schneedecke oder zum Wetter miteinzubeziehen. Ghinoi und Chung (2005) konnten bereits zeigen, dass je nach meteorologischem Szenario Geländeparameter zur Anrissgebietsidentifizierung verschieden gewichtet werden sollten. Durch die Kopplung von aktuellen Informationen zum Gelände, der Vegetation, der Schneedecke, des Wetters und der numerischen Lawinensimulationen könnte eine nahezu zeitnahe Gefährdungsbeurteilung über grosse Flächen automatisch vorgenommen werden. Die Verfügbarkeit der dafür benötigten Informationen, in ausreichender Qualität und Auflösung, die für ein solch gekoppeltes System notwendig wäre, ist jedoch zurzeit (noch) sehr begrenzt. Kurzum: Die Forschungs- und Entwicklungsmöglichkeiten bezüglich automatischer Identifizierung potentieller Lawinenanrissgebiete sind bisher noch nicht ausgeschöpft.

## 8 Literatur

- AG NAGEF. 2015. *Klimawandel und Naturgefahren –Veränderungen im Hochgebirge des Berner Oberlandes und ihre Folgen*. Arbeitsgruppe Naturgefahren des Kantons Bern.
- Allouche, O., A. Tsoar, R. Kadmon. 2006. Assessing the accuracy of species distribution models: prevalence, kappa and the true skill statistic (TSS). *Journal of Applied Ecology* 43/6: 1223-1232.
- Andres, A. J., J. Chueca Cí'a. 2012. Mapping of avalanche start zones susceptibility: Arazas basin, Ordesa and Monte Perdido National Park (Spanish Pyrenees). *Journal of Maps* 8/1: 14-21.
- Asharyanto, H., B. Soeksmantono, K. Wikantika. 2015. *Three dimensional city building modelling with LIDAR data (case study: ciwaruga, bandung)*.
- Auer, M. 2005. *Überarbeitung der Periodizität der Schneehöhen im schweizerischen Alpengebiet*. Davos: SLF.
- Baatz, M., A. Schäpe. 2000. Multiresolution Segmentation: an optimization approach for high quality multi-scale image segmentation. Vortrag auf der *Angewandte Geographische Informationsverarbeitung XII. Beiträge zum AGIT-Symposium Salzburg 2000*. Salzburg, Österreich.
- Barbolini, M., M. Pagliardi, F. Ferro, P. Corradeghini. 2011. Avalanche hazard mapping over large undocumented areas. *Natural Hazards* 56/2: 451-464.
- Bebi, P., D. Kulakowski, C. Rixen. 2009. Snow avalanche disturbances in forest ecosystems—State of research and implications for management. *Forest Ecology and Management* 257/9: 1883-1892.
- Beguiría, S. 2006. Validation and Evaluation of Predictive Models in Hazard Assessment and Risk Management. *Natural Hazards* 37/3: 315-329.
- Blaschke, T. 2010. Object based image analysis for remote sensing. *ISPRS Journal of Photogrammetry and Remote Sensing* 65/1: 2-16.
- Bovet, S. 2013. Das DTM der Schweiz: für ihre Sicherheit neu bearbeitet. *Geomatik Schweiz : Geoinformation und Landmanagement* 111/9: 5.
- Bründl, M., H. Romang. 2009. Teil B: Anwendung des Risikokonzepts Prozess Lawine. Bern: Nationale Plattform Naturgefahren PLANAT.
- Bühler, Y., M. Marty, C. Ginzler. 2012. High Resolution DEM Generation in High-Alpine Terrain Using Airborne Remote Sensing Techniques. *Transactions in GIS* 16/5: 635-647.
- Bühler, Y., M. Christen, J. Kowalski, P. Bartelt. 2011. Sensitivity of snow avalanche simulations to digital elevation model quality and resolution. *Annals of Glaciology* 52/58.

- Bühler, Y., M. S. Adams, R. Bösch, A. Stoffel. 2016. Mapping snow depth in alpine terrain with unmanned aerial systems (UASs): potential and limitations. *The Cryosphere* 10/3: 1075-1088.
- Bühler, Y., A. Hüni, M. Christen, R. Meister, T. Kellenberger. 2009. Automated detection and mapping of avalanche deposits using airborne optical remote sensing data. *Cold Regions Science and Technology* 57/2: 99-106.
- Bühler, Y., S. Kumar, J. Veitinger, M. Christen, A. Stoffel, Snehmani. 2013. Automated identification of potential snow avalanche release areas based on digital elevation models. *Natural Hazards and Earth System Science* 13/5: 1321-1335.
- Bühler, Y., D. von Rickenbach, A. Stoffel, S. Margreth, L. Stoffel, M. Christen. 2018. Automated snow avalanche release area delineation – validation of existing algorithms and proposition of a new object-based approach for large scale hazard indication mapping. *Nat. Hazards Earth Syst. Sci. Discuss.* 2018: 1-24.
- Bühler, Y. et al. 2017. *Vorprojekt Gefahrenhinweiskarte Lawinen Kanton Graubünden*. Davos: WSL-Institut für Schnee- und Lawinenforschung SLF.
- Burrough, P. A., R. A. McDonnell, C. D. Lloyd. 2015. *Principle of Geographic Information Systems*. Oxford: Oxford University Press.
- Camenzind-Wildi, R., R. Baumann, C. Guggisberg, R. Loat, I. Diethelm. 2000. Empfehlung Raumplanung und Naturgefahren. Bern: Bundesamt für Raumentwicklung, Bundesamt für Wasser und Geologie, Bundesamt für Umwelt, Wald und Landschaft.
- Carrara, A. 1983. Multivariate models for landslide hazard evaluation. *Journal of the International Association for Mathematical Geology* 15/3: 403-426.
- Carrara, A., F. Guzzetti, C. Lomnitz. 1995. *Geographical information systems in assessing natural hazards*. Advances in Natural and Technological Hazards Research. Band 5. New York: Springer Science+Business Media.
- Cascini, L. 2008. Applicability of landslide susceptibility and hazard zoning at different scales. *Engineering Geology* 102/3: 164-177.
- Christen, M., J. Kowalski, P. Bartelt. 2010. RAMMS: Numerical simulation of dense snow avalanches in three-dimensional terrain. *Cold Regions Science and Technology* 63/1-2: 1-14.
- Chueca Cía, J., A. J. Andrés, A. Montañés Magallón. 2014. A proposal for avalanche susceptibility mapping in the Pyrenees using GIS: the Formigal-Peyreget area (Sheet 145-I; scale 1:25.000). *Journal of Maps* 10/2: 203-210.
- Cohen, J. 1960. A Coefficient of Agreement for Nominal Scales. *Educational and Psychological Measurement* 20/1: 37-46.

- Congalton, R. G., K. Green. 2009. *Assessing the Accuracy of Remotely Sensed Data, Principles and Practices*. Band 2. Boca Raton: CRC Press.
- Congalton, R. G., R. G. Oderwald, R. A. Mead. 1983. Assessing Landsat classification accuracy using discrete multivariate analysis statistical techniques. *Photogrammetric Engineering & Remote Sensing* 49/12: 1671-1678.
- Dikau, R. 1989. The Application of a Digital Relief Model to Landform Analysis in Geomorphology. In: *Three Dimensional Applications in Geographical Information Systems*. Herausgegeben von J. F. Raper. London: Taylor & Francis. 57-777.
- Dikau, R., H. Voss. 2000. Lexikon der Geowissenschaften. *Naturkatastrophe*. Heidelberg: Spektrum Akademischer Verlag.
- Doolittle, M. H. 1888. Association ratios. *Bulletin of the Philosophical Society, Washington* 7: 122-127.
- Eckerstorfer, M., E. Malnes, K. Müller. 2017. A complete snow avalanche activity record from a Norwegian forecasting region using Sentinel-1 satellite-radar data. *Cold Regions Science and Technology* 144: 39-51.
- Eckerstorfer, M., Y. Bühler, R. Frauenfelder, E. Malnes. 2016. Remote sensing of snow avalanches: Recent advances, potential, and limitations. *Cold Regions Science and Technology* 121: 126-140.
- ESRI. 2018a. *Resampling*. <https://pro.arcgis.com/de/pro-app/tool-reference/data-management/resample.htm> (abgerufen 23.04.2018).
- ESRI. 2018b. *Mapping and visualization in ArcGIS for Desktop*. <http://desktop.arcgis.com/en/arcmap/10.3/main/map/mapping-and-visualization-in-arcgis-for-desktop.htm> (abgerufen 09.04.2018).
- ESRI. 2018c. *An overview of the Hydrology toolset*. <http://desktop.arcgis.com/en/arcmap/10.3/tools/spatial-analyst-toolbox/an-overview-of-the-hydrology-tools.htm> (abgerufen 29.05.2018).
- EVA+. 2018. *Extreme Value Analysis enhanced*. <http://eva.zamg.ac.at/evaplus/> (abgerufen 27.06.2018).
- Fonstad, M. A., J. T. Dietrich, B. C. Courville, J. L. Jensen, P. E. Carbonneau. 2013. Topographic structure from motion: a new development in photogrammetric measurement. *Earth Surface Processes and Landforms* 38/4: 421-430.
- Frattini, P., G. Crosta, A. Carrara. 2010. Techniques for evaluating the performance of landslide susceptibility models. *Engineering Geology* 111/1-4: 62-72.
- Frauenfelder, R., M. J. Lato, M. Biskupič. 2015. Using eCognition to automatically detect and map avalanche deposits from the spring 2009 avalanche cycle in the Tatra Mts., Slovakia.

Vortrag auf der *International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences - ISPRS Archives*.

- Gariano, S. L. et al. 2015. Calibration and validation of rainfall thresholds for shallow landslide forecasting in Sicily, southern Italy. *Geomorphology* 228: 653-665.
- Ghini, A., C.-J. Chung. 2005. STARTER: a statistical GIS-based model for the prediction of snow avalanche susceptibility using terrain features—application to Alta Val Badia, Italian Dolomites. *Geomorphology* 66/1-4: 305-325.
- Gleason, J. A. 1994. Terrain parameters of avalanche starting zones and their effect on avalanche frequency. Vortrag auf der *International Snow Science Workshop*. Snowbird, Utah, USA.
- Gruber, S., C. Huggel, R. Pike. 2009. Chapter 23 Modelling Mass Movements and Landslide Susceptibility. In: *Developments in Soil Science*. Herausgegeben von Tomislav Hengl, Hannes I. Reuter. Band 33. Elsevier. 527-550.
- Gruber Schmid, U., S. Peckham. 2009. Chapter 7 Land-Surface Parameters and Objects in Hydrology. In: *Developments in Soil Science*. Herausgegeben von Tomislav Hengl, Hannes I. Reuter. Band 33. Elsevier. 171-194.
- Gruber, U., S. Margreth. 2001. Winter 1999: A valuable test of the avalanche-hazard mapping procedure in Switzerland. *Annals of Glaciology* 3: 328-332.
- Hanssen, A. W., W. J. A. Kuipers. 1965. *On the Relationship Between the Frequency of Rain and Various Meteorological Parameters: (with Reference to the Problem of Objective Forecasting)*. The Hague: Staatsdrukkerij-en Uitgeverijbedrijf.
- Harvey, S., H. Ryhner, J. Schweizer. 2013. *Lawinenkunde - Praxiswissen für Einsteiger und Profis zu Gefahren, Risiken und Strategien*. Bruckmann Verlag GmbH.
- Harvey, S., G. Schudlach, Y. Bühler, L. Dürr, A. Stoffel, M. Christen. 2018a. Automatic high-resolution mapping and classification of avalanche terrain. *in prep.*
- Harvey, S., G. Schudlach, Y. Bühler, L. Dürr, A. Stoffel, M. Christen. 2018b. Avalanche terrain maps for backcountry skiing in Switzerland. Vortrag auf der *International Snow Science Workshop*. Innsbruck.
- Heidemann, H. K. 2018. Lidar base specifications. In: *U.S. Geological Survey Techniques and Methods*. Band book 11. 101.
- Heidke, P. 1926. Berechnung Des Erfolges Und Der Güte Der Windstärkevorhersagen Im Sturmwarnungsdienst. *Geografiska Annaler* 8/4: 301-349.
- Heierli, J., R. S. Purves, A. Felber, J. Kowalski. 2004. Verification of nearest-neighbours interpretations in avalanche forecasting. *Annals of Glaciology* 38: 84-88.

- Hengl, T., R. A. MacMillan. 2009. Chapter 19 Geomorphometry — A Key to Landscape Mapping and Modelling. In: *Developments in Soil Science*. Herausgegeben von Tomislav Hengl, Hannes I. Reuter. Band 33. Elsevier. 433-460.
- Hengl, T., I. S. Evans. 2009. Chapter 2 Mathematical and Digital Models of the Land Surface. In: *Developments in Soil Science*. Herausgegeben von Tomislav Hengl, Hannes I. Reuter. Band 33. Elsevier. 31-63.
- Horn, B. K. 1981. Hill shading and the reflectance map. *Proceedings of the IEEE* 69/1: 14-47.
- Jenness, J. 2013. DEM Surface Tools for ArcGIS. *An ArcGIS Extension for Analyzing Raster Elevation Datasets*: Jenness Enterprises 98.
- Kanton Graubünden. 2010. Kurzinfo Ereigniskataster Naturgefahren / StorMe: Amt für Wald und Naturgefahren, Tiefbauamt Graubünden, Rhätische Bahn 3.
- Khavrus, V., I. Shelevytsky. 2012. Geometry and the physics of seasons. *Physics Education* 47: 680-692.
- Kienholz, H., P. Zeilstra, K. Hollenstein. 1998. Begriffsdefinitionen zu den Themen Geomorphologie, Naturgefahren, Forstwesen, Sicherheit und Risiko. Bern: BUWAL Bundesamt für Umwelt, Wald und Landschaft, Eidg. Forstdirektion.
- Korzeniowska, K., Y. Bühler, M. Marty, O. Korup. 2017. Regional snow-avalanche detection using object-based image analysis of near-infrared aerial imagery. *Natural Hazards and Earth System Sciences* 17/10: 1823-1836.
- Landinfo Worldwide Mapping, L. 2018. *Satellite Imagery Pricing*. <http://www.landinfo.com/satellite-imagery-pricing.html> (abgerufen 21.08.2018).
- Lato, M. J., R. Frauenfelder, Y. Bühler. 2012. Automated detection of snow avalanche deposits: Segmentation and classification of optical remote sensing imagery. *Natural Hazards and Earth System Science* 12/9: 2893-2906.
- Leonarduzzi, E., P. Molnar, B. W. McArdell. 2017. Predictive performance of rainfall thresholds for shallow landslides in Switzerland from gridded daily data. *Water Resources Research* 53/8: 6612-6625.
- Losey, S. 2013. SilvaProtect-CH Prozessmodellierung. *Anhang 1*. Bern: Eidgenössisches Departement für Umwelt, Verkehr, Energie und Kommunikation. Bundesamt für Umwelt BAFU, Abteilung Gefahrenprävention.
- Maggioni, M. 2005. *Avalanche release areas and their influence on uncertainty in avalanche hazard mapping*. Universität Zürich.
- Maggioni, M., U. Gruber. 2003. The influence of topographic parameters on avalanche release dimension and frequency. *Cold Regions Science and Technology* 37/3: 407-419.

- Maggioni, M., U. Gruber Schmid, A. Stoffel. 2002. Definition and characterisation of potential avalanche release areas. Vortrag auf der *Proceedings of the ESRI Conference*. San Diego.
- Maggioni, M. et al. 2013. Influence of summer and winter surface topography on numerical avalanche simulations. Vortrag auf der *International Snow Science Workshop*. Grenoble – Chamonix Mont-Blanc.
- Margreth, S. 2004. Die Wirkung des Waldes bei Lawinen. Vortrag auf der *Forum für Wissen*. Birmensdorf.
- Martinec, J. 1990. Schneehöhen und Wasseräquivalente der Schneedecke. *Hydrologischer Atlas der Schweiz*. Wabern, Bern: Bundesamt für Landestopographie.
- McClung, D., P. A. Schaerer. 1993. *The Avalanche Handbook*. Seattle, WA: The Mountaineers.
- McClung, D. M. 2001. Characteristics of terrain, snow supply and forest cover for avalanche initiation caused by logging. *Annals of Glaciology* 32: 223-229.
- Michoud, C. et al. 2012. Rockfall hazard and risk assessments along roads at a regional scale: example in Swiss Alps. *Natural Hazards and Earth System Science* 12/3: 615-629.
- Nairz, P., S. Sauer Moser, K. Kleemayr, K. Gabl, M. Stoffel, S. Margreth. 2011. Lawinen Entstehung und Wirkung. In: *Handbuch Technischer Lawinenschutz*. Herausgegeben von Florian Rudolf-Miklau, Siegfried Sauer Moser. Berlin: Wilhelm Ernst & Sohn, Verlag für Architektur und technische Wissenschaften GmbH & Co. KG. 21-62.
- Olaya, V. 2009. Chapter 6 Basic Land-Surface Parameters. In: *Developments in Soil Science*. Herausgegeben von Tomislav Hengl, Hannes I. Reuter. Band 33. Elsevier. 141-169.
- Pappenberger, F., K. Frodsham, K. Beven, R. Romanowicz, P. Matgen. 2007. Fuzzy set approach to calibrating distributed flood inundation models using remote sensing observations. *Hydrol. Earth Syst. Sci.* 11/2: 739-752.
- Peirce, C. S. 1884. The numerical measure of the success of predictions. *Science* ns-4/93: 453.
- Phillips, M., A. Haberkorn, H. Rhyner. 2017. Snowpack characteristics on steep frozen rock slopes. *Cold Regions Science and Technology* 141: 54-65.
- Pike, R. J., I. S. Evans, T. Hengl. 2009. Chapter 1 Geomorphometry: A Brief Guide. In: *Developments in Soil Science*. Herausgegeben von Tomislav Hengl, Hannes I. Reuter. Band 33. Elsevier. 3-30.
- Pistocchi, A., C. Notarnicola. 2013. Data-driven mapping of avalanche release areas: a case study in South Tyrol, Italy. *Natural Hazards* 65/3: 1313-1330.
- Pradhan, B., M. Buchroithner. 2012. *Terrigenous Mass Movements: Detection, Modelling, Early Warning and Mitigation Using Geoinformation Technology*. Berlin Heidelberg: Springer Science & Business Media.

- Purves, R. S., K. W. Morrison, G. Moss, D. S. B. Wright. 2003. Nearest neighbours for avalanche forecasting in Scotland—development, verification and optimisation of a model. *Cold Regions Science and Technology* 37/3: 343-355.
- R-Project. 2018. *The R Project for Statistical Computing*. <https://www.r-project.org/> (abgerufen 09.04.2018).
- Rudolf-Miklau, F. et al. 2011. *Handbuch Technischer Lawinenschutz*. Berlin: Wilhelm Ernst & Sohn, Verlag für Architektur und technische Wissenschaften GmbH & Co. KG.
- Rykiel, E. J. 1996. Testing ecological models: the meaning of validation. *Ecological Modelling* 90/3: 229-244.
- Sampl, P., T. Zwinger. 2004. Avalanche simulation with SAMOS. *Annals of Glaciology* 3: 393-398.
- Sappington, J. M., K. M. Longshore, D. B. Thompson. 2007. Quantifying Landscape Ruggedness for Animal Habitat Analysis: A Case Study Using Bighorn Sheep in the Mojave Desert. *Journal of Wildlife Management* 71/5: 1419-1426.
- Schneebeli, M., P. Bebi. 2004. Snow and Avalanche Control. In: *Encyclopedia of Forest Sciences*. Herausgegeben von Jeffery Burley, Julian Evans, John A. Youngquist. Oxford: Elsevier. 397-402.
- Schüepp, M., C. Urfer. 1962. Die Windverhältnisse im Davoser Hochtal. *Archiv für Meteorologie, Geophysik und Bioklimatologie, Serie B* 12/2: 337-349.
- Schweizer, J., J. B. Jamieson, M. Schneebeli. 2003. Snow avalanche formation. *Reviews of Geophysics* 41/4: 25.
- Shcherbakov, R., D. L. Turcotte, J. B. Rundle, K. F. Tiampo, J. R. Holliday. 2010. Forecasting the Locations of Future Large Earthquakes: An Analysis and Verification. *Pure and Applied Geophysics* 167/6: 743-749.
- Singh, L. P., C. J. van Westen, P. K. Champati Ray, P. Pasquali. 2005. Accuracy assessment of InSAR derived input maps for landslide susceptibility analysis: a case study from the Swiss Alps. *Landslides* 2/3: 221-228.
- WSL-Institut für Schnee- und Lawinenforschung SLF. (Hrsg.) 2000. *Der Lawinenwinter 1999. Ereignisanalyse*. Davos: WSL-Institut für Schnee- und Lawinenforschung SLF.
- SLF. 2018a. *Schneedecken-modellierung – SNOWPACK*. <https://www.slf.ch/de/services-und-produkte/snowpack.html> (abgerufen 05.04.2018).
- SLF. 2018b. *Lawinenarten*. <https://www.slf.ch/de/lawinen/lawinenkunde-und-praevention/lawinenarten.html#tabelement1-tab2> (abgerufen 08.03.2018).
- Statham, G. et al. 2017. A conceptual model of avalanche hazard. *Natural Hazards* 90/2: 663-691.



