

LEOPOLD-FRANZENS-UNIVERSITÄT INNSBRUCK

INSTITUT FÜR GEOGRAPHIE

# **Konzeption und Implementierung eines Open Source Workflows zur Verwaltung und Verteilung von LiDAR-Daten**

## **Bachelorarbeit**

**Klaus Förster**

Matrikelnummer: 8315834

Studienkennzahl: C 033 655

Betreuer: Dr. Martin Rutzinger

Innsbruck, 15. Jänner 2016

# Inhaltsverzeichnis

1 Einleitung.....	3
2 Stand der Forschung.....	5
3 Zielsetzung.....	9
4 Methoden.....	10
4.1 Datenaufbereitung und Ablage.....	10
4.2 Ableitung und Speicherung von Metadaten in der Datenbank.....	12
4.3 Web-Interface für Metadaten und Rohdaten.....	15
5 Technische Infrastruktur.....	16
6 Implementierung des Workflows.....	16
6.1 Migration der Rohdaten in eine einheitliche Struktur.....	16
6.1.1 Bereinigen der Rohdaten.....	16
6.1.2 Konvertieren der Punktwolken in das LAS-Format.....	17
6.2 Ableitung von Metadaten.....	19
6.3 Import der Metadaten in die PostgreSQL-Datenbank.....	21
6.3.1 Zentrale Metadatentabelle meta.....	21
6.3.2 Datenbankview für Abfragen view_meta.....	22
6.3.3 Datenimport der Metadaten in die PostgreSQL-Datenbank.....	23
6.4 Web-Interface für Metadaten und Rohdaten.....	24
6.4.1 Kartenhintergrund über Web Map Tile Services.....	25
6.4.2 Interaktiver Vektorlayer & Datendownload.....	26
7 Ergebnisse und Fazit.....	28
8 Ausblick.....	29
9 Zusammenfassung.....	30
10 Summary.....	32
11 Literaturverzeichnis.....	33
12 Anhang.....	39

## Tabellenverzeichnis

Tabelle 1: Dateigröße der unsortierten Rohdatenverzeichnisse (Stand Jänner 2016).....	5
Tabelle 2: LAS Punktdaten-Format 1.....	10
Tabelle 3: Verzeichnisstruktur der Rohdatenablage.....	12
Tabelle 4: Struktur des Metadatenverzeichnisses einer Befliegung.....	13
Tabelle 5: Datenbestand der LAS-Dateien nach Bereinigung der Daten.....	14

## Abbildungsverzeichnis

Abbildung 1: Screenshot des Karteninterfaces.....	25
Abbildung 2: Einzelne Kacheln der WMTS-Hintergrundlayer im Karteninterface.....	26

## Listings

Listing 1: Shell-Befehle zur Migration der Rohdaten in die neue Dateistruktur.....	16
Listing 2: Bereinigung der X-Kordinaten bei TopScan-Daten.....	17
Listing 3: Shell-Befehle zur Ableitung der generalisierten konkaven Hülle der Punktwolke.....	20
Listing 4: SQL-Befehl zur Erzeugung der Metadatentabelle meta.....	21
Listing 5: SQL-Befehl zur Erzeugung des Datenbankviews view_meta.....	22
Listing 6: SQL-Befehle zum Einfügen der Metadaten in die Tabelle meta.....	24
Listing 7: SQL-Befehl zum Filtern der LAS-Dateien in einem Rechtecksausschnitt.....	27
Listing 8: SQL-Befehl zur Ermittlung der Rechtecks-Koordinaten in der Projektion der LAS-Datei.....	28
Listing 9: las2las-Befehl zur Filterung der Laserpunkte in einem Rechteck.....	28

# 1 Einleitung

Airborne Laserscanning (ALS) hat sich zu einem leistungsfähigen, operationell einsetzbaren Standardverfahren in den Naturwissenschaften entwickelt (Geist et al. 2004). Sowohl staatliche als auch akademische Institutionen erfassen LiDAR-Daten (Light Detection and Ranging) im großen Stil zur Erstellung hochauflösender Geländemodelle und zum Monitoring von Naturraumprozessen (Krishnan 2011, Martinez-Rubi 2015, Rieg et al. 2014). Die dabei entstehenden, räumlich und zeitlich immer höher aufgelösten Datensätze eröffnen "neue Möglichkeiten in der Auswertung, Analyse und Anwendung" (Rutzinger et al. 2013, S. 209), bedeuten aber auch gleichzeitig große Herausforderungen beim effizienten Datenhandling der riesigen Datenmengen zur Sicherung der langfristigen Verfügbarkeit (Boehm & Liu 2015, Cura et al. 2015, Rieg et al. 2014).

Am Institut für Geographie der Leopold-Franzens-Universität Innsbruck besteht eine lange Tradition der Forschung mit LiDAR-Daten. Die erste Befliegung zur Erfassung von ALS-Daten liegt beinahe fünfzehn Jahre zurück und fand am 11. Oktober 2001 am Hintereisferner im Hinteren Ötztal statt (Sailer et al. 2010b). Damit erfolgte der Startschuss für den Aufbau einer weltweit einzigartigen, multitemporalen ALS-Daten Serie (Geist et al. 2004, Geist et al. 2005a, Sailer et al. 2012, Sailer 2013, Rieg et al. 2014) mit jährlichen Befliegungen zum Ende des hydrologischen Jahres im Herbst bis in das Jahr 2013, ergänzt durch einzelne Befliegungen im Frühjahr und Sommer in den Jahren 2002 bis 2003 (Rieg 2014 et al.).

Im Fokus der Erforschung der bis dato 23 Befliegungen umfassenden Serie standen neben der Evaluierung der Eignung von ALS als Instrument zum Monitoring von Gletschern (Geist et al. 2005a, S.195) auch die Ableitung geodätischer Massenbilanzen (Rieg et al. 2014, S.141), die multitemporale Analyse der Flächen- und Volumsänderung (Sailer et al. 2010b, S. 180; Geist et al. 2005a, S. 195) sowie die Klassifizierung der Gletscheroberfläche über Intensitätswerte (Sailer et al. 2010b, S. 180) als Basis für eine bessere Einschätzung der Dichtewerte von Schnee, Eis und Firn bei der Bestimmung geodätischer Massenbilanzen (Sailer et al. 2010b, S. 183).

Die Finanzierung der Befliegungen erfolgte über öffentliche Gelder im Rahmen von Projekten wie dem EU-Projekt OMEGA (Operational Monitoring of European Glacial Areas) (2001 bis 2003), Mitteln aus dem Tiroler Wissenschaftsfonds (TFW) (2004 bis 2005), Eigenmitteln des Instituts für Geographie der Leopold-Franzens-Universität Innsbruck (2006, 2013), dem Austrian Space Applications Programm (ASAP) (2007 bis 2009), dem Austrian Climate Research Programme (ACRP) (2009), Mitteln des alpS-Centre for Climate Change Adaption Technologies (2010, 2013), dem Projekt Climate Change Consequences for the Cryosphere (C4AUSTRIA) des ACRP (2010 bis 2011) (Sailer et al. 2010b, S. 176) sowie dem Projekt Multiscale Snow/Ice Melt Discharge Simulations for Alpine Reservoirs (MUSICALS), das von der Tiroler Wasserkraft AG (TIWAG) und der Österreichischen Forschungsförderungsgesellschaft (FFG) finanziert wurde (Helfricht et al. 2015, S. 734).

Die Ausweitung der Forschungsfragen auf geomorphologische Fragestellungen brachte auch eine Erweiterung der Projektgebiete mit sich. Neben dem Bereich Hintereisferner standen im C4AUSTRIA-Projekt damit auch ALS-Datensätze im Bereich Ötztal (Inneres Reichenkar, Inneres- und Äußeres Hochebenkar), im Stubaital (Schränkar) und im Bereich Kaunergrat (Innere Ölgrube) (Sailer 2013, S.7) im Mittelpunkt der Forschung zur Entwicklung, Ableitung und Erweiterung von "Airborne Laserscanning (ALS) basierten Methoden und Workflows zum Monitoring und zur Analyse von Oberflächenänderungen in Gletscher-, Toteis- und Permafrostgebieten" (Sailer 2013, S. 2). Aufbauend auf diesen Ergebnissen entstand im Rahmen des permAfrost-Projekts der Austrian Permafrost Research Initiative auf Basis von ALS-Daten ein Blockgletscherinventar für das Montafon in Vorarlberg (Klug et al. 2014, S. 67) mit Befliegung im Jahr 2010 und der Nachbefliegung einzelner, wetterbedingt fehlender Flugstreifen im Jahr 2011 (Klug et al. 2014, S. 55).

Mit dem MALS-Projekt (Multitemporal Airborne Laserscanning Südtirol) zur Quantifizierung der Veränderung von Blockgletschern (Autonome Provinz Bozen – Südtirol 2009) und dem ebenfalls von der Autonomen Provinz Bozen – Südtirol geförderten Projekt "A physically based regional mass balance approach for the glaciers of the Vinschgau catchment – glacier contribution to water availability" standen schließlich auch die Gletscher südlich des Alpenhauptkamms im Mittelpunkt der Forschung. Am Beispiel von knapp 100 Gletschern wurde dabei die Gletscheränderung von 2005 bis 2013 sowie deren Beitrag zum Wasserhaushalt der Flüsse im Südtiroler Vinschgau untersucht (Galos et al. 2015).

Der im Rahmen dieser Forschungsprojekte entstandene, umfangreiche Pool an ALS-Datensätzen umfasst aber nicht nur ALS-Punktwolken im Alpenbereich, sondern auch ALS-Daten aus Skandinavien, die in der Anfangszeit des Laserscannings am Institut für Geographie der Leopold-Franzens-Universität Innsbruck entstanden sind. Vier Befliegungen am Engabreen, einem Teil des Svartisen Gletschers in Norwegen, für die Zeitpunkte September 2001, Mai 2002, August 2002 und im Juni 2003 als Teil des EU-Projekts OMEGA zur Evaluierung der Eignung von ALS als Instrument zum Monitoring von Gletschern sind hier zu nennen (Geist et al. 2005a, S.195).

Tabelle 1 zeigt mit Stand Jänner 2016 den Umfang der im Rahmen dieser Arbeit verwendeten Rohdatensätze. Sie listet neben einem Kürzel, das im weiteren Verlauf der Arbeit zur Kennzeichnung der einzelnen Datenpools verwendet wird, auch den Gebietsstand der Daten sowie den belegten Speicherplatz in Gigabyte (GB) beziehungsweise Terrabyte (TB) auf.