- Stehman, S. V. 1997. Selecting and interpreting measures of thematic classification accuracy. *Remote Sensing of Environment* 62/1: 77-89.
- Stehman, S. V., J. D. Wickham. 2011. Pixels, blocks of pixels, and polygons: Choosing a spatial unit for thematic accuracy assessment. *Remote Sensing of Environment* 115/12: 3044-3055.
- Stross-Radschinski, A. C. 2018. python - a programming language changes the world. *Case Studies & Success Stories*. 1: 32.
- Studio, R. 2018. *Take control of your R code*. <https://www.rstudio.com/products/RStudio/> (abgerufen 09.04.2018).
- swisstopo. 2018a. *swissALTI3D Das hoch aufgelöste Terrainmodell der Schweiz*. Bern: Bundesamt für Landestopografie swisstopo.
- swisstopo. 2018b. *Beschaffung von LiDAR-Daten*. <https://www.swisstopo.admin.ch/de/wissen-fakten/geoinformation/lidar-daten.html> (abgerufen 14.05.2018).
- Techel, F. et al. 2016. Avalanche fatalities in the European Alps: long-term trends and statistics. *Geographica Helvetica* 71/2: 147-159.
- Teich, M., P. Bebi. 2009. Evaluating the benefit of avalanche protection forest with GIS-based risk analyses—A case study in Switzerland. *Forest Ecology and Management* 257/9: 1910-1919.
- Trimble. 2018. *eCognition Developer 9*. <http://www.ecognition.com/suite/ecognition-developer> (abgerufen 09.04.2018).
- UNESCO. 1981. *Lawinen-Atlas Bebilderte Internationale Lawinenklassifikation. Natur-Katastrophen. Switzerland: Internationale Kommission für Schnee und Eis der Internationalen Assoziation für Hydrologische Wissenschaften*.
- van Herwijnen, A., J. Heierli. 2009. Measurement of crack-face friction in collapsed weak snow layers. *Geophysical Research Letters* 36/23.
- Veitinger, J. 2015. *Release Areas of Snow Avalanches: New Methods and Parameters*. Dissertation, Universität Zürich.
- Veitinger, J., R. S. Purves, B. Sovilla. 2016. Potential slab avalanche release area identification from estimated winter terrain: a multi-scale, fuzzy logic approach. *Natural Hazards and Earth System Sciences* 16/10: 2211-2225.
- Viglietti, D., S. Letey, R. Motta, M. Maggioni, M. Freppaz. 2010. Snow avalanche release in forest ecosystems: A case study in the Aosta Valley Region (NW-Italy). *Cold Regions Science and Technology* 64/2: 167-173.
- Voellmy, A. 1955. Über die Zerstörungskraft von Lawinen. *Schweizerische Bauzeitung* 73/15: 212-217.

- Vontobel, I. 2011. *Geländeanalysen von Unfalllawinen*. Masterarbeit, Universität Zürich.
- Wilks, D. S. 2006. *Statistical methods in atmospheric sciences*. International geophysics series. 2nd ed Auflage. Amsterdam: Elsevier.
- Wilks, D. S. 2011a. Chapter 7 - Statistical Forecasting. In: *International Geophysics*. Herausgegeben von Daniel S. Wilks. Band 100. Academic Press. 215-300.
- Wilks, D. S. 2011b. Chapter 8 - Forecast Verification. In: *International Geophysics*. Herausgegeben von Daniel S. Wilks. Band 100. Academic Press. 301-394.
- Wirz, V., M. Schirmer, S. Gruber, M. Lehning. 2011. Spatio-temporal measurements and analysis of snow depth in a rock face. *The Cryosphere* 5/4: 893-905.
- Zevenbergen, L. W., C. R. Thorne. 1987. Quantitative analysis of land surface topography. *Earth Surface Processes and Landforms* 12/1: 47-56.
- Zweifel, B. 2017. Anleitung Lawinenkartierung. Davos: SLF 3.

# 9 Anhang

## 9.1 Python-Skripte

### 9.1.1 Umsetzung Voellmy (1955)

```
# -*- coding: utf-8 -*-
# -----
# Slope.py
# Created on: 2017-10-22 21:08:57.00000
# (generated by ArcGIS/ModelBuilder)
# Usage: Slope <DEM> <Forest>
# Description:
# -----

# Set the necessary product code
# Import arcinfo
import os

# Import arcpy module
import arcpy

# Script arguments
WorkDir = arcpy.GetParameterAsText(0) + "\\\"
MethodRun = arcpy.GetParameterAsText(1)
DEM = arcpy.GetParameterAsText(2)
SlopeMin = arcpy.GetParameterAsText(3)
SlopeMax =arcpy.GetParameterAsText(4)
Forest = arcpy.GetParameterAsText(5)
AreaMin = arcpy.GetParameterAsText(6)
Perimeter = arcpy.GetParameterAsText(7)
PRA_Referenz = arcpy.GetParameterAsText(8)

# Local variables:
WorkDir_temp_model = WorkDir + "\\temp_model\\"
Slope_Deg = WorkDir_temp_model + "Slope_Deg.tif"
Slope_Min_Max_tif = WorkDir_temp_model + "Slope_Min_Max.tif"
Slope_Min_Max_Forest_tif = WorkDir_temp_model + "Slope_Min_Max_Forest.tif"
Slope_Min_Max_Forest_Reclass = WorkDir_temp_model + "Slope_Min_Max_Forest_Reclass.tif"
Slope_Min_Max_Forest_shp = WorkDir_temp_model + "Slope_Min_Max_Forest.shp"
Slope_Min_Max_Forest_Area = WorkDir_temp_model + "Slope_Min_Max_Forest_Area.shp"
Slope_Min_Max_Forest_Area_Smooth = WorkDir_temp_model +
"Slope_Min_Max_Forest_Area_Smooth.shp"
```

```

# Process: Create Folder
arcpy.CreateFolder_management(WorkDir, "temp_model")

# Process: Slope
arcpy.gp.Slope_sa(DEM, Slope_Deg, "DEGREE", "1")

# Process: Raster Calculator
arcpy.gp.RasterCalculator_sa("SetNull(\ "%s\ "<%s,1) & SetNull(\ "%s\ ">%s,1)" %
(Slope_Deg,SlopeMin,Slope_Deg,SlopeMax), Slope_Min_Max_tif)

# Process: Raster Calculator
arcpy.gp.RasterCalculator_sa("'"%s" + "%s" % (Slope_Min_Max_tif,Forest),
Slope_Min_Max_Forest_tif)

arcpy.RasterToPolygon_conversion(Slope_Min_Max_Forest_tif, Slope_Min_Max_Forest_shp,
"NO_SIMPLIFY")

#####
#####

# Process: Add Geometry Attributes
arcpy.AddGeometryAttributes_management(Slope_Min_Max_Forest_shp, "AREA;AREA_GEODESIC",
"METERS", "SQUARE_METERS", "PRO-
JCS['CH1903_LV03',GEOGCS['GCS_CH1903',DATUM['D_CH1903',SPHEROID['Bessel_1841',6377397.1
55,299.1528128]],PRIMEM['Greenwich',0.0],UNIT['Degree',0.0174532925199433]],PROJECTION['H
otine_Oblique_Mercator_Azimuth_Center'],PARAMETER['False_Easting',600000.0],PARAMETER['Fal
se_Northing',200000.0],PARAMETER['Scale_Factor',1.0],PARAMETER['Azimuth',90.0],PARAMETER['
Longi-
tude_Of_Center',7.439583333333333],PARAMETER['Latitude_Of_Center',46.95240555555556],UNI
T['Meter',1.0]]")

# Process: Select
arcpy.Select_analysis(Slope_Min_Max_Forest_shp, Slope_Min_Max_Forest_Area, "POLY_AREA >=%s"
% AreaMin)

arcpy.SmoothPolygon_cartography(Slope_Min_Max_Forest_Area,
Slope_Min_Max_Forest_Area_Smooth, "PAEK", 25, "FIXED_ENDPOINT", "NO_CHECK")

```

### 9.1.2 Tabelle mit den Parameternwerten zur Kalibrierung

SlopeMin	SlopeMax	MeanHS	ProbMin	RoughnessMovWin	RoughnessMax	CurvMax	AreaMin
20	50	1	0.000000001	3	0.5	0.5	0
21	51	2	0.01	5	1	1	2000
22	52	3	0.02	7	1.5	1.5	4000
23	53	3.25	0.03	9	2	2	6000
24	54	3.5	0.04	11	2.5	2.5	8000
25	55	3.75	0.05	13	3	3	10000
26	56	4	0.06	15	3.5	3.5	12000
27	57	5	0.07		4	4	14000
28	58	6	0.08		4.5	4.5	16000
29	59	7	0.09		5	5	18000
30	60	8	0.1		5.5	5.5	20000
31	61	9	0.2		6	6	
32	62	10	0.3		6.5	6.5	
33	63		0.4		7	7	
34	64				7.5	7.5	
35	65				8	8	
36	66				8.5	8.5	
37	67				9	9	
38	68				9.5	9.5	
39	69				10	10	
40	70						

### 9.1.3 Kalibrierung Bühler et al. (2013)

```
# -*- coding: utf-8 -*-
# -----
# Value_Iteration.py
# Created on: 2017-10-24 10:42:10.00000
# Description: Calibration
# -----
import os

# Import arcpy module
import arcpy

# Load required toolboxes
arcpy.ImportToolbox("H:/Masterarbeit/Geo_Data_Prozessierung/Buehler_ArcGIS/Autorel.tbx")
arcpy.ImportToolbox("H:/Masterarbeit/Geo_Data_Prozessierung/Valid/Valid.tbx")

# Script arguments
WorkDir = arcpy.GetParameterAsText(0) + "\\\"
InputPara = arcpy.GetParameterAsText(1)
DEM = arcpy.GetParameterAsText(2)
Forest = arcpy.GetParameterAsText(3)
ResampCSize = arcpy.GetParameterAsText(4)
ForestBuffer = arcpy.GetParameterAsText(5)
CurvMax = arcpy.GetParameterAsText(6)
SlopeMin = arcpy.GetParameterAsText(7)
SlopeMax = arcpy.GetParameterAsText(8)
RoughnessMovWin = arcpy.GetParameterAsText(9)
RoughnessMax = arcpy.GetParameterAsText(10)
AreaMin = arcpy.GetParameterAsText(11)
AltiMin = arcpy.GetParameterAsText(12)
AltiMax = arcpy.GetParameterAsText(13)
SmoothPoly = arcpy.GetParameterAsText(14)
Perimeter = arcpy.GetParameterAsText(15)
PRA_Referenz = arcpy.GetParameterAsText(16)
PRA_Referenz_W = arcpy.GetParameterAsText(17)
TypeofRun = arcpy.GetParameterAsText(18)

# Local variables:
DirMethod = os.path.basename(os.path.normpath(WorkDir))

# Set Geoprocessing environments
arcpy.env.outputCoordinateSystem = arcpy.SpatialReference("CH1903+ LV95")
```

```

#####
# Data Preperation
path_base_data = WorkDir + "base_data\\"

#Local Vairables
Perimeter_Buf200m = path_base_data + "Perimeter_Buf200m.shp"
DEM_Perimeter_Buf200m = path_base_data + "DEM_Perimeter_Buf200m.tif"
DEM_Perimeter_Buf200m_Resamp = path_base_data + "DEM_Perimeter_Buf200m_Resamp.tif"
Forest_Resamp = path_base_data + "Forest_Resamp.tif"
Slope = path_base_data + "Slope.tif"
Curv = path_base_data + "Curv.tif"
Curv_plan = path_base_data + "Curv_plan.tif"
Curv_plan_expanded = path_base_data + "Curv_plan_expanded.tif"
DEM_Perimeter_Statistics_dbf = "in_memory/DEM_Peri_Stat"
DEM_Perimeter_Statistics_xls = path_base_data + "DEM_Perimeter_Statistics.xls"
Slope_Perimeter_Statistics_dbf = "in_memory/Slope_Peri_Stat"
Slope_Perimeter_Statistics_xls = path_base_data + "Slope_Perimeter_Statistics.xls"

if os.path.isdir(path_base_data) == False:
    arcpy.CreateFolder_management(WorkDir, "base_data")

    arcpy.Buffer_analysis(Perimeter, Perimeter_Buf200m, '200 Meters', 'FULL', 'ROUND', 'NONE', '#',
'PLANAR')
    arcpy.gp.ExtractByMask_sa(DEM, Perimeter_Buf200m, DEM_Perimeter_Buf200m)
    arcpy.Resample_management(DEM_Perimeter_Buf200m, DEM_Perimeter_Buf200m_Resamp,
ResampCSize, 'BILINEAR')
    arcpy.Resample_management(Forest, Forest_Resamp, ResampCSize, 'BILINEAR')

    # create slope
    arcpy.AddMessage("create slope...")
    arcpy.gp.Slope_sa(DEM_Perimeter_Buf200m_Resamp, Slope, 'DEGREE', '1')

    # create curvature
    arcpy.AddMessage("create curvature...")
    arcpy.gp.Curvature_sa(DEM_Perimeter_Buf200m_Resamp, Curv, '1', '#', Curv_plan)

    # create curvature expanded
    arcpy.AddMessage("create curvature expanded...")
    arcpy.gp.FocalStatistics_sa(Curv_plan, Curv_plan_expanded, 'Rectangle 3 3 CELL', 'MEAN',
'DATA')

    # Overall Statistics

```

```

    arcpy.gp.ZonalStatisticsAsTable_sa(Perimeter, 'Name', DEM_Perimeter_Buf200m_Resamp,
DEM_Perimeter_Statistics_dbf, 'DATA', 'ALL')
    arcpy.TableToExcel_conversion(DEM_Perimeter_Statistics_dbf, DEM_Perimeter_Statistics_xls,
'ALIAS', 'CODE')
    arcpy.gp.ZonalStatisticsAsTable_sa(Perimeter, 'Name', Slope, Slope_Perimeter_Statistics_dbf,
'DATA', 'ALL')
    arcpy.TableToExcel_conversion(Slope_Perimeter_Statistics_dbf, Slope_Perimeter_Statistics_xls,
'ALIAS', 'CODE')

##  arcpy.SiteStatistics(path_base_data, Perimeter, DEM_Perimeter_Buf200m_Resamp, Slope)
#####
# Function to Export Input Parameters in File
def export_input_para(MethodRun, DEM, Forest, ResampCSize, ForestBuffer, CurvMax, SlopeMin,
SlopeMax, RoughnessMovWin, RoughnessMax, AreaMin, AltiMin, AltiMax, SmoothPoly, Perimeter,
PRA_Referenz, PRA_Referenz_W):
    import csv
    column1 = ["Name", "DHM", "Wald", "Neuberechnung Zellgroesse [m]", "Puffer Wald [m]", "Max Kur-
vatur [ ]", "Min Hangneigung [Deg]", "Max Hangneigung [Deg]", "Groesse Zellfenster Rauigkeit [
]", "Max Rauigkeit [ ]", "Min Anrissflaeche [m2]", "Min Hoehe [m.ue.M.]", "Max Hoehe
[m.ue.M.]", "Glaettung Polygone", "Perimeter", "PRA_Referenz", "PRA_Referenz_W"]
    column2 = [MethodRun, DEM, Forest, ResampCSize, ForestBuffer, CurvMax, SlopeMin, SlopeMax,
RoughnessMovWin, RoughnessMax, AreaMin, AltiMin, AltiMax, SmoothPoly, Perimeter,
PRA_Referenz, PRA_Referenz_W]
    array = [column1, column2]
    FileName = WorkDir + MethodRun + "\\\" + MethodRun
    with open("%s_Input_Parameters.csv" % FileName, "wb") as f:
        writer = csv.writer(f, delimiter=";")
        writer.writerows(array)
        f.close()

#####
# Function to iterate over Input Parameter
def iterate_over_one_InputPara(rowName, InputPara, WorkDir, DEM_Perimeter_Buf200m_Resamp,
Forest_Resamp, ResampCSize, ForestBuffer, CurvMax, SlopeMin, SlopeMax, RoughnessMovWin,
RoughnessMax, AreaMin, AltiMin, AltiMax, SmoothPoly, Perimeter, PRA_Referenz, PRA_Referenz_W,
Slope, Curv_plan_expanded):
    arcpy.AddMessage("Iteration over " + rowName + ".....")
    cursor = arcpy.SearchCursor(InputPara)
    for row in cursor:
        Value = row.getValue(rowName)

        if Value != None:
            if rowName == "RoughnessMax":

```



```

Value = Value/100
Value = str(Value)
Value_mod = Value.replace(".", "p")
locals_dict = dict(vars())
locals_dict[rowName] = Value
arcpy.AddMessage(str(rowName) + " = " + locals_dict[rowName])
MethodRun = DirMethod + "_" + rowName + Value_mod

if (os.path.isdir(WorkDir + MethodRun) == 0):
    arcpy.CreateFolder_management(WorkDir, MethodRun)
    export_input_para(MethodRun, DEM, Forest, ResampCSize, ForestBuffer, locals_dict["CurvMax"], locals_dict["SlopeMin"], locals_dict["SlopeMax"], locals_dict["RoughnessMovWin"], locals_dict["RoughnessMax"], locals_dict["AreaMin"], AltiMin, AltiMax, SmoothPoly, Perimeter, PRA_Referenz, PRA_Referenz_W)
    arcpy.Autorel2(WorkDir + MethodRun, MethodRun, DEM_Perimeter_Buf200m_Resamp, Forest_Resamp, ResampCSize, ForestBuffer, locals_dict["CurvMax"], locals_dict["SlopeMin"], locals_dict["SlopeMax"], locals_dict["RoughnessMovWin"], locals_dict["RoughnessMax"], locals_dict["AreaMin"], AltiMin, AltiMax, SmoothPoly, Slope, Curv_plan_expanded)
    PRA_Methode = WorkDir + MethodRun + "\\temp_model\\relzones4_shp.shp"
    PRA_Methode_NoSimplify = WorkDir + MethodRun + "\\temp_model\\relzones3_shp.shp"
    arcpy.Valid(WorkDir + MethodRun, MethodRun, DEM_Perimeter_Buf200m_Resamp, Slope, PRA_Methode, PRA_Methode_NoSimplify, Perimeter, PRA_Referenz, PRA_Referenz_W, locals_dict["AreaMin"])

elif rowName == "CurvMax":
    Value = str(Value)
    Value_mod = Value.replace(".", "p")
    locals_dict = dict(vars())
    locals_dict[rowName] = Value
    arcpy.AddMessage(str(rowName) + " = " + locals_dict[rowName])
    MethodRun = DirMethod + "_" + rowName + Value_mod

if (os.path.isdir(WorkDir + MethodRun) == 0):
    arcpy.CreateFolder_management(WorkDir, MethodRun)
    export_input_para(MethodRun, DEM, Forest, ResampCSize, ForestBuffer, locals_dict["CurvMax"], locals_dict["SlopeMin"], locals_dict["SlopeMax"], locals_dict["RoughnessMovWin"], locals_dict["RoughnessMax"], locals_dict["AreaMin"], AltiMin, AltiMax, SmoothPoly, Perimeter, PRA_Referenz, PRA_Referenz_W)
    arcpy.Autorel2(WorkDir + MethodRun, MethodRun, DEM_Perimeter_Buf200m_Resamp, Forest_Resamp, ResampCSize, ForestBuffer, locals_dict["CurvMax"], locals_dict["SlopeMin"], locals_dict["SlopeMax"], locals_dict["RoughnessMovWin"], locals_dict["RoughnessMax"], locals_dict["AreaMin"], AltiMin, AltiMax, SmoothPoly, Slope, Curv_plan_expanded)

```

```

PRA_Methode = WorkDir + MethodRun + "\\temp_model\\relzones4_shp.shp"
PRA_Methode_NoSimplify = WorkDir + MethodRun +
"\\temp_model\\relzones3_shp.shp"
arcpy.Valid(WorkDir + MethodRun, MethodRun, DEM_Perimeter_Buf200m_Resamp,
Slope, PRA_Methode, PRA_Methode_NoSimplify, Perimeter, PRA_Referenz, PRA_Referenz_W, lo-
cals_dict["AreaMin"])

else:
    Value = str(int(Value))
    Value_mod = Value.replace(".", "p")
    locals_dict = dict(vars())
    locals_dict[rowName] = Value
    arcpy.AddMessage(str(rowName) + " = " + locals_dict[rowName])
    MethodRun = DirMethod + "_" + rowName + Value_mod

if (os.path.isdir(WorkDir + MethodRun) == 0):
    arcpy.CreateFolder_management(WorkDir, MethodRun)
    export_input_para(MethodRun, DEM, Forest, ResampCSize, ForestBuffer, lo-
cals_dict["CurvMax"], locals_dict["SlopeMin"], locals_dict["SlopeMax"], lo-
cals_dict["RoughnessMovWin"], locals_dict["RoughnessMax"], locals_dict["AreaMin"], AltiMin, Alti-
Max, SmoothPoly, Perimeter, PRA_Referenz, PRA_Referenz_W)
    arcpy.Autorel2(WorkDir + MethodRun, MethodRun, DEM_Perimeter_Buf200m_Resamp,
Forest_Resamp, ResampCSize, ForestBuffer, locals_dict["CurvMax"], locals_dict["SlopeMin"], lo-
cals_dict["SlopeMax"], locals_dict["RoughnessMovWin"], locals_dict["RoughnessMax"], lo-
cals_dict["AreaMin"], AltiMin, AltiMax, SmoothPoly, Slope, Curv_plan_expanded)
    PRA_Methode = WorkDir + MethodRun + "\\temp_model\\relzones4_shp.shp"
    PRA_Methode_NoSimplify = WorkDir + MethodRun +
"\\temp_model\\relzones3_shp.shp"
    arcpy.Valid(WorkDir + MethodRun, MethodRun, DEM_Perimeter_Buf200m_Resamp,
Slope, PRA_Methode, PRA_Methode_NoSimplify, Perimeter, PRA_Referenz, PRA_Referenz_W, lo-
cals_dict["AreaMin"])

#####
if TypeofRun == "m":

    # Execute Functions, which iterate over Input Parameter
    iterate_over_one_InputPara("RoughnessMax", InputPara, WorkDir,
DEM_Perimeter_Buf200m_Resamp, Forest_Resamp, ResampCSize, ForestBuffer, CurvMax, SlopeMin,
SlopeMax, RoughnessMovWin, RoughnessMax, AreaMin, AltiMin, AltiMax, SmoothPoly, Perimeter,
PRA_Referenz, PRA_Referenz_W, Slope, Curv_plan_expanded)
    iterate_over_one_InputPara("SlopeMin", InputPara, WorkDir, DEM_Perimeter_Buf200m_Resamp,
Forest_Resamp, ResampCSize, ForestBuffer, CurvMax, SlopeMin, SlopeMax, RoughnessMovWin,

```

```

RoughnessMax, AreaMin, AltiMin, AltiMax, SmoothPoly, Perimeter, PRA_Referenz, PRA_Referenz_W,
Slope, Curv_plan_expanded)
    iterate_over_one_InputPara("CurvMax", InputPara, WorkDir, DEM_Perimeter_Buf200m_Resamp,
Forest_Resamp, ResampCSize, ForestBuffer, CurvMax, SlopeMin, SlopeMax, RoughnessMovWin,
RoughnessMax, AreaMin, AltiMin, AltiMax, SmoothPoly, Perimeter, PRA_Referenz, PRA_Referenz_W,
Slope, Curv_plan_expanded)
    iterate_over_one_InputPara("SlopeMax", InputPara, WorkDir, DEM_Perimeter_Buf200m_Resamp,
Forest_Resamp, ResampCSize, ForestBuffer, CurvMax, SlopeMin, SlopeMax, RoughnessMovWin,
RoughnessMax, AreaMin, AltiMin, AltiMax, SmoothPoly, Perimeter, PRA_Referenz, PRA_Referenz_W,
Slope, Curv_plan_expanded)
    iterate_over_one_InputPara("RoughnessMovWin", InputPara, WorkDir,
DEM_Perimeter_Buf200m_Resamp, Forest_Resamp, ResampCSize, ForestBuffer, CurvMax, SlopeMin,
SlopeMax, RoughnessMovWin, RoughnessMax, AreaMin, AltiMin, AltiMax, SmoothPoly, Perimeter,
PRA_Referenz, PRA_Referenz_W, Slope, Curv_plan_expanded)
    iterate_over_one_InputPara("AreaMin", InputPara, WorkDir, DEM_Perimeter_Buf200m_Resamp,
Forest_Resamp, ResampCSize, ForestBuffer, CurvMax, SlopeMin, SlopeMax, RoughnessMovWin,
RoughnessMax, AreaMin, AltiMin, AltiMax, SmoothPoly, Perimeter, PRA_Referenz, PRA_Referenz_W,
Slope, Curv_plan_expanded)

#####
if TypeofRun == "s":

    i = 1
    MethodRun = (DirMethod + "_bestInputPara_" + str(i))

    while (os.path.isdir(WorkDir + MethodRun) == 1):
        i = i+1
        MethodRun = DirMethod + "_bestInputPara_" + str(i)

    # Execute Run with best Input Parameters
    arcpy.AddMessage(MethodRun)
    arcpy.CreateFolder_management(WorkDir, MethodRun)
    export_input_para(MethodRun, DEM, Forest, ResampCSize, ForestBuffer, CurvMax, SlopeMin,
SlopeMax, RoughnessMovWin, RoughnessMax, AreaMin, AltiMin, AltiMax, SmoothPoly, Perimeter,
PRA_Referenz, PRA_Referenz_W)
    arcpy.Autorel2(WorkDir + MethodRun, MethodRun, DEM_Perimeter_Buf200m_Resamp, For-
est_Resamp, ResampCSize, ForestBuffer, CurvMax, SlopeMin, SlopeMax, RoughnessMovWin, Rough-
nessMax, AreaMin, AltiMin, AltiMax, SmoothPoly, Slope, Curv_plan_expanded)
    PRA_Methode = WorkDir + MethodRun + "\\temp_model\\relzones4_shp.shp"
    PRA_Methode_NoSimplify = WorkDir + MethodRun + "\\temp_model\\relzones3_shp.shp"
    arcpy.Valid(WorkDir + MethodRun, MethodRun, DEM_Perimeter_Buf200m_Resamp, Slope,
PRA_Methode, PRA_Methode_NoSimplify, Perimeter, PRA_Referenz, PRA_Referenz_W, AreaMin)

```

### 9.1.4 Kalibrierung Veitinger et al. (2016)

```
# -*- coding: utf-8 -*-
# -----
# Value_Iteration.py
# Created on: 2017-10-24 10:42:10.00000
# Description: Calibration
# -----
import os

# Import arcpy module
import arcpy

# Load required toolboxes
arcpy.ImportToolbox("H:/Masterarbeit/Geo_Data_Prozessierung/Veitinger/Avalanches.tbx")
arcpy.ImportToolbox("H:/Masterarbeit/Geo_Data_Prozessierung/Valid/Valid.tbx")

# Script arguments
WorkDir = arcpy.GetParameterAsText(0) + "\\\"
InputPara = arcpy.GetParameterAsText(1)
DEM = arcpy.GetParameterAsText(2)
MeanHS = arcpy.GetParameterAsText(3)
SmoothDeg = arcpy.GetParameterAsText(4)
MainWindDir = arcpy.GetParameterAsText(5)
MainWindDirTol = arcpy.GetParameterAsText(6)
Forest = arcpy.GetParameterAsText(7)
ProbMin = arcpy.GetParameterAsText(8)
AreaMin = arcpy.GetParameterAsText(9)
Perimeter = arcpy.GetParameterAsText(10)
PRA_Referenz = arcpy.GetParameterAsText(11)
PRA_Referenz_W = arcpy.GetParameterAsText(12)
ResampCSize = arcpy.GetParameterAsText(13)
TypeofRun = arcpy.GetParameterAsText(14)

# Set Geoprocessing environments
SpatialRef = arcpy.SpatialReference("CH1903+ LV95")
arcpy.env.outputCoordinateSystem = SpatialRef

# Local variables:
DirMethod = os.path.basename(os.path.normpath(WorkDir))

#####
# Data Preperation DEM and Forest for Perimeter
path_base_data = WorkDir + "base_data\\"
```

```

#Local Vairables
Perimeter_Buf200m_tif = path_base_data + "Perimeter_Buf200m.shp"
DEM_Perimeter_Buf200m_tif = path_base_data + "DEM_Perimeter_Buf200m.tif"
DEM_Perimeter_Buf200m_Resamp_tif = path_base_data + "DEM_Perimeter_Buf200m_Resamp.tif"
DEM_Perimeter_Buf200m_Resamp_asc = path_base_data + "DEM_Perimeter_Buf200m_Resamp.asc"
Forest_Perimeter_Buf200m_tif = path_base_data + "Forest_Perimeter_Buf200m.tif"
Forest_Perimeter_Buf200m_Resamp_tif = path_base_data + "Forest_Perimeter_Buf200m_Resamp.tif"
Forest_Perimeter_Buf200m_Resamp_asc = path_base_data + "For-
est_Perimeter_Buf200m_Resamp.asc"
DEM_Perimeter_Statistics_dbf = path_base_data + "DEM_Perimeter_Statistics.dbf"
DEM_Perimeter_Statistics_xls = path_base_data + "DEM_Perimeter_Statistics.xls"

if os.path.isdir(path_base_data) == False:

    arcpy.CreateFolder_management(WorkDir, "base_data")

    arcpy.Buffer_analysis(Perimeter, Perimeter_Buf200m_tif, '200 Meters', 'FULL', 'ROUND', 'NONE',
'#', 'PLANAR')
    arcpy.gp.ExtractByMask_sa(DEM, Perimeter_Buf200m_tif, DEM_Perimeter_Buf200m_tif)
    arcpy.Resample_management(DEM_Perimeter_Buf200m_tif,
DEM_Perimeter_Buf200m_Resamp_tif, ResampCSize, 'BILINEAR')
    arcpy.RasterToASCII_conversion(DEM_Perimeter_Buf200m_Resamp_tif,
DEM_Perimeter_Buf200m_Resamp_asc)
    arcpy.gp.ExtractByMask_sa(Forest, Perimeter_Buf200m_tif, Forest_Perimeter_Buf200m_tif)
    arcpy.Resample_management(Forest_Perimeter_Buf200m_tif, For-
est_Perimeter_Buf200m_Resamp_tif, ResampCSize, 'BILINEAR')
    arcpy.RasterToASCII_conversion(Forest_Perimeter_Buf200m_Resamp_tif, For-
est_Perimeter_Buf200m_Resamp_asc)
    arcpy.gp.ZonalStatisticsAsTable_sa(Perimeter, 'Id', DEM_Perimeter_Buf200m_Resamp_tif,
DEM_Perimeter_Statistics_dbf, 'DATA', 'ALL')
    arcpy.TableToExcel_conversion(DEM_Perimeter_Statistics_dbf, DEM_Perimeter_Statistics_xls,
'ALIAS', 'CODE')

#####
# Function to Export Input Parameters in File
def export_input_para(MethodRun, DEM, Forest, ResampCSize, MeanHS, SmoothDeg, MainWindDir,
MainWindDirTol, ProbMin, AreaMin, Perimeter, PRA_Referenz, PRA_Referenz_W):
    import csv
    column1 = ["Name", "DHM", "Wald", "Neuberechnung Zellgroesse [m]", "Mittl. Schneehoehe [m]",
"Glaettungsgrad", "Hauptwindrichtung [Deg]", "Windtoleranz [Deg]", "Min Anrissanfaelligkeit [ ]",
"Min Anrissflaeche [m2]", "Perimeter", "PRA_Referenz", "PRA_Referenz_W"]

```

```

column2 = [MethodRun, DEM, Forest, ResampCSize, MeanHS, SmoothDeg, MainWindDir, Main-
WindDirTol, ProbMin, AreaMin, Perimeter, PRA_Referenz, PRA_Referenz_W]
array = [column1,column2]
FileName = WorkDir + MethodRun + "\\\" + MethodRun
with open("%s_Input_Parameters.csv" % FileName,"wb") as f:
    writer = csv.writer(f,delimiter=";")
    writer.writerows(array)
f.close()

```

```
#####
```

```
# Function to prepare Data for Validation
```

```
def prepare_for_valid(WorkDir, MethodRun, outPRA, ProbMin, AreaMin):
```

```
    # Process: Create Folder
```

```
    arcpy.CreateFolder_management(WorkDir + MethodRun, "temp_post_process")
```

```
    # Set Geoprocessing environments
```

```
    arcpy.env.extent = Perimeter
```

```
    # Local variables:
```

```
    WorkDir_temp_post_process = WorkDir + MethodRun + "\\temp_post_process\\"
```

```
    PRA_Methode_tif = WorkDir_temp_post_process + "\\\" + MethodRun + "_PRA.tif"
```

```
    PRA_Methode_PorbMin_tif = WorkDir_temp_post_process + "PRA_Methode_PorbMin.tif"
```

```
    PRA_Methode_PorbMin_shp = WorkDir_temp_post_process + "PRA_Methode_PorbMin.shp"
```

```
    PRA_Methode_PorbMin_AreaMin_shp = WorkDir_temp_post_process +
"PRA_Methode_ProbMin_AreaMin.shp"
```

```
    PRA_Methode_PorbMin_AreaMin_Smooth_shp = WorkDir_temp_post_process +
```

```
"PRA_Methode_ProbMin_AreaMin_Smooth.shp"
```

```
    PRA_Methode = PRA_Methode_PorbMin_AreaMin_Smooth_shp
```

```
    DEM_tif = WorkDir_temp_post_process + "DEM.tif"
```

```
    outslope = WorkDir + "\\temp_model\\" + "slope1.asc"
```

```
    outslope_tif = WorkDir_temp_post_process + "slope1.tif"
```

```
    # Prepare Model Output for Valididation
```

```
    arcpy.ASCIIToRaster_conversion(outPRA, PRA_Methode_tif, 'FLOAT')
```

```
    arcpy.DefineProjection_management(PRA_Methode_tif, SpatialRef)
```

```
    arcpy.gp.RasterCalculator_sa('Con("%s" >= %s,1)' % (PRA_Methode_tif, ProbMin),
PRA_Methode_PorbMin_tif)
```

```
    arcpy.RasterToPolygon_conversion(PRA_Methode_PorbMin_tif, PRA_Methode_PorbMin_shp,
'NO_SIMPLIFY', 'Value')
```

```
    arcpy.AddGeometryAttributes_management(PRA_Methode_PorbMin_shp, 'AREA', 'METERS',
'SQUARE_METERS', '#')
```

```

    arcpy.Select_analysis(PRA_Methode_PorbMin_shp, PRA_Methode_PorbMin_AreaMin_shp,
    ""POLY_AREA" > %s' % AreaMin)
    arcpy.SmoothPolygon_cartography(PRA_Methode_PorbMin_AreaMin_shp, PRA_Methode, "PAEK",
    25, "FIXED_ENDPOINT", "NO_CHECK")
    arcpy.ASCIIToRaster_conversion(WorkDir + MethodRun + "\\temp_model\\slope1.asc", out-
    slope_tif, 'FLOAT')
    arcpy.DefineProjection_management(outslope_tif, SpatialRef)

```

```
#####
```

```

# Function to iterate over Input Parameter
def iterate_over_one_InputPara(rowName, InputPara, WorkDir,
DEM_Perimeter_Buf200m_Resamp_asc, MeanHS, SmoothDeg, MainWindDir, MainWindDirTol, For-
est_Perimeter_Buf200m_Resamp_asc, ProbMin, AreaMin, Perimeter, PRA_Referenz,
PRA_Referenz_W):

```

```

    arcpy.AddMessage("Iteration over " + rowName + ".....")