Mit dem neuesten Datenzugang vom Amt der Vorarlberger Landesregierung in Form eines 914 GB großen Laserhöhenmodells Vorarlbergs aus dem Jahr 2011, beläuft sich das Gesamtvolumen der unsortierten Rohdaten damit auf zirka 3,9 TB. Eine Analyse des Forschungsstands aktueller Ansätze zur Verwaltung und Verteilung von ALS-Punktwolken soll dazu beitragen, einen geeigneten Workflow zur Neuorganisation der ALS-Daten zu definieren, implementieren und für Projekte am Institut für Geographie der Leopold-Franzens-Universität Innsbruck zugänglich zu machen. Bei der Analyse des Forschungsstands wird

auf das Funktionsprinzip des Laserscannings nicht näher eingegangen. Der Leser, die Leserin sei an dieser Stelle auf die Arbeiten von Vosselman & Mass (2010) und Shan & Toth (2008) verwiesen.

*Tabelle 1: Dateigröße der unsortierten Rohdatenverzeichnisse (Stand Jänner 2016)*

<b>Kürzel</b>	<b>Gebiet</b>	<b>Dateigröße</b>
c4austria	Öztaler Alpen	12 GB
engabreen	Engabreen Gletscher (Norwegen)	27 GB
hef	Hintereisferner	421 GB
montafon	Montafon (Vorarlberg)	389 GB
musicals	Kaunertal / Öztaler Alpen	2.0 TB
vinschgau	Ortler Gebiet / Südliche Öztaler Alpen (Schnalstal)	129 GB
vogis	Vorarlberg	914 GB
<b>Summe</b>		<b>~ 3,9 TB</b>

*Quelle: eigene Tabelle*

## 2 Stand der Forschung

Erste Ansätze zur Verwaltung und Verteilung von LiDAR-Daten mit freier Software gehen zurück auf die Diplomarbeit von Höfle am Institut für Geographie der Leopold-Franzens-Universität Innsbruck (Höfle 2005). Die im Rahmen des alpS-Forschungsprojekts "Analyse von Laserscannerdaten im Hinblick auf die Bestimmung von Oberflächeneigenschaften" durchgeführte Arbeit präsentiert am Beispiel eines 500 mal 500 m großen Testdatensatzes im Vorarlberger Rheintal bei Hohenems das "Konzept eines Systems für die Verwaltung, Verarbeitung und Visualisierung dieser Daten" (Höfle 2005, S. II). Kernstück der Datenhaltung ist eine PostgreSQL 7.4.5 Datenbank (PostgreSQL Global Development Group 2015) mit dem PostGIS-Modul in Version 0.9 (PostGIS Project Steering Committee 2015a) zur Speicherung der Attribut- und Geometriedaten für jeden Punkt der Punktwolke. Zur Verarbeitung und Analyse der Daten kommen das geographische Informationssystem GRASS 5.7 (Geographic Resources Analysis Support System) (GRASS Development Team 2002) sowie das Statistikpaket R 2.0.0 (The R Foundation 2015) zum Einsatz. Die Visualisierung erfolgt über das Python Modul Tkinter in Version 8.4 (Python Software Foundation 2015) (Höfle 2005, S. 54).

Die Weiterentwicklung dieses unter der Bezeichnung "LISA" (LIDAR Surface Analyses) (Höfle et al. 2006) vorgestellten Prototypen mündete später in das kommerzielle Laserdaten Informationssystem "LIS" der Firma Laserdata GmbH (Laserdata GmbH 2015), einem Spin-Off des Instituts für Geographie der Leopold-Franzens-Universität Innsbruck (Rieg et al. 2014, S. 139). Die Komponenten zur Verarbeitung, Analyse und Visualisierung der Daten (GRASS, R und Tkinter) wurden dabei durch das Opensource GIS

SAGA (System for Automated Geoscientific Analyses) (Conrad et al. 2015) ersetzt und durch zahlreiche Module zur Filterung und Analyse von Punktwolken erweitert (Petrini-Monteferri et al. 2009, S. 623).

Das von der National Science Foundation (NFS) geförderte und am San Diego Supercomputer Center (SDSC) beheimatete OpenTopography Projekt (OpenTopography Facility 2015a) verfolgt das Ziel der Schaffung einer Service-orientierten "Geospatial Infrastructure (GCI)" (Krishnan et al. 2011, S. 1) zur webbasierten Verteilung und Analyse von LiDAR-Daten. Kernstück ist ein Geo-Processing-Service, der die Nutzung der Rechenleistung des San Diego Supercomputer Centers zur Erzeugung von Derivaten aus LiDAR-Daten wie Raster für Hangneigung, Relief oder hydrologische Modelle im GeoTIFF oder ERDAS Imagine (HFA) Format erlaubt (Krishnan et al. 2011, S. 5).

Beim Download der LiDAR-Rohdaten gilt für anonyme Benutzer und Benutzerinnen eine Punktebeschränkung auf 50 Millionen Punkte, nach Registrierung kann dieses Limit auf 250 Millionen Punkte pro Download erhöht werden. Darüber hinaus können on-the-fly Analysen beliebiger Rechtecksausschnitte in Auftrag gegeben werden, deren Status im so genannten "MyOpenTopo" Interface (OpenTopography Facility 2015c) verfolgt werden kann. Alternativ dazu besteht, auch für anonyme Benutzer und Benutzerinnen, die Möglichkeit sich via E-Mail über den Abschluss eines Auftrags informieren zu lassen. Aufträge lassen sich im MyOpenTopo Interface speichern und zu einem späteren Zeitpunkt auch mit anderen Parametern oder Ausschnitten wiederholt ausführen.

Als Datenbank *back end* kommt für Punktdaten einerseits das relationale Datenbankmanagementsystem DB2 von IBM zum Einsatz, andererseits werden statische LAS-Dateien auch über Metadaten-Tabellen in der DB2 Datenbank verwaltet. Räumliche Abfragen auf die Punkte, die in der Datenbank liegen, finden dabei direkt in der Datenbank statt, Abfragen auf die statischen LAS-Dateien erfolgen zweistufig: Zuerst werden über das Hüllrechteck der Abfrage die betroffenen LAS-Dateien ermittelt und dann die entsprechenden Daten aus den LAS-Dateien heraus gefiltert (Krishnan et al. 2011, S. 4).

Mit Stand Dezember 2015 finden sich im LiDAR-Datenbereich damit 181 ALS- und 7 TLS-Datensätze, wobei der Fokus auf tektonisch aktiven Zonen Nordamerikas sowie hydrologischen Daten amerikanischer Behörden liegt. Zusammen mit 82 vorgefertigten Rasterdatensätzen beläuft sich das Datenvolumen auf mehr als 23 TB (OpenTopography Facility 2015b). Darüber hinaus bietet OpenTopography noch weltweite Datensätze der Shuttle Radar Topography Missionen SRTM GL1 (Global 30m) und SRTM GL3 (Global 90m), die automatisiert über eine Webschnittstelle heruntergeladen werden können (OpenTopography Facility 2015d).

Im Gegensatz zum OpenTopography Projekt am San Diego Supercomputer Center verfolgt die "LiDAR Data Management Pipeline" von Lewis, Mc Elhinney & McCarthy an der National University of Ireland (Lewis et al. 2012) den Ansatz, Punktdaten ausschließlich in einer Datenbank zu speichern. Als Datenbank *back end* dient dabei eine PostgreSQL-Datenbank in Kombination mit dem PostGIS-Modul, das das Handling der Punktgeometrie und räumliche Abfragen übernimmt. Neben ALS-Daten werden auch TLS- und MLS-Daten

(Mobile Laser Scanning) in die Datenbank importiert, mit räumlichen Indizes und Attribut-Indizes zur Beschleunigung der Abfragen versehen und über ein Web-Interface ausgeliefert. Mittels WebGL (Web Graphics Library), einem freien Webstandard zur hardwarebeschleunigten Anzeige von 3D-Grafiken (Khronos Group, 2014), können die Punktwolken dann direkt im Browser angezeigt werden. Sowohl Intensitätswerte als auch RGB-Werte von Punkten können so farblich dargestellt werden (Lewis et al. 2012, S.7).

Zur Visualisierung des Datenbestands im Web-Interface dienen konkave Hüllen, die über die `ST_ConcaveHull`-Funktion des PostGIS-Moduls (PostGIS Project Steering Committee 2015b) in der Datenbank nach Ausdünnung der Punktwolke berechnet und in verschiedenen Generalisierungsstufen im Interface zur Verwendung kommen. Damit kann der Datenbestand unabhängig von den Originalpunkten auf lokaler, regionaler und nationaler Ebene angezeigt werden. (Lewis et al. 2012, S. 6).

Paul Ramsey, der Hauptentwickler des PostGIS-Moduls, geht in einem von Natural Resources Canada geförderten Projekt einen Schritt weiter als Lewis et al. und erweitert die PostgreSQL-Datenbank kurzerhand um ein neues Modul mit dem Namen "Pointcloud" (Ramsey 2015). Er erkennt, dass die Speicherung von Punkten als PostGIS Punktgeometrien einen großen Datenoverhead mit sich bringt, der zur massiven Erhöhung des benötigten Speicherplatzes in der Datenbank führt (Ramsey 2013).

Seine Lösung für dieses Problem ist die Einführung eines neuen PostgreSQL-Datentyps mit dem Namen *patch*, der sich aus mehreren 100 Punkten (*points*) zusammensetzt. Jeder dieser Punkte kann beliebige Attribute wie X-, Y-, Z-Koordinaten, Intensität, etc. besitzen. Art und Reihenfolge dieser Attribute werden in einem XML Schema Dokument zentral abgelegt und nur ein Verweis auf dieses Dokument (*pcid*) gemeinsam mit dem Patch abgespeichert. Damit ist sicher gestellt, dass Attribute von Punkten innerhalb eines Patches richtig interpretiert werden können (ibid.).

Das Laden von Punktwolken im LAS-Format in die PostgreSQL Pointcloud erfolgt mittels PDAL, der "Point Data Abstraction Library" (Butler & Gerlek 2015). Beim Import werden die Punktwolkendaten in überschneidungsfreie, rechteckige Patches aufgesplittet, was am Beispiel der 420 MB großen LAS-Datei des Mt. St. Helens mit 12 Millionen Punkten, 30971 Patches mit einer maximalen Punktzahl von 400 ergibt. In Verbindung mit den räumlichen Abfragemöglichkeiten von PostGIS und zusätzlichen Datenbankfunktionen für das Pointcloud-Modul können die Punkte der Patches schließlich nach räumlicher Lage, Höhe oder anderen Attributen abgefragt und das Resultat, ähnlich wie beim Import, über PDAL wieder als LAS-Datei gespeichert werden.