```

```

    cursor = arcpy.SearchCursor(InputPara)

```

```

    for row in cursor:

```

```

        Value = row.getValue(rowName)

```

```

        if Value != None:

```

```

            if rowName == "AreaMin":

```

```

                Value = str(int(Value))

```

```

                Value_mod = Value.replace(".", "p")

```

```

                locals_dict = dict(vars())

```

```

                locals_dict[rowName] = Value

```

```

            arcpy.AddMessage(str(rowName) + " = " + locals_dict[rowName])

```

```

            MethodRun = DirMethod + "_" + rowName + "_" + Value_mod

```

```

            if (os.path.isdir(WorkDir + MethodRun) == 0):

```

```

                arcpy.CreateFolder_management(WorkDir, MethodRun)

```

```

                export_input_para(MethodRun, DEM, Forest, ResampCSize, locals_dict["MeanHS"], lo-
                cals_dict["SmoothDeg"], MainWindDir, MainWindDirTol, locals_dict["ProbMin"], lo-
                cals_dict["AreaMin"], Perimeter, PRA_Referenz, PRA_Referenz_W)

```

```

                outPRA = WorkDir + MethodRun + "\\temp_model\\" + MethodRun + "_PRA.asc"

```

```

                arcpy.PRA2(DEM_Perimeter_Buf200m_Resamp_asc, outPRA, locals_dict["MeanHS"], lo-
                cals_dict["SmoothDeg"], MainWindDir, MainWindDirTol, WorkDir + MethodRun, For-
                est_Perimeter_Buf200m_Resamp_asc)

```

```

                arcpy.env.extent = Perimeter

```

```

        prepare_for_valid(WorkDir, MethodRun, outPRA, locals_dict["ProbMin"], locals_dict["AreaMin"])

        outslope_tif = WorkDir + MethodRun + "\\temp_post_process\\slope1.tif"
        PRA_Methode = WorkDir + MethodRun +
        "\\temp_post_process\\PRA_Methode_ProbMin_AreaMin_Smooth.shp"
        PRA_Methode_NoSimplify = WorkDir + MethodRun +
        "\\temp_post_process\\PRA_Methode_ProbMin_AreaMin.shp"
        arcpy.Valid(WorkDir + MethodRun, MethodRun, DEM_Perimeter_Buf200m_Resamp_tif,
        outslope_tif, PRA_Methode, PRA_Methode_NoSimplify, Perimeter, PRA_Referenz, PRA_Referenz_W,
        locals_dict["AreaMin"])

    elif rowName == "SmoothDeg":
        Value = str(Value)
        Value_mod = Value.replace(".", "p")
        locals_dict = dict(vars())
        locals_dict[rowName] = Value

        arcpy.AddMessage(str(rowName) + " = " + locals_dict[rowName])
        MethodRun = DirMethod + "_" + rowName + "_" + Value_mod
        if (os.path.isdir(WorkDir + MethodRun) == 0):
            arcpy.CreateFolder_management(WorkDir, MethodRun)

        export_input_para(MethodRun, DEM, Forest, ResampCSize, locals_dict["MeanHS"], locals_dict["SmoothDeg"], MainWindDir, MainWindDirTol, locals_dict["ProbMin"], locals_dict["AreaMin"], Perimeter, PRA_Referenz, PRA_Referenz_W)

        outPRA = WorkDir + MethodRun + "\\temp_model\\" + MethodRun + "_PRA.asc"
        arcpy.PRA2(DEM_Perimeter_Buf200m_Resamp_asc, outPRA, locals_dict["MeanHS"], locals_dict["SmoothDeg"], MainWindDir, MainWindDirTol, WorkDir + MethodRun, Forest_Perimeter_Buf200m_Resamp_asc)

        arcpy.env.extent = Perimeter

        prepare_for_valid(WorkDir, MethodRun, outPRA, locals_dict["ProbMin"], locals_dict["AreaMin"])

        outslope_tif = WorkDir + MethodRun + "\\temp_post_process\\slope1.tif"
        PRA_Methode = WorkDir + MethodRun +
        "\\temp_post_process\\PRA_Methode_ProbMin_AreaMin_Smooth.shp"
        PRA_Methode_NoSimplify = WorkDir + MethodRun +
        "\\temp_post_process\\PRA_Methode_ProbMin_AreaMin.shp"

```



```
        arcpy.Valid(WorkDir + MethodRun, MethodRun, DEM_Perimeter_Buf200m_Resamp_tif,
outslope_tif, PRA_Methode, PRA_Methode_NoSimplify, Perimeter, PRA_Referenz, PRA_Referenz_W,
locals_dict["AreaMin"])
```

```
    else:
```

```
        Value = str(Value)
```

```
        if 'e' in Value:
```

```
            superscript = Value[3:]
```

```
            superscript = int(superscript)
```

```
            print(superscript)
```

```
            Value = float(Value)
```

```
            Value = 'prec f'.format(Value, prec=superscript)
```

```
            Value = str(Value)
```

```
        Value_mod = Value.replace(".", "p")
```

```
        locals_dict = dict(vars())
```

```
        locals_dict[rowName] = Value
```

```
        arcpy.AddMessage(str(rowName) + " = " + locals_dict[rowName])
```

```
        MethodRun = DirMethod + "_" + rowName + "_" + Value_mod
```

```
        if (os.path.isdir(WorkDir + MethodRun) == 0):
```

```
            arcpy.CreateFolder_management(WorkDir, MethodRun)
```

```
            export_input_para(MethodRun, DEM, Forest, ResampCSize, locals_dict["MeanHS"], lo-
cals_dict["SmoothDeg"], MainWindDir, MainWindDirTol, locals_dict["ProbMin"], lo-
cals_dict["AreaMin"], Perimeter, PRA_Referenz, PRA_Referenz_W)
```

```
            outPRA = WorkDir + MethodRun + "\\temp_model\\" + MethodRun + "_PRA.asc"
```

```
            arcpy.PRA2(DEM_Perimeter_Buf200m_Resamp_asc, outPRA, locals_dict["MeanHS"], lo-
cals_dict["SmoothDeg"], MainWindDir, MainWindDirTol, WorkDir + MethodRun, For-
est_Perimeter_Buf200m_Resamp_asc)
```

```
            arcpy.env.extent = Perimeter
```

```
            prepare_for_valid(WorkDir, MethodRun, outPRA, locals_dict["ProbMin"], lo-
cals_dict["AreaMin"])
```

```
            outslope_tif = WorkDir + MethodRun + "\\temp_post_process\\slope1.tif"
```

```
            PRA_Methode = WorkDir + MethodRun +
```

```
            "\\temp_post_process\\PRA_Methode_ProbMin_AreaMin_Smooth.shp"
```

```
            PRA_Methode_NoSimplify = WorkDir + MethodRun +
```

```
            "\\temp_post_process\\PRA_Methode_ProbMin_AreaMin.shp"
```

```
arcpy.Valid(WorkDir + MethodRun, MethodRun, DEM_Perimeter_Buf200m_Resamp_tif,
outslope_tif, PRA_Methode, PRA_Methode_NoSimplify, Perimeter, PRA_Referenz, PRA_Referenz_W,
locals_dict["AreaMin"])
```

```
#####
if TypeofRun == "m":
```

```
    # Execute Functions, which iterate over Input Parameter
    iterate_over_one_InputPara("MeanHS", InputPara, WorkDir,
DEM_Perimeter_Buf200m_Resamp_asc, MeanHS, SmoothDeg, MainWindDir, MainWindDirTol, For-
est_Perimeter_Buf200m_Resamp_asc, ProbMin, AreaMin, Perimeter, PRA_Referenz,
PRA_Referenz_W)
    iterate_over_one_InputPara("ProbMin", InputPara, WorkDir,
DEM_Perimeter_Buf200m_Resamp_asc, MeanHS, SmoothDeg, MainWindDir, MainWindDirTol, For-
est_Perimeter_Buf200m_Resamp_asc, ProbMin, AreaMin, Perimeter, PRA_Referenz,
PRA_Referenz_W)
    iterate_over_one_InputPara("SmoothDeg", InputPara, WorkDir,
DEM_Perimeter_Buf200m_Resamp_asc, MeanHS, SmoothDeg, MainWindDir, MainWindDirTol, For-
est_Perimeter_Buf200m_Resamp_asc, ProbMin, AreaMin, Perimeter, PRA_Referenz,
PRA_Referenz_W)
    iterate_over_one_InputPara("AreaMin", InputPara, WorkDir,
DEM_Perimeter_Buf200m_Resamp_asc, MeanHS, SmoothDeg, MainWindDir, MainWindDirTol, For-
est_Perimeter_Buf200m_Resamp_asc, ProbMin, AreaMin, Perimeter, PRA_Referenz,
PRA_Referenz_W)
```

```
#####
if TypeofRun == "s":
```

```
    i = 1
    MethodRun = (DirMethod + "_bestInputPara_" + str(i))

    while (os.path.isdir(WorkDir + MethodRun) == 1):
        i = i+1
        MethodRun = DirMethod + "_bestInputPara_" + str(i)
```

```
    # Execute Run with best Input Parameters
    arcpy.AddMessage(MethodRun)
    arcpy.CreateFolder_management(WorkDir, MethodRun)
```

```
    export_input_para(MethodRun, DEM, Forest, ResampCSize, MeanHS, SmoothDeg, MainWindDir,
MainWindDirTol, ProbMin, AreaMin, Perimeter, PRA_Referenz, PRA_Referenz_W)
```

```
    outPRA = WorkDir + MethodRun + "\\temp_model\\" + MethodRun + "_PRA.asc"
```

```
arcpy.PRA2(DEM_Perimeter_Buf200m_Resamp_asc, outPRA, MeanHS, SmoothDeg, MainWindDir,  
MainWindDirTol, WorkDir + MethodRun, Forest_Perimeter_Buf200m_Resamp_asc)
```

```
arcpy.env.extent = Perimeter
```

```
prepare_for_valid(WorkDir, MethodRun, outPRA, ProbMin, AreaMin)
```

```
outslope_tif = WorkDir + MethodRun + "\\temp_post_process\\slope1.tif"
```

```
PRA_Methode = WorkDir + MethodRun +
```

```
"\\temp_post_process\\PRA_Methode_ProbMin_AreaMin_Smooth.shp"
```

```
PRA_Methode_NoSimplify = WorkDir + MethodRun +
```

```
"\\temp_post_process\\PRA_Methode_ProbMin_AreaMin.shp"
```

```
arcpy.Valid(WorkDir + MethodRun, MethodRun, DEM_Perimeter_Buf200m_Resamp_tif, out-  
slope_tif, PRA_Methode, PRA_Methode_NoSimplify, Perimeter, PRA_Referenz, PRA_Referenz_W,  
AreaMin)
```

### 9.1.5 Kalibrierung Bühler et al. (2018)

```
# -*- coding: utf-8 -*-
# -----
# Value_Iteration.py
# Created on: 2017-10-24 10:42:10.00000
# Description: Calibration
# -----
import os

# Import arcpy module
import arcpy
from arcpy import env
from arcpy.sa import *
import traceback

# Check out any necessary licenses
arcpy.CheckOutExtension("spatial")
# Check out ArcGIS 3D Analyst extension license
arcpy.CheckOutExtension("3d")

# Load required toolboxes
arcpy.ImportToolbox("H:/Masterarbeit/Geo_Data_Prozessierung/Buehler_OBIA/ArcGIS/LSHM/LSH
M.tbx")
arcpy.ImportToolbox("H:/Masterarbeit/Geo_Data_Prozessierung/Valid/Valid.tbx")

# Script arguments
WorkDir = arcpy.GetParameterAsText(0) + "\\\"
InputPara = arcpy.GetParameterAsText(1)
DEM = arcpy.GetParameterAsText(2)
Forest = arcpy.GetParameterAsText(3)
RoughnessMovWin = arcpy.GetParameterAsText(4)
SlopeMin = arcpy.GetParameterAsText(5)
SlopeMax = arcpy.GetParameterAsText(6)
RoughnessMax = (arcpy.GetParameterAsText(7))
AreaMin = arcpy.GetParameterAsText(8)
Perimeter = arcpy.GetParameterAsText(9)
PRA_Referenz = arcpy.GetParameterAsText(10)
PRA_Referenz_W = arcpy.GetParameterAsText(11)
ResampCSize = arcpy.GetParameterAsText(12)
TypeofRun = arcpy.GetParameterAsText(13)
CurvMax = arcpy.GetParameterAsText(14)

# Local variables:
```

```

DirMethod = os.path.basename(os.path.normpath(WorkDir))

# Set Geoprocessing environments
arcpy.env.outputCoordinateSystem = arcpy.SpatialReference("CH1903+ LV95")

#####
# Data Preparation
path_base_data = WorkDir + "base_data\\"

#Local Vairables
Perimeter_Buf200m = path_base_data + "Perimeter_Buf200m.shp"
MinEnvelop_Perimeter_Buf200m = path_base_data + "MinEnvelop_Perimeter_Buf200m.shp"
DEM_MinEnvelop_Perimeter_Buf200m = path_base_data +
"DEM_MinEnvelop_Perimeter_Buf200m.tif"
DEM_MinEnvelop_Perimeter_Buf200m_Resamp_temp = path_base_data +
"DEM_MinEnvelop_Perimeter_Buf200m_Resamp_temp.tif"
DEM_MinEnvelop_Perimeter_Buf200m_Resamp = path_base_data +
"DEM_MinEnvelop_Perimeter_Buf200m_Resamp.tif"
Forest_Resamp = path_base_data + "Forest_Resamp.tif"
Slope = path_base_data + "Slope.tif"
Aspect = path_base_data + "Aspect.tif"
Aspect_sectors_N0 = path_base_data + "Aspect_sectors_N0.tif"
Aspect_sectors_Nmax = path_base_data + "Aspect_sectors_Nmax.tif"
Curv = path_base_data + "Curv.tif"
Curv_profile = path_base_data + "Curv_profile.tif"
Curv_profile_expanded = path_base_data + "Curv_profile_expanded.tif"
Curv_plan = path_base_data + "Curv_plan.tif"
Curv_plan_expanded = path_base_data + "Curv_plan_expanded.tif"
Hillsh = path_base_data + "Hillshade.tif"
Hillsh_Perimeter = path_base_data + "Hillshade_Perimeter.tif"

if os.path.isdir(path_base_data) == False:
    arcpy.CreateFolder_management(WorkDir, "base_data")

    arcpy.Buffer_analysis(Perimeter, Perimeter_Buf200m, '200 Meters', 'FULL', 'ROUND', 'NONE', '#',
'PLANAR')
    arcpy.FeatureEnvelopeToPolygon_management(Perimeter_Buf200m, MinEnvel-
op_Perimeter_Buf200m, 'SINGLEPART')
    arcpy.Buffer_analysis(MinEnvelop_Perimeter_Buf200m, "in_memory/Perimeter_Buf200m", '-20
Meters', 'FULL', 'ROUND', 'NONE', '#', 'PLANAR')
    arcpy.gp.ExtractByMask_sa(DEM, MinEnvelop_Perimeter_Buf200m,
DEM_MinEnvelop_Perimeter_Buf200m)

```

```

arcpy.Resample_management(DEM_MinEnvelop_Perimeter_Buf200m,
DEM_MinEnvelop_Perimeter_Buf200m_Resamp_temp, ResampCSize, 'BILINEAR')
arcpy.gp.FocalStatistics_sa(DEM_MinEnvelop_Perimeter_Buf200m_Resamp_temp,
"in_memory/DEM_MinEnvelop_Perimeter_Buf200m_Resamp", 'Weight
H:\Masterarbeit\Geo_Data_Prozessierung\Buehler_OBIA\ArcGIS\LSHM\5x5kernel.txt',
'MEAN', 'DATA')
arcpy.env.snapRaster = "in_memory/DEM_MinEnvelop_Perimeter_Buf200m_Resamp"
arcpy.gp.ExtractByMask_sa("in_memory/DEM_MinEnvelop_Perimeter_Buf200m_Resamp",
"in_memory/Perimeter_Buf200m", DEM_MinEnvelop_Perimeter_Buf200m_Resamp)
arcpy.Resample_management(Forest, Forest_Resamp, ResampCSize, 'BILINEAR')

# create slope
arcpy.AddMessage("create slope...")
arcpy.gp.Slope_sa(DEM_MinEnvelop_Perimeter_Buf200m_Resamp, Slope, 'DEGREE', '1')

# create aspect
arcpy.AddMessage("create aspect...")
arcpy.gp.Aspect_sa(DEM_MinEnvelop_Perimeter_Buf200m_Resamp, Aspect)

# classify aspect into sectors
arcpy.gp.Reclassify_sa(Aspect, 'VALUE', '0 22.5 0; 22.5 67.5 20; 67.5 112.5 40; 112.5 157.5 20;
157.5 202.5 0; 202.5 247.5 -20; 247.5 292.5 -40; 292.5 337.5 -20; 337.5 360 0', Aspect_sectors_N0,
'DATA')
arcpy.gp.Reclassify_sa(Aspect, 'VALUE', '0 22.5 40; 22.5 67.5 20; 67.5 112.5 0; 112.5 157.5 -20;
157.5 202.5 -40; 202.5 247.5 -20; 247.5 292.5 0; 292.5 337.5 20; 337.5 360 40', As-
pect_sectors_Nmax, 'DATA')

# create curvature
arcpy.AddMessage("create curvature...")
arcpy.gp.Curvature_sa(DEM_MinEnvelop_Perimeter_Buf200m_Resamp, Curv, '1', Curv_profile,
Curv_plan)

arcpy.AddMessage("create curvature expanded...")
arcpy.gp.FocalStatistics_sa(Curv_profile, Curv_profile_expanded, 'Rectangle 3 3 CELL', 'MEAN',
'DATA')

arcpy.AddMessage("create curvature expanded...")
arcpy.gp.FocalStatistics_sa(Curv_plan, Curv_plan_expanded, 'Rectangle 3 3 CELL', 'MEAN',
'DATA')

# create hillshade
arcpy.gp.HillShade_sa(DEM_MinEnvelop_Perimeter_Buf200m_Resamp, Hillsh, '315', '45',
'NO_SHADOWS', '1')

```

```

    arcpy.gp.ExtractByMask_sa(Hillsh, Perimeter, Hillsh_Perimeter)
#####
# Function to Export Input Parameters in File
def export_input_para(MethodRun, DEM, Forest, ResampCSize, RoughnessMovWin, SlopeMin,
SlopeMax, RoughnessMax, CurvMax, AreaMin, Perimeter, PRA_Referenz, PRA_Referenz_W):
    import csv
    column1 = ["Name", "DHM", "Wald", "Neuberechnung Zellgroesse [m]", "Groesse Zellfenster Rauig-
keit [ ]", "Min Hangneigung [Deg]", "Max Hangneigung [Deg]", "Max Rauigkeit [ ]", "Max Kurvatur [
]", "Min Anrissflaeche [m2]", "Perimeter", "PRA_Referenz", "PRA_Referenz_W"]
    column2 = [MethodRun, DEM, Forest, ResampCSize, RoughnessMovWin, SlopeMin, SlopeMax,
RoughnessMax, CurvMax, AreaMin, Perimeter, PRA_Referenz, PRA_Referenz_W]
    array = [column1,column2]
    FileName = WorkDir + MethodRun + "\\\" + MethodRun
    with open("%s_Input_Parameters.csv" % FileName,"wb") as f:
        writer = csv.writer(f,delimiter=";")
        writer.writerows(array)
    f.close()

#####
# Function to Raster of final PRA
def raster_of_finalPRA(DEM_MinEnvelop_Perimeter_Buf200m_Resamp, WorkDir, MethodRun):
    arcpy.env.extent = DEM_MinEnvelop_Perimeter_Buf200m_Resamp
    PRA_Methode_final_shp = WorkDir + MethodRun + "\\temp_model\\" + MethodRun +
"_slopebinaryfilter_notrough_notcurv_smooth.shp"
    PRA_Methode_final_tif = WorkDir + MethodRun + "\\\" + MethodRun + "_PRA.tif"
    arcpy.FeatureToRaster_conversion(PRA_Methode_final_shp, 'gridcode',
"in_memory/PRA_Methode_final_tif", ResampCSize)
    arcpy.gp.Reclassify_sa("in_memory/PRA_Methode_final_tif", 'Value', '200 200;NODATA 0',
PRA_Methode_final_tif, 'DATA')

#####
# Function to iterate over Input Parameter
def iterate_over_one_InputPara(rowName, WorkDir,
DEM_MinEnvelop_Perimeter_Buf200m_Resamp, Forest_Resamp, RoughnessMovWin, SlopeMin,
SlopeMax, RoughnessMax, CurvMax, AreaMin, Slope, Aspect, Curv_plan_expanded):
    arcpy.AddMessage("Iteration over " + rowName + ".....")
    cursor = arcpy.SearchCursor(InputPara)
    for row in cursor:
        Value = row.getValue(rowName)
        if Value != None:
            if rowName == "RoughnessMax":
                Value = str(Value)
                Value_mod = Value.replace(".", "p")

```

```

locals_dict = dict(vars())
locals_dict[rowName] = Value
arcpy.AddMessage(str(rowName) + " = " + locals_dict[rowName])
MethodRun = DirMethod + "_" + rowName + Value_mod

if (os.path.isdir(WorkDir + MethodRun) == 0):
    arcpy.CreateFolder_management(WorkDir, MethodRun)
    export_input_para(MethodRun, DEM, Forest, ResampCSize, lo-
locals_dict["RoughnessMovWin"], locals_dict["SlopeMin"], locals_dict["SlopeMax"], lo-
locals_dict["RoughnessMax"], locals_dict["CurvMax"], locals_dict["AreaMin"], Perimeter,
PRA_Referenz, PRA_Referenz_W)
    arcpy.dataprepPRA333(WorkDir + MethodRun, MethodRun,
DEM_MinEnvelop_Perimeter_Buf200m_Resamp, Forest_Resamp, locals_dict["RoughnessMovWin"],
locals_dict["SlopeMin"], locals_dict["SlopeMax"], locals_dict["RoughnessMax"], lo-
locals_dict["CurvMax"], Slope, Aspect, Curv_plan_expanded)
    PRA_Methode = WorkDir + MethodRun + "\\temp_model\\" + MethodRun +
"_slopebinaryfilter_notrough_notcurv_smooth.shp"
    PRA_Methode_NoSmooth = WorkDir + MethodRun + "\\temp_model\\" + MethodRun
+"_slopebinaryfilter_notrough_notcurv.shp"
    arcpy.Valid(WorkDir + MethodRun, MethodRun,
DEM_MinEnvelop_Perimeter_Buf200m_Resamp, Slope, PRA_Methode, PRA_Methode_NoSmooth, Pe-
rimeter, PRA_Referenz, PRA_Referenz_W, locals_dict["AreaMin"])
    raster_of_finalPRA(DEM_MinEnvelop_Perimeter_Buf200m_Resamp, WorkDir,
MethodRun)

elif rowName == "CurvMax":
    Value = str(Value)
    Value_mod = Value.replace(".", "p")
    locals_dict = dict(vars())
    locals_dict[rowName] = Value
    arcpy.AddMessage(str(rowName) + " = " + locals_dict[rowName])
    MethodRun = DirMethod + "_" + rowName + Value_mod

if (os.path.isdir(WorkDir + MethodRun) == 0):
    arcpy.CreateFolder_management(WorkDir, MethodRun)
    export_input_para(MethodRun, DEM, Forest, ResampCSize, lo-
locals_dict["RoughnessMovWin"], locals_dict["SlopeMin"], locals_dict["SlopeMax"], lo-
locals_dict["RoughnessMax"], locals_dict["CurvMax"], locals_dict["AreaMin"], Perimeter,
PRA_Referenz, PRA_Referenz_W)
    arcpy.dataprepPRA333(WorkDir + MethodRun, MethodRun,
DEM_MinEnvelop_Perimeter_Buf200m_Resamp, Forest_Resamp, locals_dict["RoughnessMovWin"],
locals_dict["SlopeMin"], locals_dict["SlopeMax"], locals_dict["RoughnessMax"], lo-
locals_dict["CurvMax"], Slope, Aspect, Curv_plan_expanded)

```



```

        PRA_Methode = WorkDir + MethodRun + "\\temp_model\\" + MethodRun +
"_slopebinaryfilter_notrough_notcurv_smooth.shp"
        PRA_Methode_NoSmooth = WorkDir + MethodRun + "\\temp_model\\" + MethodRun
+ "_slopebinaryfilter_notrough_notcurv.shp"
        arcpy.Valid(WorkDir + MethodRun, MethodRun,
DEM_MinEnvelop_Perimeter_Buf200m_Resamp, Slope, PRA_Methode, PRA_Methode_NoSmooth, Pe-
rimeter, PRA_Referenz, PRA_Referenz_W, locals_dict["AreaMin"])
        raster_of_finalPRA(DEM_MinEnvelop_Perimeter_Buf200m_Resamp, WorkDir,
MethodRun)

else:
    Value = str(int(Value))
    locals_dict = dict(vars())
    locals_dict[rowName] = Value
    arcpy.AddMessage(str(rowName) + " = " + locals_dict[rowName])
    MethodRun = DirMethod + "_" + rowName + Value

    if (os.path.isdir(WorkDir + MethodRun) == 0):
        arcpy.CreateFolder_management(WorkDir, MethodRun)
        export_input_para(MethodRun, DEM, Forest, ResampCSize, lo-
cals_dict["RoughnessMovWin"], locals_dict["SlopeMin"], locals_dict["SlopeMax"], lo-
cals_dict["RoughnessMax"], locals_dict["CurvMax"], locals_dict["AreaMin"], Perimeter,
PRA_Referenz, PRA_Referenz_W)
        arcpy.dataprepPRA333(WorkDir + MethodRun, MethodRun,
DEM_MinEnvelop_Perimeter_Buf200m_Resamp, Forest_Resamp, locals_dict["RoughnessMovWin"],
locals_dict["SlopeMin"], locals_dict["SlopeMax"], locals_dict["RoughnessMax"], lo-
cals_dict["CurvMax"], Slope, Aspect, Curv_plan_expanded)
        PRA_Methode = WorkDir + MethodRun + "\\temp_model\\" + MethodRun +
"_slopebinaryfilter_notrough_notcurv_smooth.shp"
        PRA_Methode_NoSmooth = WorkDir + MethodRun + "\\temp_model\\" + MethodRun
+ "_slopebinaryfilter_notrough_notcurv.shp"
        arcpy.Valid(WorkDir + MethodRun, MethodRun,
DEM_MinEnvelop_Perimeter_Buf200m_Resamp, Slope, PRA_Methode, PRA_Methode_NoSmooth, Pe-
rimeter, PRA_Referenz, PRA_Referenz_W, locals_dict["AreaMin"])
        raster_of_finalPRA(DEM_MinEnvelop_Perimeter_Buf200m_Resamp, WorkDir,
MethodRun)

#####
if TypeofRun == "m":

    # Execute Functions, which iterate over Input Parameter

```

```

    iterate_over_one_InputPara("SlopeMin", WorkDir,
DEM_MinEnvelop_Perimeter_Buf200m_Resamp, Forest_Resamp, RoughnessMovWin, SlopeMin,
SlopeMax, RoughnessMax, CurvMax, AreaMin, Slope, Aspect, Curv_plan_expanded)
    iterate_over_one_InputPara("SlopeMax", WorkDir,
DEM_MinEnvelop_Perimeter_Buf200m_Resamp, Forest_Resamp, RoughnessMovWin, SlopeMin,
SlopeMax, RoughnessMax, CurvMax, AreaMin, Slope, Aspect, Curv_plan_expanded)
    iterate_over_one_InputPara("RoughnessMovWin", WorkDir,
DEM_MinEnvelop_Perimeter_Buf200m_Resamp, Forest_Resamp, RoughnessMovWin, SlopeMin,
SlopeMax, RoughnessMax, CurvMax, AreaMin, Slope, Aspect, Curv_plan_expanded)
##  iterate_over_one_InputPara("AreaMin", WorkDir,
DEM_MinEnvelop_Perimeter_Buf200m_Resamp, Forest_Resamp, RoughnessMovWin, SlopeMin,
SlopeMax, RoughnessMax, CurvMax, AreaMin, Slope, Aspect, Curv_plan_expanded)
    iterate_over_one_InputPara("CurvMax", WorkDir, DEM_MinEnvelop_Perimeter_Buf200m_Resamp,
Forest_Resamp, RoughnessMovWin, SlopeMin, SlopeMax, RoughnessMax, CurvMax, AreaMin, Slope,
Aspect, Curv_plan_expanded)
    iterate_over_one_InputPara("RoughnessMax", WorkDir,
DEM_MinEnvelop_Perimeter_Buf200m_Resamp, Forest_Resamp, RoughnessMovWin, SlopeMin,
SlopeMax, RoughnessMax, CurvMax, AreaMin, Slope, Aspect, Curv_plan_expanded)

#####
if TypeofRun == "s":

    i = 1
    MethodRun = (DirMethod + "_bestInputPara_" + str(i))

    while (os.path.isdir(WorkDir + MethodRun) == 1):
        i = i+1
        MethodRun = DirMethod + "_bestInputPara_" + str(i)

    # Execute Run with best Input Parameters
    arcpy.AddMessage(MethodRun)
    arcpy.CreateFolder_management(WorkDir, MethodRun)
    export_input_para(MethodRun, DEM, Forest, ResampCSize, RoughnessMovWin, SlopeMin, Slope-
Max, RoughnessMax, CurvMax, AreaMin, Perimeter, PRA_Referenz, PRA_Referenz_W)
    arcpy.dataprepPRA333(WorkDir + MethodRun, MethodRun,
DEM_MinEnvelop_Perimeter_Buf200m_Resamp, Forest_Resamp, RoughnessMovWin, SlopeMin,
SlopeMax, RoughnessMax, CurvMax, Slope, Aspect, Curv_plan_expanded)
    PRA_Methode = WorkDir + MethodRun + "\\temp_model\\" + MethodRun +
"_slopebinaryfilter_notrough_notcurv_smooth.shp"
    PRA_Methode_NoSmooth = WorkDir + MethodRun + "\\temp_model\\" + MethodRun
+"_slopebinaryfilter_notrough_notcurv.shp"

```

```
arcpy.Valid(WorkDir + MethodRun, MethodRun, DEM_MinEnvelop_Perimeter_Buf200m_Resamp,  
Slope, PRA_Methode, PRA_Methode_NoSmooth, Perimeter, PRA_Referenz, PRA_Referenz_W, AreaM-  
in)
```

### 9.1.6 Validierung

```
# -*- coding: utf-8 -*-  
  
# -----  
  
# Valid.py  
  
# Created on: 2017-10-22 21:08:57.00000  
  
# (generated by ArcGIS/ModelBuilder)  
  
# Usage: Slope <DEM> <Forest>  
  
# Description:  
# -----  
  
  
# Set the necessary product code  
  
# Import arcinfo  
  
import os  
  
  
# Import arcpy module  
  
import arcpy  
  
  
# Script arguments  
WorkDir = arcpy.GetParameterAsText(0) + "\\\"  
MethodRun = arcpy.GetParameterAsText(1)  
DEM = arcpy.GetParameterAsText(2)  
Slope = arcpy.GetParameterAsText(3)  
PRA_Methode = arcpy.GetParameterAsText(4)  
PRA_Methode_NoSimplify = arcpy.GetParameterAsText(5)  
Perimeter = arcpy.GetParameterAsText(6)  
PRA_Referenz = arcpy.GetParameterAsText(7)  
PRA_Referenz_W =arcpy.GetParameterAsText(8)  
AreaMin = arcpy.GetParameterAsText(9)  
  
  
# Local variables:  
  
Perimeter_AreaAdded = Perimeter
```

```

WorkDir_temp_valid = WorkDir + "\\temp_valid\\"
PRA_Referenz_Clip_shp = WorkDir_temp_valid + "PRA_Referenz_Clip.shp"
PRA_Methode_Clip_shp = WorkDir_temp_valid + "PRA_Methode_Clip.shp"
PRA_Methode_Clip_singlepart_shp = WorkDir_temp_valid + "PRA_Methode_Clip_singlepart.shp"
PRA_Methode_Clip_Area_shp = WorkDir_temp_valid + "PRA_Methode_Clip_Area.shp"
PRA_Methode_Clip_Area_out_shp = WorkDir + "\\\" + MethodRun + "_PRA.shp"
PRA_Referenz_PRA_Methode_shp = WorkDir_temp_valid + "PRA_Referenz_PRA_Methode.shp"
PRA_Referenz_PRA_Methode_Diss_shp = WorkDir_temp_valid +
"PRA_Referenz_PRA_Methode_Diss.shp"
PRA_Referenz_PRA_Methode_Diss_AreaAdded_shp = PRA_Referenz_PRA_Methode_Diss_shp
NoPRA_Referenz_PRA_Methode_shp = WorkDir_temp_valid + "NoPRA_Referenz_PRA_Methode.shp"
NoPRA_Referenz_PRA_Methode_Diss_shp = WorkDir_temp_valid +
"NoPRA_Referenz_PRA_Methode_Diss.shp"
NoPRA_Referenz_PRA_Methode_Diss_AreaAdded_shp = NoPRA_Referenz_PRA_Methode_Diss_shp
PRA_Referenz_NoPRA_Methode_shp = WorkDir_temp_valid + "PRA_Referenz_NoPRA_Methode.shp"
PRA_Referenz_NoPRA_Methode_Diss_shp = WorkDir_temp_valid +
"PRA_Referenz_NoPRA_Methode_Diss.shp"
PRA_Referenz_NoPRA_Methode_Diss_AreaAdded_shp = PRA_Referenz_NoPRA_Methode_Diss_shp
NoPRA_Referenz_shp = WorkDir_temp_valid + "NoPRA_Referenz.shp"
NoPRA_Methode_shp = WorkDir_temp_valid + "NoPRA_Methode.shp"
NoPRA_Referenz_NoPRA_Methode_shp = WorkDir_temp_valid +
"NoPRA_Referenz_NoPRA_Methode.shp"
NoPRA_Referenz_NoPRA_Methode_Diss_shp = WorkDir_temp_valid +
"NoPRA_Referenz_NoPRA_Methode_Diss.shp"
NoPRA_Referenz_NoPRA_Methode_Diss_AreaAdded_shp =
NoPRA_Referenz_NoPRA_Methode_Diss_shp
ALL_PRA_NoPRA_shp = WorkDir_temp_valid + "ALL_Diss.shp"
ALL_PRA_NoPRA_csv = WorkDir + MethodRun + "_Kl_PRA_NoPRA.csv"
ZonalStat_SinglePoly_DEM = "in_memory/ZonalStat_DEM"
ZonalStat_OverallPoly_DEM_temp = WorkDir + MethodRun +
"_ZonalStat_OverallPoly_DEM_temp.csv"
ZonalStat_OverallPoly_DEM = WorkDir + MethodRun + "_ZonalStat_OverallPoly_DEM.xls"
ZonalStat_SinglePoly_Slope = "in_memory/ZonalStat_Slo"

```

```
ZonalStat_OverallPoly_Slope_temp = WorkDir + MethodRun +
"_ZonalStat_OverallPoly_Slope_temp.csv"
ZonalStat_OverallPoly_Slope = WorkDir + MethodRun + "_ZonalStat_OverallPoly_Slope.xls"
```

```
PRA_Referenz_PRA_Methode_Diss_temp_shp = WorkDir_temp_valid +
"PRA_Referenz_PRA_Methode_Diss_temp.shp"
```

```
NoPRA_Referenz_PRA_Methode_Diss_temp_shp = WorkDir_temp_valid +
"NoPRA_Referenz_PRA_Methode_Diss_temp.shp"
```

```
PRA_Referenz_NoPRA_Methode_Diss_temp_shp = WorkDir_temp_valid +
"PRA_Referenz_NoPRA_Methode_Diss_temp.shp"
```

```
NoPRA_Referenz_NoPRA_Methode_Diss_temp_shp = WorkDir_temp_valid +
"NoPRA_Referenz_NoPRA_Methode_Diss_temp.shp"
```

```
PRA_Referenz_W_Clip_shp = WorkDir_temp_valid + "PRA_Referenz_W_Clip.shp"
```

```
PRA_Referenz_W_PRA_Methode_shp = WorkDir_temp_valid + "PRA_Referenz_W_PRA_Methode.shp"
```

```
PRA_Referenz_W_PRA_Methode_Diss_shp = WorkDir_temp_valid +
"PRA_Referenz_W_PRA_Methode_Diss.shp"
```

```
PRA_Referenz_W_NoPRA_Methode_shp = WorkDir_temp_valid +
"PRA_Referenz_W_NoPRA_Methode.shp"
```

```
PRA_Referenz_W_NoPRA_Methode_Diss_shp = WorkDir_temp_valid +
"PRA_Referenz_W_NoPRA_Methode_Diss.shp"
```

```
ALL_W_PRA_NoPRA_shp = WorkDir_temp_valid + "ALL_W_Diss.shp"
```

```
ALL_W_PRA_NoPRA_csv = WorkDir + MethodRun + "_Kl_PRA_NoPRA_W.csv"
```

```
# Set Geoprocessing environments
```

```
##arcpy.env.outputCoordinateSystem = "PRO-
JCS['CH1903_LV03',GEOGCS['GCS_CH1903',DATUM['D_CH1903',SPHEROID['Bessel_1841',6377397.1
55,299.1528128]],PRIMEM['Greenwich',0.0],UNIT['Degree',0.0174532925199433]],PROJECTION['H
otine_Oblique_Mercator_Azimuth_Center'],PARAMETER['False_Easting',600000.0],PARAMETER['Fal
se_Northing',200000.0],PARAMETER['Scale_Factor',1.0],PARAMETER['Azimuth',90.0],PARAMETER['
Longi-
tude_Of_Center',7.439583333333333],PARAMETER['Latitude_Of_Center',46.95240555555556],UNI
T['Meter',1.0]]"
```

```
##arcpy.env.extent = Perimeter
```

```
arcpy.env.workspace = WorkDir_temp_valid
```

```
# Process: Create Folder
```

```
arcpy.CreateFolder_management(WorkDir, "temp_valid")
```

```
#####
```

```
# Unweighted Area
```

```
#####
```

```
# Process: Add Geometry Attributes (5)
```

```
arcpy.AddGeometryAttributes_management(Perimeter, "AREA;AREA_GEODESIC", "METERS",  
"SQUARE_METERS", "PRO-  
JCS['CH1903_LV03',GEOGCS['GCS_CH1903',DATUM['D_CH1903',SPHEROID['Bessel_1841',6377397.1  
55,299.1528128]],PRIMEM['Greenwich',0.0],UNIT['Degree',0.0174532925 199433]],PROJECTION['H  
otline_Oblique_Mercator_Azimuth_Center'],PARAMETER['False_Easting',600000.0],PARAMETER['Fal  
se_Northing',200000.0],PARAMETER['Scale_Factor',1.0],PARAMETER['Azimuth',90.0],PARAMETER['  
Longi-  
tude_Of_Center',7.439583333333333],PARAMETER['Latitude_Of_Center',46.95240555555556],UNI  
T['Meter',1.0]]")
```

```
# Process: Clip
```

```
arcpy.Clip_analysis(PRA_Referenz, Perimeter, PRA_Referenz_Clip_shp, "")
```

```
# Process: Clip (2)
```

```
arcpy.Clip_analysis(PRA_Methode, Perimeter, PRA_Methode_Clip_shp, "")
```

```
# Multipart to Singlepart
```

```
arcpy.MultipartToSinglepart_management(PRA_Methode_Clip_shp,  
PRA_Methode_Clip_singlepart_shp)
```

```
# Process: Add Geometry Attributes (3)
```

```
arcpy.AddGeometryAttributes_management(PRA_Methode_Clip_singlepart_shp, "AR-  
EA;AREA_GEODESIC", "METERS", "SQUARE_METERS", "PRO-  
JCS['CH1903_LV03',GEOGCS['GCS_CH1903',DATUM['D_CH1903',SPHEROID['Bessel_1841',6377397.1
```

```
55,299.1528128]],PRIMEM['Greenwich',0.0],UNIT['Degree',0.0174532925199433]],PROJECTION['H  
otine_Oblique_Mercator_Azimuth_Center'],PARAMETER['False_Easting',600000.0],PARAMETER['Fal  
se_Northing',200000.0],PARAMETER['Scale_Factor',1.0],PARAMETER['Azimuth',90.0],PARAMETER['  
Longi-  
tude_Of_Center',7.439583333333333],PARAMETER['Latitude_Of_Center',46.95240555555556],UNI  
T['Meter',1.0]]")
```

```
# Process: Select
```

```
AreaMin_mod = str(float(AreaMin) - 0)
```

```
arcpy.Select_analysis(PRA_Methode_Clip_singlepart_shp, PRA_Methode_Clip_Area_shp, "POLY_AREA  
>=%s" % AreaMin_mod)
```

```
# Execute CopyFeatures
```

```
arcpy.CopyFeatures_management(PRA_Methode_Clip_Area_shp, PRA_Methode_Clip_Area_out_shp)
```

```
### Process: Clip (2)
```

```
##arcpy.Clip_analysis(PRA_Methode, Perimeter, PRA_Methode_Clip_out_shp, "")
```

```
# Process: Intersect
```

```
arcpy.Intersect_analysis([PRA_Referenz_Clip_shp, PRA_Methode_Clip_Area_shp],  
PRA_Referenz_PRA_Methode_shp, "ALL", "", "INPUT")
```

```
# Process: Dissolve
```

```
arcpy.Dissolve_management(PRA_Referenz_PRA_Methode_shp,  
PRA_Referenz_PRA_Methode_Diss_shp, "", "", "MULTI_PART", "DISSOLVE_LINES")
```

```
# Process: Erase (2)
```

```
arcpy.Erase_analysis(PRA_Methode_Clip_Area_shp, PRA_Referenz_Clip_shp,  
NoPRA_Referenz_PRA_Methode_shp, "")
```

```
# Process: Dissolve (2)
```

```
arcpy.Dissolve_management(NoPRA_Referenz_PRA_Methode_shp,  
NoPRA_Referenz_PRA_Methode_Diss_shp, "", "", "MULTI_PART", "DISSOLVE_LINES")
```



```

# Process: Erase

arcpy.Erase_analysis(PRA_Referenz_Clip_shp, PRA_Methode_Clip_Area_shp,
PRA_Referenz_NoPRA_Methode_shp, "")

# Process: Dissolve (3)

arcpy.Dissolve_management(PRA_Referenz_NoPRA_Methode_shp,
PRA_Referenz_NoPRA_Methode_Diss_shp, "", "", "MULTI_PART", "DISSOLVE_LINES")

# Process: Erase (3)

arcpy.Erase_analysis(Perimeter, PRA_Referenz_Clip_shp, NoPRA_Referenz_shp, "")

# Process: Erase (4)

arcpy.Erase_analysis(Perimeter, PRA_Methode_Clip_Area_shp, NoPRA_Methode_shp, "")

# Process: Intersect (2)

arcpy.Intersect_analysis([NoPRA_Referenz_shp, NoPRA_Methode_shp],
NoPRA_Referenz_NoPRA_Methode_shp, "ALL", "", "INPUT")

# Process: Dissolve (4)

arcpy.Dissolve_management(NoPRA_Referenz_NoPRA_Methode_shp,
NoPRA_Referenz_NoPRA_Methode_Diss_shp, "", "", "MULTI_PART", "DISSOLVE_LINES")

# Check if Dissolve worked

result = arcpy.GetCount_management(PRA_Referenz_PRA_Methode_Diss_shp)
count_1 = int(result.getOutput(0))

if count_1 > 1:

    # Execute Copy

    arcpy.Copy_management(PRA_Referenz_PRA_Methode_Diss_shp,
PRA_Referenz_PRA_Methode_Diss_temp_shp)

    # Process: Dissolve (2)

    arcpy.Dissolve_management(PRA_Referenz_PRA_Methode_Diss_temp_shp,
PRA_Referenz_PRA_Methode_Diss_shp, "", "", "MULTI_PART", "DISSOLVE_LINES")