Eine praktische Anwendung des Pointcloud-Moduls von Paul Ramsey zeigen Cura, Perret & Paparoditis von der Université Paris-Est mit ihrem "Point Cloud Server (PCS)" (Cura et al. 2015). Sie nützen die ganze Bandbreite der PostgreSQL - PostGIS - Pointcloud Architektur zur Ablage, Analyse, Filterung und Ausgabe von ALS-, TLS- und Photogrammetrischen-Punktwolken samt deren Metadaten und erweiterten Metadaten wie z.B. Flugtrajektorien bei ALS-Daten. Umfangreiche Benchmarks beim Laden, Komprimieren, Indizieren



und Filtern der Daten zeigen, dass der Flaschenhals bei diesen Operationen nicht beim Arbeitsspeicher des Servers, sondern meist bei der Lese / Schreibgeschwindigkeit der Festplatte (I/O) zu finden ist (Cura et al. 2015, S. 535).

Einen dateizentrierten Ansatz in Verbindung mit einer NoSQL-Datenbank verfolgen Boehm & Liu (Boehm & Liu 2015, S. 577). Am Beispiel des frei verfügbaren Laserscanning-Höhenmodells der Niederlande AHN2 (Actueel Hoogtbestand Nederland) zeigen sie Möglichkeiten auf, gekachelte Punktwolkendaten samt deren Metadaten in einer Mongo-Datenbank (MongoDB Inc. 2015a) zu verwalten. Dazu werden die Hüllrechtecke der Kacheln in geographische Koordinaten (WGS84) umprojiziert und mit zusätzlichen Attributen wie Aufnahmezeitpunkt oder Name der Originaldatei in einem MongoDB-Dokument gespeichert (Boehn & Liu 2015, S. 578). Die Punktdaten der Kacheln selbst liegen ursprünglich im LAS-Format vor und werden beim Import in einem sogenannten "GridFS" (MongoDB Inc. 2015b), einer Emulation eines verteilten Dateisystems, das die Mongo-Datenbank zur Verfügung stellt, abgelegt. GridFS splittet dabei die Originaldateien in maximal 255 Kilobyte (KB) große, binäre Blöcke auf, die mit ihren Metadaten über eine gemeinsame ID verknüpft sind. Damit steht für jede Punktwolkenkachel ein Dokument mit Metadaten und einem Hüllrechteck in WGS84-Projektion sowie den Punktdaten im GridFS zur Verfügung. Räumliche Abfragen werden dann über die Hüllrechtecke der Kacheln realisiert und liefern als Resultat die betroffenen Punktwolkenkacheln zurück. Benchmarks zeigen, dass die Ablage der Punktwolken in der Datenbank Geschwindigkeitseinbußen bei der Ausgabe gegenüber dem einfachen Kopieren im Dateisystem mit sich bringt (Boehn & Liu 2015, S. 581).

Einen wertvollen Beitrag zur aktuellen Forschung im Bereich webbasierter Visualisierung und Verteilung von Punktwolkendaten liefert das "Massive Point Clouds for eSciences" Projekt (Netherlands eScience Center 2015). Neben Grundlagenarbeit wie der Erhebung von Nutzer-Anforderungen beim Arbeiten mit Punktwolken (Suijker et al. 2014) oder dem Benchmarking verschiedener datenbank- und dateibasierter Ansätze zur Verwaltung von Punktwolken mit PostgreSQL, Oracle, MonetDB und den LAAstools (Martinez-Rubi et al. 2014, Van Oosterom et al. 2015) ist die Entwicklung einer webbasierten, interaktiven Applikation zur Visualisierung von Punktwolken erklärtes Ziel des Projekts (Martinez-Rubi et al. 2015, S. 1). Im Mittelpunkt steht dabei das als Open Data Datensatz verfügbare und zum Monitoring der Deiche, für 3D-Stadtmodelle oder Klimasimulationen eingesetzte AHN2 Laserscanning-Modell der Niederlande mit 640 Milliarden Punkten in 1.6 TB komprimierten LAS-Dateien. (Martinez-Rubi et al. 2015, S. 1).

Die interaktive Visualisierung dieser Punktwolkendaten wird im Browser mittels WebGL realisiert. Dabei kommt "Potree" (Schütz 2015) zum Einsatz, ein frei verfügbarer, WebGL basierter Viewer für Punktwolken. Für das schnelle Laden im Browser müssen die Punktwolkendaten bei Potree in einer so genannten "Octree-Baumstruktur" (Martinez-Rubi et al. 2015, S. 3) vorliegen. Ausgehend von einem Mindestabstand zwischen zwei Punkten im Wurzelknoten der dreidimensionalen Octree-Baumstruktur, wird dabei mit jedem neuen Level des Octrees der geforderte Mindestabstand halbiert und nur jene Punkte in diesen Level aufgenommen, die den Mindestabstand erfüllen. Die Erzeugung dieser Octree-Baumstruktur ist sehr aufwendig und würde

im Falle der AHN2 Punktwolke mit 640 Milliarden Punkten, die auf 14 Levels mit über 38 Millionen Knoten aufgeteilt werden müssen (Martinez-Rubi et al. 2015, S. 7), rund 100 Tage in Anspruch nehmen. Durch Entwicklung eines eigenen Algorithmus zur Beschleunigung dieses Erstellungsvorgangs konnte die Prozessierungszeit auf 10 Tage verringert werden (Martinez-Rubi et al. 2015, S. 6).

Das interaktive Web-Interface (Actueel Hoogtebestand Nederland 2015) erlaubt neben der Anzeige der 3D-Punktwolke in vier verschiedenen Qualitätsstufen auch das Messen von Distanzen, Winkeln, Flächen, Volumina und Höhen sowie die Einfärbung der Punkte nach Höhe, Intensität, Klassifikation oder weiteren Attributen. Eine Ortssuche sowie die Anzeige des Blickfelds in einer 2D Orientierungskarte, die auch zur Bestimmung von Rechtecksausschnitten beim Datendownload verwendet werden kann, ergänzen das eindrucksvolle Web-Interface (Martinez-Rubi et al. 2015, S. 7).

### 3 Zielsetzung

Wenngleich die direkte Darstellung der Punktwolken im Browser mittels WebGL wie beim AHN2 *pointcloud viewer* (Actueel Hoogtebestand Nederland 2015) wünschenswert erscheint, ist sie zur Verwaltung und Verteilung von heterogenen Punktwolkendaten aus verschiedenen Quellen mit unterschiedlichen Aufnahmezeitpunkten derzeit leider nicht geeignet. Deshalb orientiert sich der im Rahmen der vorliegenden Arbeit entwickelte Workflow zur Verwaltung und Verteilung der LiDAR-Daten am Institut für Geographie der Leopold-Franzens-Universität Innsbruck eher am Ansatz von Boehm & Liu. Als Datenbank *back end* kommt allerdings eine PostgreSQL-Datenbank mit PostGIS anstelle der Mongo-Datenbank zum Einsatz, und die Punktwolkendaten werden nicht in der Datenbank abgelegt, sondern im Dateisystem belassen. Die Vorteile, die sich daraus ergeben, betreffen neben einfacherem Handling der Geometriedaten über das PostGIS-Modul auch die platzsparende Speicherung der Laserdaten, die nicht in ein weiteres System kopiert werden müssen. Zentrales Dateiformat für die Ablage der Rohdaten wird ähnlich wie bei OpenTopography oder dem Massive Point Clouds for eSciences Projekt das LAS-Format sein. Für die Visualisierung des Datenbestands, die Anzeige von Metadaten und die Implementierung des Datendownloads über ein Web-Interface zeigt OpenTopography geeignete Möglichkeiten auf, die in abgewandelter Form im eigenen Workflow zur Anwendung kommen werden.

Die Motivation diesen Workflow gänzlich mit freier Software zu implementieren begründet sich vor allem darin, dass die Server-Infrastruktur am Institut für Geographie der Leopold-Franzens-Universität Innsbruck seit jeher auf Open Source Komponenten setzt. Diesem Open Source Gedanken folgend werden auch alle Skripts und HTML-Seiten, die im Rahmen der Implementierung der einzelnen Phasen des Workflows zur Verwaltung und Verteilung der LiDAR-Daten erstellt werden, in einem öffentlichen Quellcode-Repository unter der Internetadresse <https://github.com/openwebcc/ba> abgelegt (Förster 2015). Die Web-Applikation selbst ist unter der Adresse <http://geographie.uibk.ac.at/lidar/app.py> im Internet verfügbar.

## 4 Methoden

### 4.1 Datenaufbereitung und Ablage

Am Beginn der Datenaufbereitung steht die Festlegung des primären Zielformats, in das die Rohdaten vor der Ablage im Dateisystem überführt werden müssen. Dieses Zielformat soll standardisiert und frei verfügbar sein und die Weiterverarbeitung mit Hilfe freier Software ermöglichen. Für LiDAR Punktwolken bietet sich das LAS-Format an, ein von der American Society for Photogrammetry & Remote Sensing (ASPRS) geschaffener offener Standard zum Austausch von dreidimensionalen Laserscanning Punktwolkendaten. Der LAS-Standard wurde im Jahr 2000 von der ASPRS ins Leben gerufen (Samberg 2007, S.365) und liegt derzeit in Version 1.4 (R13) vor (ASPRS 2013). Eine binäre LAS-Datei besteht dabei aus einem Header für Metadaten der Punktwolke (ASPRS 2013, S. 5), gefolgt von optionalen Variable Length Records (VLRs) (ASPRS 2013, S. 8) und einer beliebigen Zahl an Einträgen für Punkte der Punktwolke, die in einem von zehn verschiedenen Punktformaten vorliegen können (ASPRS 2013, S.12 ff.).

Im Zuge der Aufbereitung der Rohdaten werden im Header jeder LAS-Datei Attribute wie der Name des verwendeten Sensors, das Erstellungsdatum mit Angabe von Jahr und der Nummer des Tages innerhalb eines Jahres, sowie Minima, Maxima, Skalierungsfaktoren und Offsets von X-, Y-, Z-Werten der Koordinaten, Projektion, das Punktformat, die Anzahl der Punkte sowie die Verteilung der einzelnen Returns gespeichert. Ein vollständige Liste aller gültigen Header-Attribute findet sich in der LAS-Spezifikation in der Tabelle "Public Header Block" (ASPRS 2013, S.5).