```

```

# Check if Dissolve worked

result = arcpy.GetCount_management(NoPRA_Referenz_PRA_Methode_Diss_shp)
count_1 = int(result.getOutput(0))
if count_1 > 1:
    # Execute Copy

    arcpy.Copy_management(NoPRA_Referenz_PRA_Methode_Diss_shp,
NoPRA_Referenz_PRA_Methode_Diss_temp_shp)

    # Process: Dissolve (2)

    arcpy.Dissolve_management(NoPRA_Referenz_PRA_Methode_Diss_temp_shp,
NoPRA_Referenz_PRA_Methode_Diss_shp, "", "", "MULTI_PART", "DISSOLVE_LINES")

result = arcpy.GetCount_management(PRA_Referenz_NoPRA_Methode_Diss_shp)
count_2 = int(result.getOutput(0))
if count_2 > 1:
    # Execute Copy

    arcpy.Copy_management(PRA_Referenz_NoPRA_Methode_Diss_shp,
PRA_Referenz_NoPRA_Methode_Diss_temp_shp)

    # Process: Dissolve (2)

    arcpy.Dissolve_management(PRA_Referenz_NoPRA_Methode_Diss_temp_shp,
PRA_Referenz_NoPRA_Methode_Diss_shp, "", "", "MULTI_PART", "DISSOLVE_LINES")

# Check if Dissolve worked

result = arcpy.GetCount_management(NoPRA_Referenz_NoPRA_Methode_Diss_shp)
count_1 = int(result.getOutput(0))
if count_1 > 1:
    # Execute Copy

    arcpy.Copy_management(NoPRA_Referenz_NoPRA_Methode_Diss_shp,
NoPRA_Referenz_NoPRA_Methode_Diss_temp_shp)

    # Process: Dissolve (2)

    arcpy.Dissolve_management(NoPRA_Referenz_NoPRA_Methode_Diss_temp_shp,
NoPRA_Referenz_NoPRA_Methode_Diss_shp, "", "", "MULTI_PART", "DISSOLVE_LINES")

# Process: Add Geometry Attributes (3)

```

```
arcpy.AddGeometryAttributes_management(PRA_Referenz_PRA_Methode_Diss_shp, "AR-  
EA;AREA_GEODESIC", "METERS", "SQUARE_METERS", "PRO-  
JCS['CH1903_LV03',GEOGCS['GCS_CH1903',DATUM['D_CH1903',SPHEROID['Bessel_1841',6377397.1  
55,299.1528128]],PRIMEM['Greenwich',0.0],UNIT['Degree',0.0174532925199433]],PROJECTION['H  
otine_Oblique_Mercator_Azimuth_Center'],PARAMETER['False_Easting',600000.0],PARAMETER['Fal  
se_Northing',200000.0],PARAMETER['Scale_Factor',1.0],PARAMETER['Azimuth',90.0],PARAMETER['  
Longi-  
tude_Of_Center',7.439583333333333],PARAMETER['Latitude_Of_Center',46.95240555555556],UNI  
T['Meter',1.0]]")
```

```
arcpy.CalculateField_management(PRA_Referenz_PRA_Methode_Diss_shp, 'Id', '1', 'VB', '#')
```

```
# Process: Add Geometry Attributes (4)
```

```
arcpy.AddGeometryAttributes_management(NoPRA_Referenz_PRA_Methode_Diss_shp, "AR-  
EA;AREA_GEODESIC", "METERS", "SQUARE_METERS", "PRO-  
JCS['CH1903_LV03',GEOGCS['GCS_CH1903',DATUM['D_CH1903',SPHEROID['Bessel_1841',6377397.1  
55,299.1528128]],PRIMEM['Greenwich',0.0],UNIT['Degree',0.0174532925199433]],PROJECTION['H  
otine_Oblique_Mercator_Azimuth_Center'],PARAMETER['False_Easting',600000.0],PARAMETER['Fal  
se_Northing',200000.0],PARAMETER['Scale_Factor',1.0],PARAMETER['Azimuth',90.0],PARAMETER['  
Longi-  
tude_Of_Center',7.439583333333333],PARAMETER['Latitude_Of_Center',46.95240555555556],UNI  
T['Meter',1.0]]")
```

```
arcpy.CalculateField_management(NoPRA_Referenz_PRA_Methode_Diss_shp, 'Id', '2', 'VB', '#')
```

```
# Process: Add Geometry Attributes
```

```
arcpy.AddGeometryAttributes_management(PRA_Referenz_NoPRA_Methode_Diss_shp, "AR-  
EA;AREA_GEODESIC", "METERS", "SQUARE_METERS", "PRO-  
JCS['CH1903_LV03',GEOGCS['GCS_CH1903',DATUM['D_CH1903',SPHEROID['Bessel_1841',6377397.1  
55,299.1528128]],PRIMEM['Greenwich',0.0],UNIT['Degree',0.0174532925199433]],PROJECTION['H  
otine_Oblique_Mercator_Azimuth_Center'],PARAMETER['False_Easting',600000.0],PARAMETER['Fal  
se_Northing',200000.0],PARAMETER['Scale_Factor',1.0],PARAMETER['Azimuth',90.0],PARAMETER['  
Longi-  
tude_Of_Center',7.439583333333333],PARAMETER['Latitude_Of_Center',46.95240555555556],UNI  
T['Meter',1.0]]")
```

```
arcpy.CalculateField_management(PRA_Referenz_NoPRA_Methode_Diss_shp, 'Id', '3', 'VB', '#')
```

```
# Process: Add Geometry Attributes (2)
```

```
arcpy.AddGeometryAttributes_management(NoPRA_Referenz_NoPRA_Methode_Diss_shp, "AR-  
EA;AREA_GEODESIC", "METERS", "SQUARE_METERS", "PRO-
```

```
JCS['CH1903_LV03',GEOGCS['GCS_CH1903',DATUM['D_CH1903',SPHEROID['Bessel_1841',6377397.155,299.1528128]],PRIMEM['Greenwich',0.0],UNIT['Degree',0.0174532925199433]],PROJECTION['Hotine_Oblique_Mercator_Azimuth_Center'],PARAMETER['False_Easting',600000.0],PARAMETER['False_Northing',200000.0],PARAMETER['Scale_Factor',1.0],PARAMETER['Azimuth',90.0],PARAMETER['Longitude_Of_Center',7.439583333333333],PARAMETER['Latitude_Of_Center',46.95240555555556],UNIT['Meter',1.0]]")
```

```
arcpy.CalculateField_management(NoPRA_Referenz_NoPRA_Methode_Diss_shp, 'Id', '4', 'VB', '#')
```

```
# Process: Merge
```

```
arcpy.Merge_management([PRA_Referenz_PRA_Methode_Diss_shp, NoPRA_Referenz_PRA_Methode_Diss_shp, PRA_Referenz_NoPRA_Methode_Diss_shp, NoPRA_Referenz_NoPRA_Methode_Diss_shp], ALL_PRA_NoPRA_shp, "POLY_AREA \"POLY_AREA\" true true false 6029364 Double 6226020 6881280 ,First,#,PRA_Referenz_PRA_Methode_Diss_shp,POLY_AREA,-1,-1,NoPRA_Referenz_PRA_Methode_Diss_shp,POLY_AREA,-1,-1,PRA_Referenz_NoPRA_Methode_Diss_shp,POLY_AREA,-1,-1,NoPRA_Referenz_NoPRA_Methode_Diss_shp,POLY_AREA,-1,-1;AREA_GEO \"AREA_GEO\" true true false 6357078 Double 6226020 6881388 ,First,#,PRA_Referenz_PRA_Methode_Diss_shp,AREA_GEO,-1,-1,NoPRA_Referenz_PRA_Methode_Diss_shp,AREA_GEO,-1,-1,PRA_Referenz_NoPRA_Methode_Diss_shp,AREA_GEO,-1,-1,NoPRA_Referenz_NoPRA_Methode_Diss_shp,AREA_GEO,-1,-1")
```

```
# Process: Export Feature Attribute to ASCII
```

```
arcpy.ExportXYv_stats(ALL_PRA_NoPRA_shp, "Id;POLY_AREA", "SEMI-COLON", ALL_PRA_NoPRA_csv, "ADD_FIELD_NAMES")
```

```
#####
```

```
# Weighted Area
```

```
#####
```

```
# Process: Clip
```

```
arcpy.Clip_analysis(PRA_Referenz_W, Perimeter, PRA_Referenz_W_Clip_shp, "")
```

```
# Process: Intersect
```

```
arcpy.Intersect_analysis([PRA_Referenz_W_Clip_shp, PRA_Methode_Clip_Area_shp],
PRA_Referenz_W_PRA_Methode_shp, "ALL", "", "INPUT")
```

```
# Process: Add Field
```

```
arcpy.AddField_management(PRA_Referenz_W_PRA_Methode_shp, "AREA_W", "DOUBLE", "", "", "",
"", "NULLABLE", "NON_REQUIRED", "")
```

```
arcpy.AddGeometryAttributes_management(PRA_Referenz_W_PRA_Methode_shp, "AR-
EA;AREA_GEODESIC", "METERS", "SQUARE_METERS", "PRO-
JCS['CH1903_LV03',GEOGCS['GCS_CH1903',DATUM['D_CH1903',SPHEROID['Bessel_1841',6377397.1
55,299.1528128]],PRIMEM['Greenwich',0.0],UNIT['Degree',0.0174532925199433]],PROJECTION['H
otine_Oblique_Mercator_Azimuth_Center'],PARAMETER['False_Easting',600000.0],PARAMETER['Fal
se_Northing',200000.0],PARAMETER['Scale_Factor',1.0],PARAMETER['Azimuth',90.0],PARAMETER['
Longi-
tude_Of_Center',7.4395833333333333],PARAMETER['Latitude_Of_Center',46.952405555555556],UNI
T['Meter',1.0]]")
```

```
# Calculate Field
```

```
arcpy.CalculateField_management(PRA_Referenz_W_PRA_Methode_shp, 'AREA_W', '[Join_Count] *
[POLY_AREA]', 'VB', '#')
```

```
# Process: Dissolve
```

```
arcpy.Dissolve_management(PRA_Referenz_W_PRA_Methode_shp,
PRA_Referenz_W_PRA_Methode_Diss_shp, "", "AREA_W SUM", "MULTI_PART", "DISSOLVE_LINES")
```

```
# Process: Add Field
```

```
arcpy.AddField_management(PRA_Referenz_W_PRA_Methode_Diss_shp, "POLY_AREA", "DOUBLE",
"", "", "", "", "NULLABLE", "NON_REQUIRED", "")
```

```
arcpy.AddField_management(PRA_Referenz_W_PRA_Methode_Diss_shp, 'Id', 'SHORT', '#', '#', '#', '#',
'NULLABLE', 'NON_REQUIRED', '#')
```

```
# Calculate Field
```

```
arcpy.CalculateField_management(PRA_Referenz_W_PRA_Methode_Diss_shp, 'POLY_AREA',
'[SUM_AREA_W]', 'VB', '#')
```

```
arcpy.CalculateField_management(PRA_Referenz_W_PRA_Methode_Diss_shp, 'Id', '1', 'VB', '#')
```

```
# Process: Erase
```

```
arcpy.Erase_analysis(PRA_Referenz_W_Clip_shp, PRA_Methode_Clip_Area_shp,  
PRA_Referenz_W_NoPRA_Methode_shp, "")
```

```
# Process: Add Field
```

```
arcpy.AddField_management(PRA_Referenz_W_NoPRA_Methode_shp, "AREA_W", "DOUBLE", "", "",  
"", "", "NULLABLE", "NON_REQUIRED", "")
```

```
arcpy.AddGeometryAttributes_management(PRA_Referenz_W_NoPRA_Methode_shp, "AR-  
EA;AREA_GEODESIC", "METERS", "SQUARE_METERS", "PRO-  
JCS['CH1903_LV03',GEOGCS['GCS_CH1903',DATUM['D_CH1903',SPHEROID['Bessel_1841',6377397.1  
55,299.1528128]],PRIMEM['Greenwich',0.0],UNIT['Degree',0.0174532925199433]],PROJECTION['H  
otone_Oblique_Mercator_Azimuth_Center'],PARAMETER['False_Easting',600000.0],PARAMETER['Fal  
se_Northing',200000.0],PARAMETER['Scale_Factor',1.0],PARAMETER['Azimuth',90.0],PARAMETER['  
Longi-  
tude_Of_Center',7.439583333333333],PARAMETER['Latitude_Of_Center',46.95240555555556],UNI  
T['Meter',1.0]]")
```

```
# Calculate Field
```

```
arcpy.CalculateField_management(PRA_Referenz_W_NoPRA_Methode_shp, 'AREA_W', '[Join_Count] *  
[POLY_AREA]', 'VB', '#')
```

```
# Process: Dissolve (3)
```

```
arcpy.Dissolve_management(PRA_Referenz_W_NoPRA_Methode_shp,  
PRA_Referenz_W_NoPRA_Methode_Diss_shp, "", "AREA_W SUM", "MULTI_PART", "DISSOLVE_LINES")
```

```
# Process: Add Field
```

```
arcpy.AddField_management(PRA_Referenz_W_NoPRA_Methode_Diss_shp, "POLY_AREA", "DOUBLE",  
"", "", "", "", "NULLABLE", "NON_REQUIRED", "")
```

```
arcpy.AddField_management(PRA_Referenz_W_NoPRA_Methode_Diss_shp, 'Id', 'SHORT', '#', '#', '#',  
'#', 'NULLABLE', 'NON_REQUIRED', '#')
```

```
# Calculate Field
```

```
arcpy.CalculateField_management(PRA_Referenz_W_NoPRA_Methode_Diss_shp, 'POLY_AREA',  
'[SUM_AREA_W]', 'VB', '#')
```

```
arcpy.CalculateField_management(PRA_Referenz_W_NoPRA_Methode_Diss_shp, 'Id', '3', 'VB', '#')
```

```
# Process: Merge
```

```
arcpy.Merge_management([PRA_Referenz_W_PRA_Methode_Diss_shp,  
NoPRA_Referenz_PRA_Methode_Diss_shp, PRA_Referenz_W_NoPRA_Methode_Diss_shp,  
NoPRA_Referenz_NoPRA_Methode_Diss_shp], ALL_W_PRA_NoPRA_shp, "POLY_AREA \"POLY_AREA\""  
true true false 6029364 Double 6226020 6881280  
,First,#,PRA_Referenz_W_PRA_Methode_Diss_shp,POLY_AREA,-1,-  
1,NoPRA_Referenz_PRA_Methode_Diss_shp,POLY_AREA,-1,-  
1,PRA_Referenz_W_NoPRA_Methode_Diss_shp,POLY_AREA,-1,-1")
```

```
# Process: Export Feature Attribute to ASCII
```

```
arcpy.ExportXYv_stats(ALL_W_PRA_NoPRA_shp, "Id;POLY_AREA", "SEMI-COLON",  
ALL_W_PRA_NoPRA_csv, "ADD_FIELD_NAMES")
```

```
#####
```

```
# Intersection PRA with Referenz
```

```
#####
```

```
PRA_Referenz_Diss = WorkDir_temp_valid + "PRA_Referenz_Diss.shp"
```

```
PRA_Methode_Referenz_Inters = WorkDir_temp_valid + "PRA_Methode_Referenz_Inters.shp"
```

```
PRA_csv = WorkDir + MethodRun + "_PRA.csv"
```

```
### Process: Add Geometry Attributes
```

```
##arcpy.AddGeometryAttributes_management(PRA_Methode_Clip_Area_shp, "AR-  
EA;AREA_GEODESIC", "METERS", "SQUARE_METERS", "PRO-  
JCS['CH1903_LV03',GEOGCS['GCS_CH1903',DATUM['D_CH1903',SPHEROID['Bessel_1841',6377397.1  
55,299.1528128]],PRIMEM['Greenwich',0.0],UNIT['Degree',0.0174532925199433]],PROJECTION['H  
otine_Oblique_Mercator_Azimuth_Center'],PARAMETER['False_Easting',600000.0],PARAMETER['Fal  
se_Northing',200000.0],PARAMETER['Scale_Factor',1.0],PARAMETER['Azimuth',90.0],PARAMETER['  
Longi-  
tude_Of_Center',7.439583333333333],PARAMETER['Latitude_Of_Center',46.95240555555556],UNI  
T['Meter',1.0]]")
```

```
# Process: Dissolve
```

```
arcpy.Dissolve_management(PRA_Referenz, PRA_Referenz_Diss, "", "", "MULTI_PART", "DIS-  
SOLVE_LINES")
```

```
# Process: Add Geometry Attributes
```

```
arcpy.AddGeometryAttributes_management(PRA_Referenz_Diss, "AREA;AREA_GEODESIC", "ME-  
TERS", "SQUARE_METERS", "PRO-  
JCS['CH1903_LV03',GEOGCS['GCS_CH1903',DATUM['D_CH1903',SPHEROID['Bessel_1841',6377397.1  
55,299.1528128]],PRIMEM['Greenwich',0.0],UNIT['Degree',0.0174532925199433]],PROJECTION['H  
otine_Oblique_Mercator_Azimuth_Center'],PARAMETER['False_Easting',600000.0],PARAMETER['Fal  
se_Northing',200000.0],PARAMETER['Scale_Factor',1.0],PARAMETER['Azimuth',90.0],PARAMETER['  
Longi-  
tude_Of_Center',7.439583333333333],PARAMETER['Latitude_Of_Center',46.95240555555556],UNI  
T['Meter',1.0]]")
```

```
# Process: Intersect
```

```
arcpy.Intersect_analysis([PRA_Methode_Clip_Area_shp, PRA_Referenz_Diss],  
PRA_Methode_Referenz_Inters, "ONLY_FID", "", "INPUT")
```

```
# Process: Add Geometry Attributes
```

```
arcpy.AddGeometryAttributes_management(PRA_Methode_Referenz_Inters, "AR-  
EA;AREA_GEODESIC", "METERS", "SQUARE_METERS", "PRO-  
JCS['CH1903_LV03',GEOGCS['GCS_CH1903',DATUM['D_CH1903',SPHEROID['Bessel_1841',6377397.1  
55,299.1528128]],PRIMEM['Greenwich',0.0],UNIT['Degree',0.0174532925199433]],PROJECTION['H  
otine_Oblique_Mercator_Azimuth_Center'],PARAMETER['False_Easting',600000.0],PARAMETER['Fal  
se_Northing',200000.0],PARAMETER['Scale_Factor',1.0],PARAMETER['Azimuth',90.0],PARAMETER['  
Longi-  
tude_Of_Center',7.439583333333333],PARAMETER['Latitude_Of_Center',46.95240555555556],UNI  
T['Meter',1.0]]")
```

```
### Process: Join Field
```

```
##seqMethodRun = MethodRun[0:6]
```

```
##arcpy.JoinField_management(PRA_Methode_Clip_Area_shp, "FID", PRA_Methode_Referenz_Inters,  
"FID_%s" %seqMethodRun, "POLY_AREA;AREA_GEO")
```

```
# Process: Join Field
```

```
arcpy.JoinField_management(PRA_Methode_Clip_Area_shp, "FID", PRA_Methode_Referenz_Inters,  
"FID_PRA_Me", "POLY_AREA;AREA_GEO")
```



```

# Process: Add Field

arcpy.AddField_management(PRA_Methode_Clip_Area_shp, "OVERL_AREA", "FLOAT", "", "", "", "",
"NULLABLE", "NON_REQUIRED", "")

# Process: Calculate Field

arcpy.CalculateField_management(PRA_Methode_Clip_Area_shp, "OVERL_AREA", "[POLY_AREA_1] /
[POLY_AREA]*100", "VB", "")

# Process: Export Feature Attribute to ASCII

arcpy.ExportXYv_stats(PRA_Methode_Clip_Area_shp, "FID; POLY_AREA_1; POLY_AREA; OVERL_AREA",
"SEMI-COLON", PRA_csv, "ADD_FIELD_NAMES")

#####
# Intersection Referenz with PRA
#####

PRA_Methode_Diss = WorkDir_temp_valid + "PRA_Methode_Diss.shp"
PRA_Referenz_Methode_Inters = WorkDir_temp_valid + "PRA_Referenz_Methode_Inters.shp"
PRA_Referenz_Methode_Inters_Diss = WorkDir_temp_valid +
"PRA_Referenz_Methode_Inters_Diss.shp"
Referenz_csv = WorkDir + MethodRun + "_Referenz.csv"

# Process: Dissolve

arcpy.Dissolve_management(PRA_Methode_Clip_Area_shp, PRA_Methode_Diss, "", "", "MULTI_PART",
"DISSOLVE_LINES")

# Process: Add Geometry Attributes

arcpy.AddGeometryAttributes_management(PRA_Methode_Diss, "AREA;AREA_GEODESIC", "ME-
TERS", "SQUARE_METERS", "PRO-
JCS['CH1903_LV03',GEOGCS['GCS_CH1903',DATUM['D_CH1903',SPHEROID['Bessel_1841',6377397.1
55,299.1528128]],PRIMEM['Greenwich',0.0],UNIT['Degree',0.0174532925199433]],PROJECTION['H

```

```
otine_Oblique_Mercator_Azimuth_Center'],PARAMETER['False_Easting',600000.0],PARAMETER['False_Northing',200000.0],PARAMETER['Scale_Factor',1.0],PARAMETER['Azimuth',90.0],PARAMETER['Longitude_Of_Center',7.439583333333333],PARAMETER['Latitude_Of_Center',46.95240555555556],UNIT['Meter',1.0]]")
```

```
# Process: Intersect
```

```
arcpy.Intersect_analysis([PRA_Referenz_Clip_shp, PRA_Methode_Diss],  
PRA_Referenz_Methode_Inters, "ONLY_FID", "", "INPUT")
```

```
# Process: Dissolve
```

```
arcpy.Dissolve_management(PRA_Referenz_Methode_Inters, PRA_Referenz_Methode_Inters_Diss,  
'FID_PRA_Re', '#', 'MULTI_PART', 'DISSOLVE_LINES')
```

```
# Process: Add Geometry Attributes
```

```
arcpy.AddGeometryAttributes_management(PRA_Referenz_Methode_Inters_Diss, "AREA;AREA_GEODESIC", "METERS", "SQUARE_METERS", "PROJCS['CH1903_LV03',GEOGCS['GCS_CH1903',DATUM['D_CH1903',SPHEROID['Bessel_1841',6377397.155,299.1528128]],PRIMEM['Greenwich',0.0],UNIT['Degree',0.0174532925199433]],PROJECTION['Hotine_Oblique_Mercator_Azimuth_Center'],PARAMETER['False_Easting',600000.0],PARAMETER['False_Northing',200000.0],PARAMETER['Scale_Factor',1.0],PARAMETER['Azimuth',90.0],PARAMETER['Longitude_Of_Center',7.439583333333333],PARAMETER['Latitude_Of_Center',46.95240555555556],UNIT['Meter',1.0]]")
```

```
# Process: Add Geometry Attributes
```

```
arcpy.AddGeometryAttributes_management(PRA_Referenz_Clip_shp, "AREA;AREA_GEODESIC", "METERS", "SQUARE_METERS", "PROJCS['CH1903_LV03',GEOGCS['GCS_CH1903',DATUM['D_CH1903',SPHEROID['Bessel_1841',6377397.155,299.1528128]],PRIMEM['Greenwich',0.0],UNIT['Degree',0.0174532925199433]],PROJECTION['Hotine_Oblique_Mercator_Azimuth_Center'],PARAMETER['False_Easting',600000.0],PARAMETER['False_Northing',200000.0],PARAMETER['Scale_Factor',1.0],PARAMETER['Azimuth',90.0],PARAMETER['Longitude_Of_Center',7.439583333333333],PARAMETER['Latitude_Of_Center',46.95240555555556],UNIT['Meter',1.0]]")
```

```
# Process: Join Field
```

```

arcpy.JoinField_management(PRA_Referenz_Clip_shp, "FID", PRA_Referenz_Methode_Inters_Diss,
"FID_PRA_Re", "POLY_AREA;AREA_GEO")

# Process: Add Field

arcpy.AddField_management(PRA_Referenz_Clip_shp, "OVERL_AREA", "FLOAT", "", "", "", "", "NULL-
LABEL", "NON_REQUIRED", "")

# Process: Calculate Field

arcpy.CalculateField_management(PRA_Referenz_Clip_shp, "OVERL_AREA", "[POLY_ARE_1] /
[POLY_AREA]*100", "VB", "")

# Process: Export Feature Attribute to ASCII

arcpy.ExportXYv_stats(PRA_Referenz_Clip_shp, "FID; POLY_ARE_1; POLY_AREA; OVERL_AREA",
"SEMI-COLON", Referenz_csv, "ADD_FIELD_NAMES")

#####
# Process: Zonal Statistics DEM
#####
##PRA_Methode_NoSimplify_Clip_shp = WorkDir_temp_valid + "PRA_Methode_NoSimplify_Clip.shp"
##PRA_Methode_NoSimplify_Clip_Diss_shp = WorkDir_temp_valid +
"PRA_Methode_NoSimplify_Clip_Diss.shp"
PRA_Methode_Clip_Area_Diss_shp = WorkDir_temp_valid + "PRA_Methode_Clip_Area_Diss.shp"

##arcpy.Clip_analysis(PRA_Methode_Clip_Area_shp, Perimeter, PRA_Methode_Clip_Area_Diss_shp,
"")

list_fields = arcpy.ListFields(PRA_Methode_Clip_Area_shp, "Id")
if len(list_fields) == 0:
    arcpy.AddField_management(PRA_Methode_Clip_Area_shp, "Id", "LONG", "", "", "", "", "NULLA-
BLE", "NON_REQUIRED", "")
    arcpy.CalculateField_management(PRA_Methode_Clip_Area_shp, 'Id', '[FID]', 'VB', '#')

```

```

count = (arcpy.management.GetCount(PRA_Methode_Clip_Area_shp).getOutput(0))
if int(count) > 0:

    arcpy.Dissolve_management(PRA_Methode_Clip_Area_shp, PRA_Methode_Clip_Area_Diss_shp, '#',
'#', 'MULTI_PART', 'DISSOLVE_LINES')

    ##arcpy.DefineProjection_management(DEM, "PRO-
JCS['CH1903_LV03',GEOGCS['GCS_CH1903',DATUM['D_CH1903',SPHEROID['Bessel_1841',6377397.1
55,299.1528128]],PRIMEM['Greenwich',0.0],UNIT['Degree',0.0174532925199433]],PROJECTION['H
otine_Oblique_Mercator_Azimuth_Center'],PARAMETER['False_Easting',600000.0],PARAMETER['Fal
se_Northing',200000.0],PARAMETER['Scale_Factor',1.0],PARAMETER['Azimuth',90.0],PARAMETER['
Longi-
tude_Of_Center',7.439583333333333],PARAMETER['Latitude_Of_Center',46.95240555555556],UNI
T['Meter',1.0]]")

    ##arcpy.DefineProjection_management(PRA_Methode_NoSimplify, "PRO-
JCS['CH1903_LV03',GEOGCS['GCS_CH1903',DATUM['D_CH1903',SPHEROID['Bessel_1841',6377397.1
55,299.1528128]],PRIMEM['Greenwich',0.0],UNIT['Degree',0.0174532925199433]],PROJECTION['H
otine_Oblique_Mercator_Azimuth_Center'],PARAMETER['False_Easting',600000.0],PARAMETER['Fal
se_Northing',200000.0],PARAMETER['Scale_Factor',1.0],PARAMETER['Azimuth',90.0],PARAMETER['
Longi-
tude_Of_Center',7.439583333333333],PARAMETER['Latitude_Of_Center',46.95240555555556],UNI
T['Meter',1.0]]")

    arcpy.gp.ZonalStatisticsAsTable_sa(PRA_Methode_Clip_Area_Diss_shp, 'FID', DEM, Zonal-
Stat_SinglePoly_DEM, 'DATA', 'ALL')

    arcpy.TableToExcel_conversion(ZonalStat_SinglePoly_DEM, ZonalStat_OverallPoly_DEM, 'ALIAS',
'CODE')

#####

# Process: Zonal Statistics Slope

#####

if int(count) > 0:

    ##arcpy.DefineProjection_management(Slope, "PRO-
JCS['CH1903_LV03',GEOGCS['GCS_CH1903',DATUM['D_CH1903',SPHEROID['Bessel_1841',6377397.1

```

```
55,299.1528128]],PRIMEM['Greenwich',0.0],UNIT['Degree',0.0174532925199433]],PROJECTION['H  
otine_Oblique_Mercator_Azimuth_Center'],PARAMETER['False_Easting',600000.0],PARAMETER['Fal  
se_Northing',200000.0],PARAMETER['Scale_Factor',1.0],PARAMETER['Azimuth',90.0],PARAMETER['  
Longi-  
tude_Of_Center',7.439583333333333],PARAMETER['Latitude_Of_Center',46.95240555555556],UNI  
T['Meter',1.0]]")
```

```
arcpy.gp.ZonalStatisticsAsTable_sa(PRA_Methode_Clip_Area_Diss_shp, 'FID', Slope, Zonal-  
Stat_SinglePoly_Slope, 'DATA', 'ALL')
```

```
arcpy.TableToExcel_conversion(ZonalStat_SinglePoly_Slope, ZonalStat_OverallPoly_Slope, 'ALI-  
AS', 'CODE')
```

```
# Execute Delete of unwanted files
```

```
list = os.listdir(WorkDir)
```

```
for item in list:
```

```
    if item.endswith(".txt.xml"):
```

```
        os.remove(WorkDir + item)
```

```
for item in list:
```

```
    if item.endswith("temp.csv"):
```

```
        os.remove(WorkDir + item)
```

```
arcpy.env.workspace = WorkDir_temp_valid
```

```
rasters = arcpy.ListRasters("", "ALL")
```

```
for raster in rasters:
```

```
    arcpy.Delete_management(raster)
```

## 9.2 R-Skript

### 9.2.1 Validierung

```
#####  
# Datavalidation  
# Masterthesis  
# Daniel von Rickenbach  
#####  
# Directory which contain Runs  
DirMethod <- "OBIA_Parsenn_8"  
  
# Set Working Directory  
path <- paste("H:\\Masterarbeit\\Geo_Data_Prozessierung\\Resultate\\PRA\\PRA_Parsenn\\",  
DirMethod, sep="")  
#path <- pas-  
te("H:\\Masterarbeit\\Geo_Data_Prozessierung\\Resultate\\Autorel\\Autorel_Jaki_Riner\\", Dir-  
Method, sep="")  
#path <- paste("H:\\Masterarbeit\\Geo_Data_Prozessierung\\Resultate\\OBIA\\OBIA_Parsenn\\",  
DirMethod, sep="")  
#path <- pas-  
te("H:\\Masterarbeit\\Geo_Data_Prozessierung\\Resultate\\OBIA\\OBIA_Parsenn\\OBIA_Parsenn  
_10_ext\\OBIA_Parsenn_10_bestInputPara_2_eCog_extreme\\", DirMethod, sep="")  
#path <- pas-  
te("H:\\Masterarbeit\\Geo_Data_Prozessierung\\Resultate\\Slope\\Slope_Jaki_Riner\\", DirMe-  
thod, sep="")  
#path <- paste("C:\\Users\\vonricken\\Documents\\ArcGIS\\", DirMethod, sep="")  
  
#TotArea_Slope_30_60 <- 5582919.48853 #all  
TotArea_Slope_30_60 <- 2855634.881889 #Parsenn  
#TotArea_Slope_30_60 <- 1025296.287813 #Jakobshorn  
#TotArea_Slope_30_60 <- 1701988.318829 #Rinerhorn  
#TotArea_Slope_30_60 <- (1701988.318829 + 1025296.287813) #Jakobshorn + Rinerhorn  
  
setwd(path)  
  
# Create List_Folders of Directories which contain Runs  
list <- list.dirs(path, full.names = FALSE, recursive = FALSE)  
list_ind <- grep(DirMethod, list)  
list <- list[list_ind]  
OV_all_list <- list()  
  
#####
```

```

# Iterate over list

# Start with first Directory which contain Runs
#for (j in 1:length(list))
for (j in list) {

# Define Function for
F_Guetemassee <- function(Kl_PRA_NoPRA_eof, j, outp) {

# Read in Kl_PRA_NoPRA
Kl_PRA_NoPRA <- read.table(paste(j, "/", j, "_", Kl_PRA_NoPRA_eof, ".csv", sep=""), header=TRUE,
sep=";")

ID <- c(1,2,3,4)
ID_Test <- ID[which(!(ID%in%Kl_PRA_NoPRA$ID))]
if (length(ID_Test)>0) {
  for (element in ID_Test) {
    Kl_PRA_NoPRA_ext <- data.frame(matrix(NA, ncol=4, nrow=1))
    colnames(Kl_PRA_NoPRA_ext) <- colnames(Kl_PRA_NoPRA)
    Kl_PRA_NoPRA_ext$ID <- element
    Kl_PRA_NoPRA_ext$POLY_AREA <- 0
    Kl_PRA_NoPRA_ext$XCoord <- 0
    Kl_PRA_NoPRA_ext$YCoord <- 0
    Kl_PRA_NoPRA <- rbind(Kl_PRA_NoPRA, Kl_PRA_NoPRA_ext)
  }
}
Kl_PRA_NoPRA <- Kl_PRA_NoPRA[order(Kl_PRA_NoPRA$ID),]

rownames(Kl_PRA_NoPRA) <-
c("PRA_Referenz_PRA_Method", "NoPRA_Referenz_PRA_Method", "PRA_Referenz_NoPRA_Method", "N
oPRA_Referenz_NoPRA_Method")
Kl_PRA_NoPRA <- Kl_PRA_NoPRA [,c(3,4,1,2)]

Kl_PRA_standard <- read.table(paste(j, "/", j, "_", "Kl_PRA_NoPRA", ".csv", sep=""), header=TRUE,
sep=";")

ID <- c(1,2,3,4)
ID_Test <- ID[which(!(ID%in%Kl_PRA_standard$ID))]
if (length(ID_Test)>0) {
  for (element in ID_Test) {
    Kl_PRA_NoPRA_ext <- data.frame(matrix(NA, ncol=4, nrow=1))
    colnames(Kl_PRA_NoPRA_ext) <- colnames(Kl_PRA_standard)
    Kl_PRA_NoPRA_ext$ID <- element

```

```

Kl_PRA_NoPRA_ext$POLY_AREA <- 0
Kl_PRA_NoPRA_ext$XCoord <- 0
Kl_PRA_NoPRA_ext$YCoord <- 0
Kl_PRA_standard <- rbind(Kl_PRA_standard, Kl_PRA_NoPRA_ext)
}
}
Kl_PRA_standard <- Kl_PRA_standard[order(Kl_PRA_standard$ID),]

Kl_PRA_standard <- Kl_PRA_standard [c(3,4,1,2)]

# Create empty Error Matrix
F_matrix <- matrix(0,ncol=2,nrow=2)

# Fill up Error Marix
F_matrix[1,1] <- Kl_PRA_NoPRA[1,2]
F_matrix[1,2] <- Kl_PRA_NoPRA[2,2]
F_matrix[2,1] <- Kl_PRA_NoPRA[3,2]
F_matrix[2,2] <- Kl_PRA_NoPRA[4,2]

# Extend Error Marix
newcol <- 1:2
newrow <- 1:3
F_matrix <- cbind(F_matrix, newColumn = newcol)
F_matrix <- rbind(F_matrix, newRow = newrow)
rownames(F_matrix)<-c("PRA_Meth","NoPRA_Meth","S_Tot")
colnames(F_matrix)<-c("PRA_Ref","NoPRA_Ref","Z_Tot")

# Fill up Extented Error Marix
F_matrix[3,1] <- sum(F_matrix[1,1],F_matrix[2,1])
F_matrix[3,2] <- sum(F_matrix[1,2],F_matrix[2,2])
F_matrix[1,3] <- sum(F_matrix[1,1],F_matrix[1,2])
F_matrix[2,3] <- sum(F_matrix[2,1],F_matrix[2,2])
F_matrix[3,3] <- sum(F_matrix[3,1],F_matrix[3,2])
Tot_of_S_Tot <- sum(F_matrix[3,1],F_matrix[3,2])
Tot_of_Z_Tot <- sum(F_matrix[1,3],F_matrix[2,3])

if (Kl_PRA_NoPRA_eof=="Kl_PRA_NoPRA_W") {
  standard_fac <- (Kl_PRA_standard[1,2]+Kl_PRA_standard[3,2])/F_matrix[3,1]
  F_matrix[1,1] <- F_matrix[1,1]*standard_fac
  F_matrix[2,1] <- F_matrix[2,1]*standard_fac
  # Fill up Extented Error Marix
  F_matrix[3,1] <- sum(F_matrix[1,1],F_matrix[2,1])
  F_matrix[3,2] <- sum(F_matrix[1,2],F_matrix[2,2])
}

```



```

F_matrix[1,3] <- sum(F_matrix[1,1],F_matrix[1,2])
F_matrix[2,3] <- sum(F_matrix[2,1],F_matrix[2,2])
F_matrix[3,3] <- sum(F_matrix[3,1],F_matrix[3,2])
F_matrix <- round(F_matrix, 0)
Tot_of_S_Tot <- sum(F_matrix[3,1],F_matrix[3,2])
Tot_of_Z_Tot <- sum(F_matrix[1,3],F_matrix[2,3])