*Tabelle 2: LAS Punktdaten-Format 1*

Attribut	Inhalt
X, Y, Z	Koordinatenwerte als Ganzzahlen. Die Kommastellen werden über Skalierungsfaktoren im Header definiert, Offsets im Header müssen berücksichtigt werden
Intensity	Intensität (Magnitude) des reflektierten Signals
Return Number	Nummer des Returns im Wertebereich 1 bis 15
Number of Returns (given pulse)	Anzahl der Returns, die für ein Signal registriert wurden
Scan Direction Flag	Bewegungsrichtung des Scannerspiegels beim Aussenden des Signals: 1 = von links nach rechts, 2 = Bewegung von rechts nach links
Edge of Flight Line	Lage des Punktes im Bezug zur Scanlinie: 1 = liegt am Ende, 0 = Punkt liegt nicht am Ende der Scanlinie

Classification	Optionale Klassifizierung eines Punktes. Einige Standardklassen dabei sind:  2 - Bodenpunkt 3 - niedere Vegetation 4 - mittelhohe Vegetation 5 - hohe Vegetation 6 - Gebäude 9 - Wasser  Weitere benutzerdefinierte Klassifikationen können folgen
Scan Angle Rank (-90 to +90) – Left side	Scannerwinkel beim Aussenden des Signals in Grad. Negative Werte bedeuten "in Flugrichtung links", positive Werte "in Flugrichtung rechts"
User Data	Beliebige Zahl zur freien Verfügung
Point Source ID	Aus welcher Datei stammt dieser Punkt
GPS Time	GPS-Zeitstempel des Laserpulses

*Quelle: modifiziert nach ASPRS 2013, S. 10-12*

Die Punkte selbst lassen sich, je nach Art der Punktwolke, in eines von zehn Punktformaten einteilen. Bei Flugstreifen mit X-, Y-, Z-Koordinaten, GPS-Zeitstempeln (Global Positioning System), Intensitätswerten und mehreren Returns pro Laserpuls kommt damit das Punktdaten-Format 1 zum Einsatz (siehe Tabelle 2).

Fehlen GPS-Zeitstempel, wie dies bei gekachelten Daten meist der Fall ist, wird das Punktdaten-Format 0 verwendet (ASPRS 2013, S.10), das sich von Format 1 eben nur dadurch unterscheidet, dass kein Attribut für GPS-Zeitstempel vorhanden ist. Der GPS-Zeitstempel eines Punktes wird in Millisekunden bezogen auf die so genannte "GPS Week Time" (Global Positioning Systems Directorate 2013, S.50) angegeben. Nachdem die GPS-Time jeden Sonntag um Mitternacht auf Null zurückgesetzt wird, ist bei Flugstreifendaten die Eindeutigkeit der Zuordnung von Punkten zu den Metadaten der dazugehörigen Flugtrajektorie nur dann gegeben, wenn auch die GPS-Woche als Referenz bekannt ist. Daraus ergibt sich die Forderung, dass bei der Ablage von Rohdaten mit GPS-Zeitstempeln auch das Erstellungsdatum mit Angabe von Jahr und der Nummer des Tages innerhalb eines Jahres im Header der LAS-Datei gespeichert werden muss.

Neben der Eindeutigkeit der GPS-Zeitstempel müssen alle bereinigten Rohdaten auch eindeutige Angaben zur Projektion der Punktwolke beinhalten. Dazu bietet sich die Ausweisung des so genannten EPSG-Codes (European Petroleum Survey Group Geodesy) als VLR in der LAS-Datei an. Die European Petroleum Survey Group Geodesy bezeichnet jene Institution, die seit Mitte der 80er-Jahre die Standardisierung von Koordinatensystemen und -transformationen vorantrieb und heute beim Geomatics Committee der International Association of Oil and Gas Producers (IOGP) beheimatet ist (IOGP 2015a). Im online verfügbaren EPSG Geodetic Parameter Dataset (IOGP 2015b) sind die standardisierten EPSG-Codes definiert.

Die Vollständigkeit bei Rohdaten von Flugstreifen ist erst dann gegeben, wenn auch die "Abweichung des Flugzeugs bzw. der Lasereinheit vom definierten Raumvektor der geplanten Flugbahn" (Rutzinger et al. 2013, S. 213) aufgezeichnet wird. Für jeden ausgesandten Laserimpuls werden so GPS-Zeitstempel, X-, Y- und Z-Koordinate der Flugzeugposition und die Winkel für das Rollen (*roll*), Nicken (*pitch*) und Driften (*heading*) des Flugzeugs sowie die Abweichung der Flugbahn von Nord in Grad (*wander angle*) (Höfle 2005, S. 26) in der Flugtrajektorie gespeichert. Nachträgliche Korrekturen der Intensitäten zum Ausgleich der Reliefenergie in gebirgigem Terrain sind nur bei Kenntnis dieser Werte möglich (Höfle & Pfeifer 2007, S. 431). Die Daten der Flugtrajektorie müssen daher bei der Ablage der Rohdaten im Dateisystem berücksichtigt werden.

Die Verzeichnisstruktur der Rohdatenablage folgt einem einheitlichen Schema, in das die bereinigten Rohdaten der Projekte und Befliegungen eingeordnet werden können. Neben der Speicherung der LAS-Dateien bietet es Platz für allenfalls vorhandene Flugtrajektorien im Textformat sowie für eine Dokumentation in Form von Befliegungsberichten mit Angaben zur Flugplanung, den Witterungsverhältnissen während der Befliegung, der Flugstreifenübersicht sowie der Genauigkeit der GPS-Messungen (TopScan 2003). Tabelle 3 zeigt diese Verzeichnisstruktur.

*Tabelle 3: Verzeichnisstruktur der Rohdatenablage*

<b>Verzeichnis</b>	<b>Inhalt</b>
als/	Stammverzeichnis für ALS-Daten
als/projekt/	Projektordner
als/projekt/011011_befliegung/	Befliegungsordner mit Zeitstempel und Kürzel
als/projekt/011011_befliegung/bet/	Unterordner für Flugtrajektorien, sofern vorhanden
als/projekt/011011_befliegung/doc/	Unterordner für Befliegungsberichte und Dokumentation
als/projekt/011011_befliegung/las/	Unterordner für LAS-Dateien

*Quelle: eigene Tabelle*

## **4.2 Ableitung und Speicherung von Metadaten in der Datenbank**

Aus den einheitlich abgelegten Rohdaten gilt es in einem nächsten Schritt Metadaten und generalisierte Geometrien abzuleiten, die später für die Visualisierung im Web-Interface verwendet werden können. Im Konkreten handelt es sich dabei um Metadatenblätter im Textformat für jede Punktwolke, Polygone mit der konkaven Hülle der zweidimensionalen Abdeckung der Punktwolke und generalisierte Linienzüge der Trajektorien sowie die ungefilterten Trajektoriendaten jedes Flugstreifens, sofern vorhanden.

Für die Ablage dieser Metadaten wird die Verzeichnisstruktur der Befliegungen (siehe Tabelle 3) um ein weiteres Unterverzeichnis *meta* mit folgenden Inhalten (siehe Tabelle 4) für jede LAS-Datei ergänzt:

Tabelle 4: Struktur des Metadatenverzeichnisses einer Befliegung

Verzeichnis / Datei	Inhalt
als/projekt/011011_befliegung/meta/	Unterordner für Metadaten
als/projekt/011011_befliegung/meta/*.info.txt	Metadatenblätter als Ergebnis von lasinfo
als/projekt/011011_befliegung/meta/*.hull.wkt	generalisierte konkave Hüllen als WKT-Geometrie
als/projekt/011011_befliegung/meta/*.traj.wkt	generalisierte Flugtrajektorien als WKT-Geometrie
als/projekt/011011_befliegung/meta/*.traj.txt	originale Flugtrajektorien der Flugstreifen

Quelle: eigene Tabelle

Mittel der Wahl bei der Ableitung der Metadatenblätter ist `lasinfo`, ein Programm aus der LAStools Softwaresammlung zur Prozessierung von Punktwolken im LAS-Format von Martin Isenburg (Isenburg 2015a). Mittels `lasinfo` erzeugte Textdateien beinhalten nicht nur alle Attribute, die der LAS-Standard im Header einer LAS-Datei vorsieht (ASPRS 2013, S.5), sondern auch Zusatzinformationen wie Projektion der Koordinaten, Minima und Maxima der Attribute des vorliegenden Punktdaten-Formats, Zusatzstatistiken über die Verteilung der Returns, die abgedeckte Fläche, mittlere Punktdichte und mittlere Punktabstände sowie bei klassifizierten Daten ein Histogramm der einzelnen Klassen.

Die Ableitung einer generalisierten konkaven Hülle der Punktwolke erfolgt ebenfalls mit Hilfe der LAStools in einem zweistufigen Prozess durch Ausdünnen der Punktwolke mit dem Programm `lasprecision` und anschließender Berechnung der konkaven Hülle durch `lasboundary`. Das Resultat wird als so genanntes WKT-Polygon gespeichert. "WKT" steht dabei für "Well Known Text", einen Standard des Open Geospatial Consortiums (OGC) zur Ausweisung von Punkt-, Linien- und Polygoneometrien in einem einheitlich definierten Textformat (Open Geospatial Consortium 2010a, S.51).

Für Punktwolken mit Trajektoriendaten werden mit Hilfe eines eigenen Python Skripts zusätzlich Textdateien mit den für den jeweilige Flugstreifen relevanten Trajektorieneinträgen sowie generalisierte Linienzüge der Trajektorien der einzelnen Flugstreifen im WKT-Format erstellt. Dabei wird zuerst für jede LAS-Datei das Zeitfenster der Befliegung aus den Minimum- und Maximumwerten der GPS Zeitstempel im Metadatenblatt ermittelt und dann, in einem zweiten Schritt, die Trajektorie der Befliegung eingelesen und jene Punkte als Originaltrajektorie des Flugstreifens herausgefiltert, die in das passende Zeitfenster fallen. Für die generalisierte Trajektorie als WKT-Geometrie werden nur jene Punkte in eine generalisierte Sub-Trajektorie übernommen, deren Punktabstand mindestens 100 Meter beträgt.

Damit sind im Metadatenverzeichnis `meta` - mit Ausnahme der Originaltrajektorien der Flugstreifen, die direkt für den Datendownload verwendet werden - alle Metadaten verfügbar und können in ein Datenbankschema übersetzt und in die Datenbank importiert werden. Als Datenbank *back end* für das Web-Interface kommt eine PostgreSQL 9.3 Datenbank mit dem PostGIS 2.1. Modul zum Einsatz. Beide Komponenten sind

als Open-Source-Software frei verfügbar und ermöglichen neben den klassischen objekt-relationalen Datenbankabfragen auch räumliche Abfrage und Transformationen geometrischer Daten direkt in der Datenbank über das PostGIS-Modul.