```

```
F_matrix[is.na(F_matrix)] <- 0
```

```
F_matrix <- round(F_matrix, 0)
```

```
# Create empty Accuracymatrix[Dez]
```

```
ncol=18
```

```
nrow=1
```

```
Guetemasse_Deiz <- matrix(0,ncol=ncol,nrow=nrow)
```

```
rownames(Guetemasse_Deiz)<-c(paste(j, sep=""))
```

```
if (Kl_PRA_NoPRA_eof=="Kl_PRA_NoPRA") {
```

```
  colnames(Guetemasse_Deiz) <- c("PA_PRA [%]", "FP_PRA [%]", "UA_PRA [%]", "ICA_PRA [%]", "OA
[%]", "ChanceAgreement [%]", "Kappa [%]", "DIF_PA_UA [%]", "MEAN_PA_UA [%]", "ZF_PA_UA [%]",
"MEAN_PA_UA_W [%]", "ZF_PA_UA_W [%]", "HSS [%]", "CSI [%]", "TSS [%]", "ETS [%]", "SEDS [%]",
"TSS_2FP [%]") }
```

```
if (Kl_PRA_NoPRA_eof=="Kl_PRA_NoPRA_W") {
```

```
  colnames(Guetemasse_Deiz) <- c("PA_PRA_W [%]", "FP_PRA_W [%]", "UA_PRA_W [%]", "ICA_PRA_W
[%]", "OA_W [%]", "ChanceAgreement_W [%]", "Kappa_W [%]", "DIF_PA_UA_W [%]",
"MEAN_PA_UA_W [%]", "ZF_PA_UA_W [%]", "MEAN_PA_UA_W_W [%]", "ZF_PA_UA_W_W [%]",
"HSS_W [%]", "CSI_W [%]", "TSS_W [%]", "ETS_W [%]", "SEDS_W [%]", "TSS_2FP_W [%]") }
```

```
# Fill up Accuracymatrix[Deiz]
```

```
Guetemasse_Deiz[1,1] <- F_matrix[1,1]/F_matrix[3,1]
```

```
Guetemasse_Deiz[1,2] <- F_matrix[1,2]/F_matrix[3,2]
```

```
Guetemasse_Deiz[1,3] <- F_matrix[1,1]/F_matrix[1,3]
```

```
Guetemasse_Deiz[1,4] <- F_matrix[1,1]/(sum(F_matrix[2,1],F_matrix[1,2]))
```

```
Guetemasse_Deiz[1,5] <- (sum(F_matrix[1,1],F_matrix[2,2]))/F_matrix[3,3]
```

```
Guetemasse_Deiz[1,6] <-
```

```
(F_matrix[3,1]*F_matrix[1,3]+F_matrix[3,2]*F_matrix[2,3])/(F_matrix[3,3]^2)
```

```
Guetemasse_Deiz[1,7] <- (Guetemasse_Deiz[1,5]-Guetemasse_Deiz[1,6])/(1-Guetemasse_Deiz[1,6])
```

```
Guetemasse_Deiz[1,8] <- abs(Guetemasse_Deiz[1,1] - Guetemasse_Deiz[1,3])
```

```
Guetemasse_Deiz[1,9] <- (Guetemasse_Deiz[1,1] + Guetemasse_Deiz[1,3])/2
```

```
Guetemasse_Deiz[1,10] <- Guetemasse_Deiz[1,9] - (Guetemasse_Deiz[1,8]/2)
```

```
Guetemasse_Deiz[1,11] <- (Guetemasse_Deiz[1,1]*3 + Guetemasse_Deiz[1,3])/4
```

```
Guetemasse_Deiz[1,12] <- Guetemasse_Deiz[1,11] - (Guetemasse_Deiz[1,8]/2)
```

```

Guetemassee_Dez[1,13] <- ((F_matrix[1,1]*F_matrix[2,2]-
F_matrix[1,2]*F_matrix[2,1])*2)/((F_matrix[1,1]+F_matrix[2,1])*(F_matrix[2,1]+F_matrix[2,2])+(F_
matrix[1,1]+F_matrix[1,2])*(F_matrix[1,2]+F_matrix[2,2]))
Guetemassee_Dez[1,14] <- F_matrix[1,1]/(F_matrix[1,1]+F_matrix[1,2]+F_matrix[2,1])
Guetemassee_Dez[1,15] <- (F_matrix[1,1]/(F_matrix[1,1]+F_matrix[2,1]))-
(F_matrix[1,2]/(F_matrix[1,2]+F_matrix[2,2]))
a_ref <- ((F_matrix[1,1]+F_matrix[1,2])*(F_matrix[1,1]+F_matrix[2,1]))/F_matrix[3,3]
Guetemassee_Dez[1,16] <- (F_matrix[1,1]-a_ref)/(F_matrix[1,1]-a_ref+F_matrix[1,2]+F_matrix[2,1])
Guetemassee_Dez[1,17] <-
(log((F_matrix[1,1]+F_matrix[1,2])/F_matrix[3,3])+log((F_matrix[1,1]+F_matrix[2,1])/F_matrix[3,3
]))/(log(F_matrix[1,1]/F_matrix[3,3]))
Guetemassee_Dez[1,17] <- Guetemassee_Dez[1,17]/2
Guetemassee_Dez[1,18] <- (F_matrix[1,1]/(F_matrix[1,1]+F_matrix[2,1]))-
((F_matrix[1,2]/(F_matrix[1,2]+F_matrix[2,2]))*2)

Guetemassee_Dez[is.na(Guetemassee_Dez)] <- 0

# Transform Accuracymatrix[Dez] to Accuracymatrix[Pzent]
matrix_Pzent <- matrix(100,ncol=ncol,nrow=nrow)
#Guetemassee <- data.frame(round(matrix_Pzent*Guetemassee_Dez,2), check.names = FALSE)
Guetemassee <- round(matrix_Pzent*Guetemassee_Dez,2)
#Guetemassee <- data.frame(Guetemassee, check.names = FALSE)

if (outp=="Kl_PRA_NoPRA") {
  return(Kl_PRA_NoPRA)}

if (outp=="F_matrix") {
  return(F_matrix)}

if (outp=="Guetemassee"){
  return(Guetemassee)}}

# Execute Function
Kl_PRA_NoPRA <- data.frame(F_Guetemassee("Kl_PRA_NoPRA", j, "Kl_PRA_NoPRA"))
F_matrix <- data.frame(F_Guetemassee("Kl_PRA_NoPRA", j, "F_matrix"))
Guetemassee <- data.frame(F_Guetemassee("Kl_PRA_NoPRA", j, "Guetemassee"), check.names = FALSE)

Kl_PRA_NoPRA_W <- data.frame(F_Guetemassee("Kl_PRA_NoPRA_W", j, "Kl_PRA_NoPRA"))
F_matrix_W <- data.frame(F_Guetemassee("Kl_PRA_NoPRA_W", j, "F_matrix"))
Guetemassee_W <- data.frame(F_Guetemassee("Kl_PRA_NoPRA_W", j, "Guetemassee"), check.names =
FALSE)

```

```

#####
# Read in and Manipulate Input Parameter
Input_Parameters <- read.table(paste(list(j), "/",list(j),"_Input_Parameters.csv", sep=""),
row.names = 1, header=TRUE, sep=";", check.names = FALSE)
names(Input_Parameters) <- gsub(x=names(Input_Parameters), pattern = "Deg", replacement = "°")

#####
# Read in and Manipulate Zonal Statistics
library(readxl)

if (file.exists(paste(list(j), "/",list(j),"_ZonalStat_OverallPoly_DEM.xls", sep = "")) == TRUE) {
  # ZonalStat_DEM <- read.table(paste(list(j), "/",list(j),"_ZonalStat_OverallPoly_DEM.csv", sep=""),
header=TRUE, sep=";", check.names = FALSE)
  ZonalStat_DEM <- read_excel(paste(j,"/",j,"_ZonalStat_OverallPoly_DEM.xls", sep = ""), range =
"C1:H2")
}

else {
  ZonalStat_DEM <- data.frame(matrix(NA, ncol=6, nrow=1))
  colnames(ZonalStat_DEM) <- c("COUNT", "AREA", "MIN", "MAX", "RANGE", "MEAN")
  ZonalStat_DEM$COUNT <- 0
}

if (file.exists(paste(list(j), "/",list(j),"_ZonalStat_OverallPoly_Slope.xls", sep="")) == TRUE) {
  # ZonalStat_Slope <- read.table(paste(list(j), "/",list(j),"_ZonalStat_OverallPoly_Slope.csv",
sep=""), header=TRUE, sep=";", check.names = FALSE)
  ZonalStat_Slope <- read_excel(paste(j,"/",j,"_ZonalStat_OverallPoly_Slope.xls", sep = ""), range =
"C1:H2")
}

else {
  ZonalStat_Slope <- data.frame(matrix(NA, ncol=6, nrow=1))
  colnames(ZonalStat_Slope) <- c("COUNT", "AREA", "MIN", "MAX", "RANGE", "MEAN")
  ZonalStat_Slope$COUNT <- 0
}

ZonalStat <- rbind(ZonalStat_DEM, ZonalStat_Slope)
# ZonalStat <- ZonalStat [,c(3,4,5,2)]
ZonalStat <- as.data.frame(ZonalStat)
rownames(ZonalStat) <- c("DEM [m.a.s.l.]", "Slope [°]")

#####

```

```

# Read in, Manipulate and plot AttrTab_PRA
Overlap <- data.frame(matrix(0,ncol=12,nrow=1))
rownames(Overlap) <- c(paste(j, sep=""))
colnames(Overlap) <- c("PRA_MeanOverlap [%]", "PRA_MinOverlap25 [%]", "PRA_MinOverlap50
[%]", "PRA_MinOverlap75 [%]", "Ref_MeanOverlap [%]", "Ref_MinOverlap25 [%]",
"Ref_MinOverlap50 [%]", "Ref_MinOverlap75 [%]", "PRA_TotArea [m2]", "TotArea_Slope_30_60
[m2]", "PRA_TotArea_Slope_30_60 [%]", "Referenz_TotArea [m2]")

AttrTab_PRA <- read.table(paste(list(j), "/",list(j), "_PRA.csv", sep=""), header=TRUE, sep=";",
check.names = FALSE)
Overlap[1,9] <- sum(AttrTab_PRA$POLY_AREA)
Overlap[1,10] <- TotArea_Slope_30_60
Overlap[1,11] <- Overlap[1,9]/Overlap[1,10]*100

if (length(AttrTab_PRA$OVERL_AREA) > 1) {
  AttrTab_PRA_sort <- AttrTab_PRA[order(AttrTab_PRA$OVERL_AREA),]
  MeanOverlap_PRA <- mean(AttrTab_PRA$OVERL_AREA)
  Overlap[1,1] <- MeanOverlap_PRA

  get_Pzent_of_Overlap <- function(Th){
    list = AttrTab_PRA_sort$OVERL_AREA
    index = 0
    for (item in list)
      if (item > Th)
        index = index + 1

    Pzent=index/length(list)*100
    return(Pzent)}

  Overlap[1,2] <- get_Pzent_of_Overlap(25)
  Overlap[1,3] <- get_Pzent_of_Overlap(50)
  Overlap[1,4] <- get_Pzent_of_Overlap(75)

  plot_area <- function(AttrTab_PRA_sort){
    xlab = "Index PRA"
    ylab = "Overlap [%]"
    library(areaplot)
    areaplot(AttrTab_PRA_sort$OVERL_AREA,
             xlab = xlab,
             ylab = ylab)}

  plot_area_grid <- function(AttrTab_PRA_sort){
    plot_area(AttrTab_PRA_sort)

```

```

grid()
par(new=TRUE)
plot_area(AttrTab_PRA_sort)}

png(paste(list(j), "//",list(j),"_PRA.png", sep=""))
plot_area_grid(AttrTab_PRA_sort)
dev.off()

win.metafile(paste(list(j), "//",list(j),"_PRA.wmf", sep=""))
plot_area_grid(AttrTab_PRA_sort)
dev.off()
}

else AttrTab_PRA_sort <- AttrTab_PRAAttrTab_PRA_sort <- AttrTab_PRAAttrTab_PRA_sort <- AttrT-
ab_PRA

# Read in, Manipulate and plot AttrTab_Referenz
AttrTab_Referenz <- read.table(paste(list(j), "//",list(j),"_Referenz.csv", sep=""), header=TRUE,
sep=";", check.names = FALSE)
Overlap[1,12] <- sum(AttrTab_Referenz$POLY_AREA)

if (length(AttrTab_Referenz$OVERL_AREA) > 1) {
  AttrTab_Referenz_sort <- AttrTab_Referenz[order(AttrTab_Referenz$OVERL_AREA),]
  MeanOverlap_Referenz <- mean(AttrTab_Referenz$OVERL_AREA)
  Overlap[1,5] <- MeanOverlap_Referenz

  get_Pzent_of_Overlap <- function(Th){
    list = AttrTab_Referenz_sort$OVERL_AREA
    index = 0
    for (item in list)
      if (item > Th)
        index = index + 1

    Pzent=index/length(list)*100
    return(Pzent)}

  Overlap[1,6] <- get_Pzent_of_Overlap(25)
  Overlap[1,7] <- get_Pzent_of_Overlap(50)
  Overlap[1,8] <- get_Pzent_of_Overlap(75)

  plot_area <- function(AttrTab_Referenz_sort){
    xlab = "Index Referenz"
    ylab = "Overlap [%]"

```

```

library(areaplot)
areaplot(AttrTab_Referenz_sort$OVERL_AREA,
         xlab = xlab,
         ylab = ylab)}

plot_area_grid <- function(AttrTab_Referenz_sort){
  plot_area(AttrTab_Referenz_sort)
  grid()
  par(new=TRUE)
  plot_area(AttrTab_Referenz_sort)}

png(paste(list(j), "//",list(j),"_Referenz.png", sep=""))
plot_area_grid(AttrTab_Referenz_sort)
dev.off()

win.metafile(paste(list(j), "//",list(j),"_Referenz.wmf", sep=""))
plot_area_grid(AttrTab_Referenz_sort)
dev.off()
}

else {
  AttrTab_PRA_sort <- AttrTab_PRA
}AttrTab_PRA_sort <- AttrTab_PRA

# png(paste(list(j), "//",list(j),"_hist_PRA_Referenz.png", sep=""))
# hist(AttrTab_Referenz_sort$POLY_ARE_1, plot = FALSE)
# hist(AttrTab_PRA_sort$POLY_AREA, add=TRUE)
# dev.off()

xrange_Referenz = range(AttrTab_Referenz_sort$POLY_AREA)
xrange_PRA = range(AttrTab_PRA_sort$POLY_AREA)
xrange_tot = range(xrange_Referenz, xrange_PRA)

# library(HistogramTools)
# png(paste(list(j), "//",list(j),"_hist_PRA_Referenz.png", sep=""))
# hist(AttrTab_Referenz_sort$POLY_AREA, xlim = xrange_tot, col = rgb(0,0,1,1/4), freq = FALSE)
# hist(AttrTab_PRA_sort$POLY_AREA, xlim = xrange_tot, col = rgb(1,0,0,1/4), freq = FALSE, add =
TRUE)
# dev.off()

library(HistogramTools)
png(paste(list(j), "//",list(j),"_hist_PRA_Referenz.png", sep=""))
Referenz <- hist(AttrTab_Referenz_sort$POLY_AREA, plot = FALSE)

```

```

PRA <- hist(AttrTab_PRA_sort$POLY_AREA, plot=FALSE)

# Referenz_BreakSize <- Referenz$breaks[2] - Referenz$breaks[1]
# PRA_BreakSize <- PRA$breaks[2] - PRA$breaks[1]
#
# if (Referenz_BreakSize > PRA_BreakSize) {
#   breaks <- c(Referenz$breaks)
# }
# else if (Referenz_BreakSize == PRA_BreakSize) {
#   breaks <- unique(c(Referenz$breaks, PRA$breaks))
# }
# else {
#   breaks <- c(PRA$breaks)
# }

# if (max(xrange_tot) > max(breaks)) {
#   xrange_tot <- xrange_tot
# }
# else {
#   xrange_tot <- range(breaks)
# }

xrange_Referenz = range(Referenz$breaks)
xrange_PRA = range(PRA$breaks)
xrange_tot <- range(xrange_Referenz, xrange_PRA)
breaks <- seq(min(xrange_tot), max(xrange_tot), length.out = 8)

Referenz <- hist(AttrTab_Referenz_sort$POLY_AREA, plot = FALSE, breaks = breaks)
PRA <- hist(AttrTab_PRA_sort$POLY_AREA, plot=FALSE, breaks = breaks)

# PlotRelativeFrequency(Referenz, xlab="area [m2]", xlim = xrange_tot, ylim = c(0,1), col =
rgb(0,0,1,1/4), density = 2)
PlotRelativeFrequency(Referenz, xlab="area [m2]", xlim = xrange_tot, ylim = c(0,1), col =
rgb(0,0,1,1/4), xaxt="n")
PlotRelativeFrequency(PRA, xlab="area [m2]", xlim = xrange_tot, ylim = c(0,1), col = rgb(1,0,0,1/4),
xaxt="n", add = TRUE)

PlotRelativeFrequency(Referenz, xlab="area [m2]", xlim = xrange_tot, ylim = c(0,1), density = 5,
angle = 45, xaxt="n", add = TRUE)
PlotRelativeFrequency(PRA, xlab="area [m2]", xlim = xrange_tot, ylim = c(0,1), density = 5, angle =
135, xaxt="n", add = TRUE)

axis(1, at = breaks)

```

```

dev.off()

#####
# For Run[j]

Sys.setenv(JAVA_HOME='C:/Program Files/Java/jre-9')
library(rJava)
library(xlsx)

# Create Workbook
outwb_1 <- createWorkbook()

add_datafr_sheet_to_wb <- function(outwb_1, datafr) {
  sheetName = as.character(substitute(datafr))
  sheet_1 <- createSheet(outwb_1, sheetName)
  addDataFrame(datafr, sheet_1)
  #return(sheet_1)
  #return(outwb_1)
}

add_manipulate_datafr_sheet_to_wb <- function(outwb_1, datafr, j) {
  sheetName = as.character(substitute(datafr))
  sheet_1 <- createSheet(outwb_1, sheetName)
  addDataFrame(datafr, sheet_1)
  # add_datafr_sheet_to_wb(outwb_1, datafr)
  rows <- getRows(sheet_1)
  cells <- getCells(rows, colIndex = 1)
  setCellValue(cells[[1]], j)
  #return(sheet_1)
  #return(outwb_1)
}

add_datafr_sheet_to_wb(outwb_1, Input_Parameters)
add_datafr_sheet_to_wb(outwb_1, Guetemassee)
add_datafr_sheet_to_wb(outwb_1, Guetemassee_W)
add_datafr_sheet_to_wb(outwb_1, Overlap)
add_manipulate_datafr_sheet_to_wb(outwb_1, Kl_PRA_NoPRA, j)
add_manipulate_datafr_sheet_to_wb(outwb_1, Kl_PRA_NoPRA_W, j)
add_manipulate_datafr_sheet_to_wb(outwb_1, F_matrix, j)
add_manipulate_datafr_sheet_to_wb(outwb_1, F_matrix_W, j)
add_manipulate_datafr_sheet_to_wb(outwb_1, ZonalStat, j)

saveWorkbook(outwb_1, paste(list(j), "/", list(j), "_Ueberblick.xlsx", sep=""))

```



```

# Add OV_run to OV_all
OV_run <- cbind(Input_Parameters, Guetemasse, Guetemasse_W, Overlap)
OV_all_list[[j]] <- OV_run
}

#####
# For all Runs
OV_all = do.call(rbind, OV_all_list)
OV_all <- OV_all[order(OV_all$'TSS [%]', decreasing = TRUE),]

# Export into xlsx.Workbook
outwb <- createWorkbook()
sheetName = DirMethod
sheet <- createSheet(outwb, sheetName)
addDataFrame(OV_all, sheet)
addAutoFilter(sheet, "A:BZ")
saveWorkbook(outwb, paste(DirMethod, "_OV_all.xlsx", sep = ""))

```

## Persönliche Erklärung

Ich erkläre hiermit, dass ich die vorliegende Arbeit selbständig verfasst und die den verwendeten Quellen wörtliche oder inhaltlich entnommene Stellen als solche kenntlich gemacht habe.

Davos, 12.09.2018

D. von Rickenbach

---

Ort, Datum

---

Name