Das Datenbankschema besteht aus einer zentralen Metadatentabelle, in der die Metadaten der Dateiablage abgebildet werden können, und einem Datenbankview als Basis für Abfragen aus dem Web-Interface zur Erstellung von Karten, Verzeichnissen und Datenblättern. Die zentrale Metadaten-Tabelle enthält dabei für jede LAS-Datei zum einen alle Attribute, die direkt aus der Dateistruktur der Datenablage abgeleitet werden können wie Projekttyp, Projektname, Name und Datum der Befliegung sowie Name und Größe der LAS-Datei. Zum anderen bietet sie Platz für die generalisierten Geometrien der konkaven Hüllen und Flugtrajektorien sowie alle Metadaten, die über lasinfo generiert wurden.

Diese Geometrien werden in der zentralen Metadatentabelle allerdings vorerst ohne dezidiert ausgewiesene Projektion gespeichert, denn Geometrien innerhalb einer Tabelle dürfen nur eine einzige Projektion aufweisen, was bei der Vielzahl an Befliegungen, die in der zentralen Metadatentabelle abgebildet werden müssen, von Haus aus nicht gegeben ist. Die Umwandlung zu "echten" und damit räumlich abfragbaren Geometrien erfolgt erst im Datenbankview, der als Grundlage für Abfragen im Web-Interface dient.

Für den Import der Metadaten in die PostgreSQL-Datenbank kommt ein eigenes Pythonmodul zum Einsatz, das aus den Metadatenblättern und WKT-Geometrien für konkave Hülle und Flugtrajektorie der einzelnen Flugstreifen im Metadatenverzeichnis der Befliegungen die gewünschten Daten extrahiert und in das Datenbankschema überführt.

*Tabelle 5: Datenbestand der LAS-Dateien nach Bereinigung der Daten*

<b>Projekt</b>	<b>Befliegungen</b>	<b>Dateien</b>	<b>Megabyte</b>	<b>Punkte</b>
c4austria	3	31	5.244	189.310.519
engabreen	4	271	3.405	170.244.143
hef	21	644	67.856	2.472.970.313
montafon	2	53	24.701	882.168.296
musicals	2	444	179.527	6.411.660.609
vinschgau	5	200	55.958	2.122.884.145
vogis	1	544	255.139	66.495.689.913
<b>Summe</b>	<b>38</b>	<b>2.187</b>	<b>591.829</b>	<b>78.744.927.938</b>

*Quelle: eigene Tabelle*

Tabelle 5 fasst die Situation nach Bereinigung und Import der Metadaten in die Datenbank zusammen und listet für jedes Projekt die wesentlichen Kennzahlen wie Anzahl der Befliegungen, Zahl der LAS-Dateien im

Dateisystem mit Größe sowie die Summe der erfassten Laserpunkte auf.

### 4.3 Web-Interface für Metadaten und Rohdaten

Der webbasierte Zugang zur Anzeige der Metadaten aus der Datenbank und dem Download der Rohdaten der einzelnen Befliegungen aus dem Dateisystem wird mit Hilfe des HTML5 Standards (Hypertext Markup Language) (WHATWG 2015) und der Programmiersprache Javascript (Ecma International 2015) über eine Pythonschnittstelle realisiert. Serverseitig kommt dabei ein Apache2 Webserver (Apache Software Foundation 2015) zum Einsatz, der über ein Pythonmodul als Schnittstelle zur Datenbank und zur Generierung des HTML5-Quelltextes die entsprechenden Webseiten über das HTTP-Protokoll (Hypertext Transfer Protocol) ausliefert. Zur Anzeige der Webseiten beim Benutzer, der Benutzerin können gängige Browser wie Firefox, Chrome, Safari oder der Internet Explorer verwendet werden.

Als Einstieg dient dabei eine tabellarische Startseite mit den wichtigsten Kennzahlen jeder Befliegung wie Projektname, Zeitpunkt und Name der Befliegung, Anzahl und Größe der LAS-Dateien, Punktzahl, durchschnittliche Punktdichte, EPSG-Code der Projektion oder verwendeter Sensor. Über Hyperlinks erfolgt von dort aus der Zugang zu den Details der einzelnen Befliegungen sowie zu weiterführenden Informationen zur Projektion der Punktwolke über die Webseite <http://spatialreference.org>, einer Online Datenbank, die parallel zum EPSG Geodetic Parameter Dataset (IOGP 2015b) eine umfassende Übersicht der vorhandenen EPSG-Codes enthält und zusätzlich die Ausgabe der jeweiligen Projektions- und Transformationsparameter in der Notation gängiger GIS-Programme erlaubt (Butler et al. 2015).

Das Karteninterface der jeweiligen Befliegung besteht aus frei wählbaren Hintergrundkarten zur Orientierung und einem interaktiven Vektorlayer, der über die generalisierten Geometrien der Flugtrajektorien beziehungsweise konkaven Hüllen der Punktwolken den Zugang zu Metadaten und den Datendownload ermöglicht. Bei der Implementierung des Kartenhintergrunds kommt der Web Map Tile Service (WMTS) Standard des Open Geospatial Consortiums zum Einsatz (Open Geospatial Consortium 2010b ). Er definiert ein einheitliches System zur Kachelung rasterbasierter Karten in verschiedene Zoomstufen, die dann über das Web effizient geladen werden können. Beliebige Raster-Hintergrundkarten freier Anbieter wie [openstreetmap.org](http://openstreetmap.org) (OpenStreetMap contributors 2015), [basemap.at](http://basemap.at) (Stadt Wien et al. 2015), der Autonomen Provinz Bozen – Südtirol (Autonome Provinz Bozen 2015) oder [kartverket.no](http://kartverket.no) (Norwegian Mapping Authority 2015) lassen sich so, je nach geographischer Lage der Befliegung, über ein Layer-Menü auswählen.

Das Karteninterface ermöglicht den direkten Zugang zu den Metadaten der einzelnen Punktwolken und bietet Downloadmöglichkeiten für die originalen LAS-Dateien, Trajektoriendaten der Flugstreifen und dem Befliegungsbericht, sofern vorhanden. Über ein Freihandtool können durch Aufziehen beliebiger Rechtecke entweder alle Flugstreifen / Kacheln die das Rechteck schneiden, oder nur jene Punkte heruntergeladen werden, die innerhalb des Rechtecks liegen. Das Klippen der betroffenen originalen LAS-Dateien am gewählten Rechteck übernimmt dabei das Programm `las2las` der LAStools.



## 5 Technische Infrastruktur

Als technische Infrastruktur zur Implementierung des Workflows kommt ein Fujitsu RX2530 M1 Rack-Server mit zwei Intel Xeon E5-2620v3 Prozessoren, 12 Kernen, 24 CPUs und 32 GB RAM mit Ubuntu 14.04 LTS (Long Term Support) als Betriebssystem zum Einsatz. Als Speichermedium für die Daten dient ein NetApp FAS2200-Server mit 22 TB Speicherplatz, der mit dem Ubuntu-Server über das Samba-Protokoll (Samba Team 2015) und iSCSI (Internet Small Computer System Interface) für schnellen Datenzugriff verbunden ist. Die neue Datenablage verwendet dabei die iSCSI-Schnittstelle, die unsortierten Rohdaten und der Arbeitsbereich der LiDAR Research Group am Institut für Geographie der Leopold-Franzens-Universität Innsbruck sind über das Samba-Protokoll als Verzeichnisse eingebunden. Nachdem die Web-Applikation in der derzeitigen Phase noch nicht öffentlich zugänglich ist, werden auch alle Download-Produkte in einem Downloadverzeichnis am Server abgelegt, zu dem nur berechtigte Mitarbeiterinnen und Mitarbeiter von Forschungsprojekten am Institut Zugang haben.

## 6 Implementierung des Workflows

### 6.1 Migration der Rohdaten in eine einheitliche Struktur

#### 6.1.1 Bereinigen der Rohdaten

Die Identifizierung der Originaldateien für Punktwolken in den unsortierten Rohdatenverzeichnissen der jeweiligen Projekte (siehe Tabelle 1) steht am Beginn der Datenmigration. Gesucht wird dabei nach Daten für Flugstreifen oder Kacheln, die meist in Form von ASCII-Textdateien, in seltenen Fällen wie beim musicals- oder vogis-Datenpool auch als LAS-Dateien beziehungsweise komprimierte LAS-Dateien vorliegen. Bei Unklarheiten des Datenbestands erfolgt ein Abgleich mit den, z.B. von der Firma TopScan - Gesellschaft zur Erfassung topographischer Information mbH als Hauptauftragnehmerin der meisten Befliegungen, gelieferten originalen CDs, DVDs und Festplatten. Im Idealfall können so für jede Befliegung die Flugstreifendaten samt dazugehöriger Trajektorie sowie ein Befliegungsbericht identifiziert und mit Hilfe von Shell-Skripts automatisiert in die neue Rohdatenstruktur (siehe Tabelle 3) überführt werden. Listing 1 zeigt am Beispiel der ersten Hintereisferner-Befliegung vom 11.10.2001 die nötigen Unix Shell-Befehle zur Erzeugung der neuen Rohdatenstruktur und zur Migration der Daten aus dem unsortierten Rohdatenbereich in die neue Dateistruktur.

*Listing 1: Shell-Befehle zur Migration der Rohdaten in die neue Dateistruktur*

```
# Ordnerstruktur erstellen
mkdir -pv /home/laser/rawdata/als/hef/011011_hef01/{asc,las,bet,doc,meta}

# Rohdaten und Dokumentation kopieren
cp -avu /mnt/netappa/Rohdaten/hef/hef01_011011/str/ala/*.ala ./asc/
cp -avu /mnt/netappa/Rohdaten/hef/hef01_011011/str/H_011011.bet ./bet/
cp -avu /mnt/netappa/Rohdaten/hef/*Befliegungsbericht*.pdf ./doc/report.pdf
```

Quelle: [https://github.com/openwebcc/ba/blob/master/maintenance/rawdata/als/hef/011011\\_hef01.sh](https://github.com/openwebcc/ba/blob/master/maintenance/rawdata/als/hef/011011_hef01.sh)

In einem nächsten Schritt müssen die Koordinaten aller von der Firma TopScan gelieferten UTM-Flugstreifen, -Trajektorien und -Kacheln bereinigt werden, denn sie liegen in einer benutzerdefinierten Projektion vor, die sich von der offiziellen EPSG-Projektion nur dadurch unterscheidet, dass bei X-Koordinaten die Streifennummer der UTM-Projektion vorangestellt ist. Diese Bereinigung wird mit Hilfe der Skriptsprache awk (Aho 1987) in der Kommandozeile über Unix Shell-Befehle durchgeführt. Listing 2 zeigt am Beispiel des ersten Flugstreifens der Hintereisbefliegung vom 11.10.2001 die dabei nötigen Shell-Befehle sowie die ersten drei Zeilen des ASCII-Flugstreifens vor und nach der Bereinigung.

#### Listing 2: Bereinigung der X-Kordinaten bei TopScan-Daten

```
# die ersten drei Zeilen vor der Bereinigung
380600.732777 32640438.47 5187492.83 2276.91 6 1 2
380600.732777 32640438.48 5187492.82 2276.72 6 2 2
380600.756418 32640426.93 5187503.81 2285.40 7 1 2

# Bereinigung der x-Koordinaten durch Entfernen von 32 in Spalte zwei
awk '{gsub(/^32/, "", $2); print}' H_011011_01.ala > H_011011_01.ala.tmp
mv H_011011_01.ala.tmp H_011011_01.ala

# die ersten drei Zeilen nach der Bereinigung
380600.732777 640438.47 5187492.83 2276.91 6 1 2
380600.732777 640438.48 5187492.82 2276.72 6 2 2
380600.756418 640426.93 5187503.81 2285.40 7 1 2
```

Quelle [https://github.com/openwebcc/ba/blob/master/maintenance/rawdata/als/hef/011011\\_hef01.sh](https://github.com/openwebcc/ba/blob/master/maintenance/rawdata/als/hef/011011_hef01.sh)

Damit kann, anstelle der Benutzer-definierten Projektion SR-ORG:6715 - ETRS89 / UTM zone 32N with leading 32, nun die offizielle, von allen gängigen GIS-Programmen unterstützte Projektion EPSG:25832 - ETRS89 / UTM zone 32N verwendet werden (Butler & Gerlek 2015).

Bei den Hintereisferner-Befliegungen der Jahre 2004, 2005 und 2006 tritt die Besonderheit auf, dass die Punktwolkendaten in jeweils zwei, nach *first* und *last* Echo, getrennten Dateien mit den Dateiendungen *.alf* (mnemotechnisch für "airborne laser first return") und *.all* (mnemotechnisch für "airborne laser last return") geliefert wurden. Mit Hilfe eines eigenen Shell- und Pythonskripts werden diese beiden Dateien zu einer Datei zusammengespielt und dabei Spalten für die Nummer des Echos und die Gesamtzahl der Echos hinzugefügt. Duplizierte Einträge werden entfernt und mit einem Single-Return belegt. Das Resultat entspricht dem Format anderer Flugstreifen und wird analog zu diesen als Datei mit der Endung *.ala* (mnemotechnisch für "airorne laser all returns") gespeichert.

### 6.1.2 Konvertieren der Punktwolken in das LAS-Format

Zum Erzeugen von und Arbeiten mit Dateien, die dem LAS-Standard entsprechen, steht mit den LAsTools von Martin Isenburg eine umfangreiche Softwaresammlung zur Verfügung. Die Bibliothek gliedert sich in freie, unter der LGPL 2.1 (Free Software Foundation 1999) verfügbare Programme sowie Programme, die

eine Lizenzierung erfordern. Ausgenommen vom Lizenzierungszwang der zweiten Kategorie sind Nutzungen für den persönlichen non-profit Bereich, nicht militärische Schulungszwecke und non-profit humanitäre Projekte (Isenburg 2015b).

Zum Konvertieren der gesäuberten Rohdaten im ASCII-Textformat in das LAS-Format kommt das Programm `txt2las` zum Einsatz. Es fällt in die erste Kategorie der freien Programme und verfügt über eine Reihe an Kommandozeilen-Parametern zur Steuerung des Konvertierungsvorgangs. Am Beispiel des ersten Flugstreifens der Hintereisferner Befliegung vom 11.10.2001 lautet der Skriptaufruf zur Umwandlung folgendermaßen:

```
|txt2las -i H_011011_01.ala -o ../las/H_011011_01.ala -parse txyzirn
```

Der Parameter `-i` bestimmt dabei den Namen der ASCII-Inputdatei, `-o` legt den Pfad der LAS-Outputdatei in einem parallelen Unterverzeichnis `las` fest (siehe Tabelle 3), und `-parse` definiert, in welcher Reihenfolge die Attribute in der ASCII-Inputdatei stehen. Ein Blick in die erste Zeile von `H_011011_01.ala` erlaubt die bessere Zuordnung der Zeichenkette `"txyzirn"` im Parameter `-parse` zu den Spalten der Inputdatei.

380600.732777	640438.47	5187492.83	2276.91	6	1	2
t	x	y	z	i	r	n

Der Buchstabe `t` bezeichnet dabei den GPS-Zeitstempel im Millisekunden der GPS-Week, `x` und `y` entsprechen den Koordinaten, `z` legt die Höhe fest, `i` enthält Intensität des Echos, `r` die Nummer des Echos und `n` die Gesamtzahl an Echos des ausgesendeten Lasersignals. Gleichzeitig können über den `txt2las`-Parameter auch weitere Einträge im Header der LAS-Datei direkt gesetzt werden:

```
| -epsg 25832 -set_file_creation 284 2001 -set_system_identifizier "ALTM 1225"
```

Der Skriptparameter `-epsg` führt zur Zuweisung des EPSG-Codes 25832 - ETRS89 / UTM zone 32N als Variable Length Record (ASPRS 2013, S.8). Das Befliegungsdatum mit Nummer des Tages im Jahr (284) und Jahr (2001) wird über `-set_file_creation` in den Header-Attributen "File Creation Day of Year" und "File Creation Year" abgelegt, der Name des verwendeten Sensors "ALTM 1225" über den Parameter `-set_system_identifizier` im Attribut "System Identifier" gespeichert (ASPRS 2013, S.5). Bei der Konvertierung konnten diese drei Parameter für alle Rohdaten aus den Befliegungsberichten bzw. aus wissenschaftlichen Artikeln der betroffenen Projekte abgeleitet werden. Zusätzlich werden für alle LAS-Dateien die Offsets der X-, Y-, Z-Koordinaten über den Parameter `-reoffset 0 0 0` auf Null gestellt und

ihre Nachkommastellen mit `-rescale 0.01 0.01 0.01` einheitlich auf "2" gesetzt.

Als Vorbereitung zur Beschleunigung von räumlichen Abfragen auf Laserpunkte der Befliegungen wird schließlich noch für jede LAS-Datei mit `lasindex` (Isenburg 2012) ein Quadtree-Index über die X-, Y-Koordinaten erstellt und parallel zu den LAS-Dateien abgelegt. Die Größe dieser binären Dateien mit der Dateiendung `.lax` beträgt maximal 0,1% der originalen LAS-Datei. Jede Quadtree-Zelle der Index-Datei speichert dabei eine Liste mit Intervallen, die festhält, in welchen Bereichen der LAS-Datei die Punkte der jeweiligen Zelle zu finden sind. Bei räumliche Abfragen müssen so nicht mehr alle Punkte der LAS-Datei sequentiell gelesen und analysiert werden, sondern nur jene, die in den zusammengeführten Intervall-Listen der betroffenen Quadtree-Geometrien übrig bleiben.

Damit ist die Migration der Rohdaten aus den Originalverzeichnissen in die neue Datenstruktur abgeschlossen. Anhang 1 zeigt am Beispiel der ersten Hintereisferner-Befliegung vom 11.10.2001 die zur Migration nötigen Shell-Befehle. Die Shell-Skripts zur Migration aller anderen Projekte sind unter <https://github.com/openwebcc/ba/tree/master/maintenance/rawdata/als> im GitHub-Repository für Skripts und HTML-Seiten, die im Rahmen der Bachelorarbeit erstellt wurden, verfügbar.

## 6.2 Ableitung von Metadaten

Zur Ableitung der Metadatenblätter im Textformat aus den LAS-Dateien der Punktwolken steht das frei nutzbare Programm `lasinfo` der LAStools zur Verfügung. Die Syntax zur Erzeugung des Metadatenblatts am Beispiel des ersten Flugstreifens der Hintereisbefliegung vom 11.10.2001 ähnelt dem Aufruf bei der Konvertierung in das LAS-Format:

```
| lasinfo -i H_011011_01.las -o ../meta/H_011011_01.las.info.txt -compute_density
```

Der Parameter `-i` bezeichnet wieder die LAS-Inputdatei, der Parameter `-o` legt fest, dass das Ergebnis in einem parallelen Verzeichnis `meta` (siehe Tabelle 4) mit der zusätzlichen Dateiendung `.info.txt` abgelegt werden soll und der Parameter `-compute_density` weist `lasinfo` an, Zusatzstatistiken zu abgedeckter Fläche, mittlerer Punktdichte und mittleren Punktabständen zu berechnen. Ein Blick in diese Datei im Anhang 2 zeigt neben allen Attributen, die der LAS-Standard im Header einer LAS-Datei vorsieht (ASPRS 2013, S.5), auch eine Fülle an Zusatzinformationen wie die beim Erzeugen der LAS-Datei hinzugefügte Projektion, Minima und Maxima aller Attribute des vorliegenden Punktdaten-Formats, die über den Parameter `-compute_density` berechneten Zusatzstatistiken, die Verteilung der Returns sowie bei klassifizierten Daten ein Histogramm der einzelnen Klassen (siehe Anhang 3).

Die Ableitung der generalisierten konkaven Hülle der Punktwolke erfolgt wiederum mit Hilfe der LAStools in einem zweistufigen Prozess durch Ausdünnen der Punktwolke mit `lasprecision` und Berechnen der konkaven Hülle mit `lasboundary`. Leider sind beide Tools nur als Windowsversionen verfügbar, weshalb

beim Ausführen der Programme unter Linux die frei verfügbar Windowsemulation wine zum Einsatz kam (Wine 2015). Die Shell-Befehlskette dafür zeigt Listing 3.

*Listing 3: Shell-Befehle zur Ableitung der generalisierten konkaven Hülle der Punktwolke*

```
wine lasprecision.exe -i H_011011_01.las \  
                      -o thinned.las \  
                      -rescale 1 1 1 \  
                      -thin_with_grid 10  
wine lasboundary.exe -i thinned.las \  
                    -o ../meta/H_011011_01.las.hull.wkt
```

Quelle: [https://github.com/openwebcc/ba/blob/master/tools/get\\_hull.sh](https://github.com/openwebcc/ba/blob/master/tools/get_hull.sh)

Beim Ausdünnen der Punktwolke mit `lasprecision.exe` wird über den Parameter `-rescale` zuerst die Präzision der X-, Y-, Z-Koordinaten auf Meter verringert und dann mit `-thin_with_grid` eine zusätzliche Ausdünnung am 10-Meter-Raster durchgeführt. Damit verringert sich die Punktzahl des ausgedünnten Flugstreifens auf einen Bruchteil der Originalgröße (z.B. zirka 2% bei `H_011011_01.las`). Im zweiten Schritt wird die konkave Hülle der ausgedünnten Punktwolke berechnet und im parallelen Metadatenverzeichnis `meta` als WKT-Polygon mit der zusätzlichen Dateiendung `.hull.wkt` gespeichert. Ein Blick in die derart erzeugte Datei `H_011011_01.las.hull.wkt` zeigt die Syntax eines WKT-Polygons:

```
POLYGON((635039 5182969, 635063 5182939, ... , 635028 5182978, 635039 5182969))
```

Leider ist `lasboundary.exe` nicht Teil der frei verfügbaren `LAStools`, sondern unterliegt der Lizenzierungspflicht. Auf Nachfrage bei Martin Isenburg ergab sich, dass die Verwendung von `lasboundary.exe` im Kontext dieser Arbeit und auch im Kontext einer möglichen Öffnung von Teilen der Applikation zu einem späteren Zeitpunkt keine Lizenzierungspflicht mit sich bringen würde (Persönliche Mitteilung Martin Isenburg, 11.12.2015). Aus diesem Grund wurde auch nicht aktiv nach einer Open Source Alternative zur Erzeugung der konkaven Hülle der Punktwolken gesucht.

Zur Ableitung der generalisierten Flugtrajektorie im WKT-Format kommt ein eigenes Python-Skript zum Einsatz. Es filtert entlang des Flugpfades die X-, Y-Koordinaten jener Punkte, deren Punktabstand mindestens 100 Meter beträgt. Bei einer Fluggeschwindigkeit von 75 m/s bedeutet dies, dass zirka alle 1,3 s ein Punkt in eine generalisierte Sub-Trajektorie übernommen wird. Am Beispiel des ersten Flugstreifens der Hintereisbefliegung vom 11.10.2001 erlaubt die damit verbundene Reduktion von 4713 auf 69 Punkte eine Ressourcen schonende Verwendung der generalisierten Geometrie im Web-Interface. Unter der Annahme, dass Flugpfade bei Laserscanning Befliegung möglichst gerade verlaufen sollten, ist die Generalisierung auf einen Punktabstand von mindestens 100 Metern damit gerechtfertigt. Ein Blick in die erstellte Datei `H_011011_01.las.traj.wkt` zeigt die Syntax für die WKT-Linestring-Geometrie:

```
|LINESTRING(640265 5187529, 640191 5187460, ..., 635412 5182836, 635341 5182764)
```

Die gefilterte Originaltrajektorie mit den Attributen GPS-Zeitstempel, X-, Y- und Z-Koordinate der Flugzeugposition, den Winkeln für *roll*, *pitch*, *heading* und *wander angle* des Flugzeugs werden im selben Durchgang wie die generalisierte WKT-Geometrie durch das Python-Skript `get_trajectories.py` erzeugt (siehe [https://github.com/openwebcc/ba/blob/master/tools/get\\_trajectories.py](https://github.com/openwebcc/ba/blob/master/tools/get_trajectories.py)). Im Gegensatz zu den Metadatenblättern und den generalisierten Geometrien wird sie später im Web-Interface zum direkten Datendownload angeboten. Ablageort ist wiederum das parallele Metadatenverzeichnis `meta`, die zusätzliche Endung lautet `.traj.txt`.

Zur Automatisierung des Arbeitsablaufs bei der Ableitung von Metadaten kommen zwei Shell-Skripts und das bereits erwähnte Python-Skript zum Einsatz, die im GitHub-Repository für Skripts unter <https://github.com/openwebcc/ba/tree/master/tools> zu finden sind. Die Erstellung der Metadatenblätter übernimmt `get_lasinfo.sh`, konkave Hüllen werden mit `get_hull.sh` erzeugt und die Filterung der Originaltrajektorie und der generalisierten Trajektorie übernimmt das Python-Skript `get_trajectories.py`.

## 6.3 Import der Metadaten in die PostgreSQL-Datenbank

### 6.3.1 Zentrale Metadatentabelle `meta`

Das Datenbankschema zur Ablage der Projekt-Eckdaten, Metadatenblätter und generalisierten Geometrien der konkaven Hüllen und Trajektorien für jede LAS-Datei besteht aus einer zentralen Metadatentabelle und einem Datenbankview als Basis für Abfragen aus dem Web-Interface. Am Beispiel des SQL-Befehls (Structured Query Language) in Listing 4 zeigt sich das Schema der Tabelle `meta`.

*Listing 4: SQL-Befehl zur Erzeugung der Metadatentabelle `meta`*

```
CREATE TABLE meta (
    gid SERIAL PRIMARY KEY NOT NULL, -- Primärschlüssel (auto-inkrement)
    ptype TEXT,                      -- Projekttyp (z.B. als)
    pname TEXT,                      -- Projektname (z.B. hef)
    cdate TEXT,                      -- Datum der Befliegung (z.B. 011011)
    cname TEXT,                      -- Befliegungskürzel (z.B. hef01)
    fname TEXT,                      -- Name der LAS-Datei
    fsize BIGINT,                    -- Größe der LAS-Datei in Bytes
    points BIGINT,                   -- Zahl der Punkte im LAS File
    srid INTEGER,                    -- EPSG Code der Projektion
    hull geometry(Polygon,0),        -- Geometrie der konkaven Hülle
    traj geometry(Linestring,0),     -- Geometrie der Trajektorie
    info JSON                        -- JSON-Objekt mit allen Attributen
);
```

Quelle: <https://github.com/openwebcc/ba/blob/master/maintenance/sql/schema.sql>

Als Primärschlüssel dient in der Spalte `gid` eine Zahl vom Datentyp `SERIAL`. Sie wird nach jeder Einfügeoperation in die Datenbank automatisch um den Wert 1 erhöht und erlaubt die eindeutige Identifikation jedes Datenbankeintrags. Die Spalten `ptype`, `pname`, `cdate` und `cname` spiegeln die Dateistruktur des Ablageschemas in Tabelle 3 wider. Die Spalten `fname`, `fsize` und `points` enthalten LAS-Datei spezifische Attribute wie Name und Größe der Datei in Bytes sowie die Zahl der Laserpunkte.

Die Spalte `srid` enthält den EPSG-Code der Projektion der Punktwolke, wobei SRID für "Spatial Reference System Identifier" steht und das Pendant zum EPSG-Code in der Datenbanksprache des PostGIS-Moduls darstellt. Er wird in der OGC Spezifikation "OpenGIS Implementation Specification for Geographic information - Simple feature access - Part 2: SQL option" definiert (Open Geospatial Consortium 2010b, S.15).

Bei den Spalten `hull` und `traj` handelt es sich um PostGIS-Geometrieobjekte vom Typ Polygon und Linie für die generalisierte konkave Hülle und Flugtrajektorie, sofern vorhanden. Geometriespalten einer Tabelle dürfen allerdings nur Geometrieobjekte mit der selben Projektion enthalten, was bei der Vielzahl an unterschiedlichen Flugstreifen, deren Hüllen und Trajektorien in der Tabelle `meta` abgebildet werden sollen, von Haus aus nicht gegeben ist. Deshalb werden die Geometrien vorerst ohne Projektion gespeichert (SRID=0) und der dazugehörige EPSG-Code in der Spalte `srid` notiert.

Die letzte Datenbankspalte mit dem Namen `info` beinhaltet schließlich alle Attribute der Metadatenblätter im so genannten JSON-Format (JavaScript Object Notation) (Ecma International 2013), ein einfaches Austauschformat für strukturierte Daten zwischen verschiedenen Programmiersprachen. Anhang 4 zeigt die Syntax des JSON-Formats.

### 6.3.2 Datenbankview für Abfragen `view_meta`

In einem nächsten Schritt wird ein so genannter Datenbankview, eine gespeicherte Abfrage, erstellt. Er enthält zusätzlich zu den Projekt-Eckdaten und LAS-Datei Eigenschaften der Tabelle `meta` eine Reihe neuer Spalten, die einfachere Datenbank Abfragen aus dem Web-Interface heraus ermöglichen sollen. Listing 5 zeigt den SQL-Befehl zur Erzeugung dieses Datenbankviews.

*Listing 5: SQL-Befehl zur Erzeugung des Datenbankviews `view_meta`*

```
CREATE VIEW view_meta AS (  
    SELECT gid,ptype,pname,cname,cdate,fname,fsize,points,info,srid,  
        to_date(cdate, 'YMMDD') AS datum,  
        info->>'system_identifizier' AS sensor,  
        (info->>'point_area')::numeric AS area,  
        (info->>'point_density')::numeric AS density,  
        ST_Transform(ST_SetSRID(hull,srid),4326) AS hull,  
        ST_Transform(ST_SetSRID(traj,srid),4326) AS traj  
    FROM meta  
    WHERE ptype='als'  
);
```

Quelle: <https://github.com/openwebcc/ba/blob/master/maintenance/sql/schema.sql>

Das Datum der Befliegung wird so über den Funktionsaufruf `to_date(cdate, 'YYMMDD')` AS `datum` von einer Zeichenkette im Format Jahr, Monat, Tag in ein echtes Datenbankdatum mit dem Spaltennamen `datum` umgewandelt. Dies ermöglicht zeitliche Abfragen auf den Datenbestand, wie zum Beispiel Selektionen von Befliegungen innerhalb einer Zeitspanne oder die zeitlich sortierte Ausgabe von Befliegungen.

Die nächsten drei Zeilen des CREATE-Befehls zeigen Abfragen auf die JSON-Struktur der vollständigen Metadatenattribute. Der Aufruf `info->>'system_identifizier'` AS `sensor` greift dabei direkt auf den Inhalt des JSON-Attributs `system_identifizier` zu und wird als Spalte `sensor` gespeichert. Gleiches gilt für die Fläche der Punktwolke im JSON-Attribut `point_area` und die durchschnittliche Punktdichte pro m<sup>2</sup> im Attribut `point_density`, die gleichzeitig über `::numeric` in echte Zahlen umgewandelt werden. Damit können für die neuen Spalten `area` und `density` Statistiken wie Summen oder Mittelwerte bei Abfrage berechnet werden.

Kernstück des SQL-Befehls zur Erzeugung des Datenbankviews für räumliche Abfragen sind die beiden mit `ST_Transform` beginnenden Zeilen. Hier werden die projektionslosen Geometrien der generalisierten konkaven Hülle und Trajektorie in projizierte PostGIS-Geometrien umgewandelt. Die Funktionsaufrufe `ST_Transform` und `ST_SetSRID` sind Datenbankfunktionen des SQL/MM Standards (Melton 2003), der vom PostGIS-Module implementiert wird (PostGIS Project Steering Committee 2015b). Der Vorgang beim Umwandeln in projizierte Geometrien ist zweistufig: `ST_SetSRID(hull, srid)` weist der konkaven Hülle zuerst ihre, in der Spalte `srid` gespeicherte, Originalprojektion zu und transformiert diese Geometrie anschließend über `ST_Transform(ST_SetSRID(hull, srid), 4326)` in eine Geometrie mit der Projektion EPSG:4326. Damit sind alle Hüllen und Trajektorien über geographische Koordinaten im WGS84 Referenzsystem in den Spalten `hull` und `traj` abfragbar.

### 6.3.3 Datenimport der Metadaten in die PostgreSQL-Datenbank

Für den Import der Metadaten in die PostgreSQL-Datenbank kommt ein eigenes Pythonskript zum Einsatz (siehe [https://github.com/openwebcc/ba/blob/master/tools/import\\_meta.py](https://github.com/openwebcc/ba/blob/master/tools/import_meta.py)), das aus den Metadatenblättern und WKT-Geometrien für konkave Hüllen und Trajektorien im Metadatenverzeichnis der Befliegungen die nötigen Daten extrahiert. Konkave Hüllen und Trajektorien können vom Importskript direkt aus den WKT-Dateien gelesen und mit `ST_GeomFromText` in PostGIS-Geometrien konvertiert werden. Die Umwandlung des Textformats der Metadatenblätter in das JSON-Format der Datenbankspalte `info` übernimmt ein eigenes Pythonmodul (siehe <https://github.com/openwebcc/ba/blob/master/www/lib/Laser/Util/las.py>). Zusätzlich werden die Pfade zu den einzelnen LAS-Dateien in ihre Unterverzeichnisse zerlegt und damit Typ und Name des Projekts, Name und Datum der Befliegung sowie der Name der LAS-Datei ermittelt und mit der Größe der LAS-Datei in Bytes ergänzt.



Das Einfügen der Metadaten und Geometrien erfolgt dann in zwei Phasen: Zuerst werden Projekt-Eckdaten und Attributdaten über einen INSERT-Befehl importiert und dann über die eindeutige gid des soeben eingefügten Datensatzes die Geometriedaten durch UPDATE-Befehle ergänzt. Listing 6 zeigt die dazu nötigen SQL-Befehle.

*Listing 6: SQL-Befehle zum Einfügen der Metadaten in die Tabelle meta*

```
INSERT INTO meta (ptype,pname,cname,cdate,fname,fsize,points,srid,info) VALUES
(
  'als', 'hef', 'hef01', '011011', 'H_011011_01.las',
  49377005, 1763453, 25832,
  {
    "system_identifizier": "ALTM 1225",
    "area" : 2854592.0,
    "density" : 0.62,
    -- weitere Einträge (siehe Anhang 4)
  }
) RETURNING gid AS NEWGID;
UPDATE meta SET hull=ST_GeomFromText('POLYGON(...))') WHERE gid=NEWGID;
UPDATE meta SET traj=ST_GeomFromText('LINRSTRING(...))') WHERE gid=NEWGID;
```

Quelle: [https://github.com/openwebcc/ba/blob/master/tools/import\\_meta.py](https://github.com/openwebcc/ba/blob/master/tools/import_meta.py)

Nach Abschluss des Imports der Metadaten und Geometrien in die PostgreSQL-Datenbankstruktur sind im Datenbankview `view_meta` automatisch alle zur Visualisierung im Web-Interface nötigen Metadaten verfügbar. Einen Screenshot dieses Karteninterfaces zeigt Abbildung 1.

## 6.4 Web-Interface für Metadaten und Rohdaten

Das Gerüst der Web-Applikation besteht aus einem Applikationsskript, das mit verschiedenen Parametern über den Browser aufgerufen werden kann, mehreren HTML-Templates, die den Basis-Quelltext der HTML-Seiten samt Platzhaltern für dynamische Inhalt beinhalten sowie einer Pythonbibliothek mit Modulen zur Kommunikation mit der Datenbank zum Befüllen der Templates und zum Abfangen von Fehlern. Im GitHub-Repository sind die Quelltexte dieser Skripts, Module und Dateien unter der Adresse <https://github.com/openwebcc/ba/blob/master/www> verfügbar.

Die Web-Applikation selbst ist über das Applikationsskript durch Eingabe der URL (Uniform Resource Locator) <http://geographie.uibk.ac.at/lidar/app.py> im Browser erreichbar und listet auf der Startseite tabellarisch die wichtigsten Kennzahlen aller vorhandenen Befliegungen auf. Über Hyperlinks gelangt man von dort aus direkt zum Karteninterface der Befliegung, dem eigentlichen Herzstück der Web-Applikation, das als Schaltbrett für Datenexploration und Datendownload dient (siehe Abbildung 1). Die Visualisierung des Karteninterfaces im Browser erfolgt mit Hilfe der Javascript-Bibliothek `Leaflet.js` (Agafonkin 2015) und zwei zusätzlichen Javascript-Modulen `Leaflet.draw` (Sánchez Ortega 2015) zum Zeichnen von Rechtecksausschnitten im Freihandtool und `Leaflet.omnivore` (MacWright 2015) zum Laden der generalisierten Geometrien für konkave Hüllen bei gekachelten Daten und Trajektorien bei Flugstreifen.

# LiDAR-Repository Geographie Innsbruck

Befliegung als: **hef:011011:hef01** (18 Dateien, 1.417 MB)

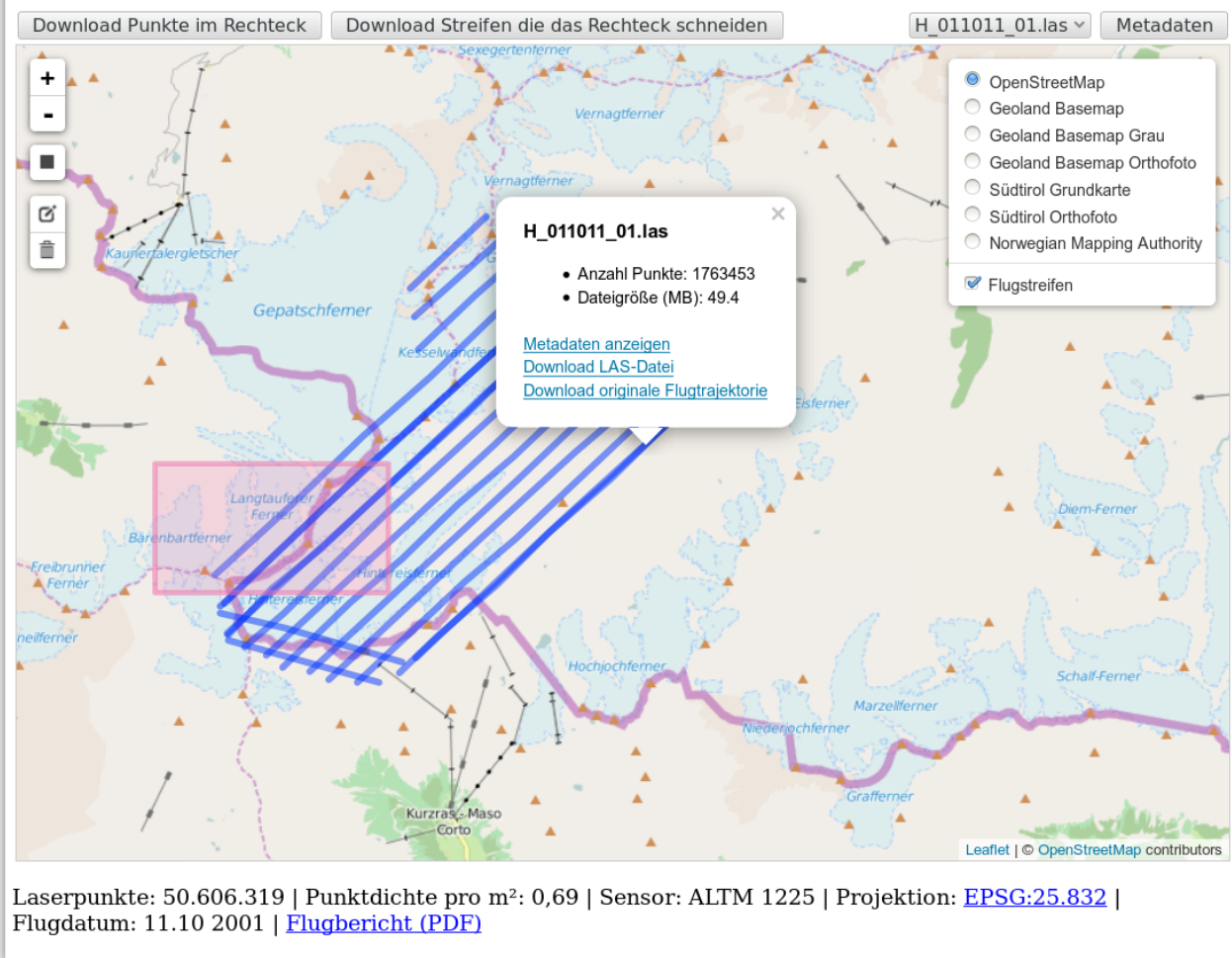


Abbildung 1: Screenshot des Karteninterfaces

## 6.4.1 Kartenhintergrund über Web Map Tile Services

Als Hintergrundkarten dienen WMTS-Dienste verschiedener Anbieter, bestehend aus vorgefertigten Serien gekachelter Bilder der Original-Rasterkarten in verschiedenen Zoomstufen. Abbildung 2 zeigt exemplarisch jeweils eine Kachel der im Web-Interface verfügbaren Hintergrundlayer. Über die Layernavigation in der rechten oberen Ecke des Karteninterfaces (siehe Abbildung 1) können die Hintergrundlayer je nach Gebietsstand ausgewählt werden.




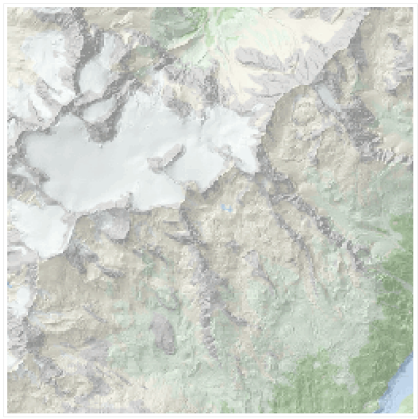

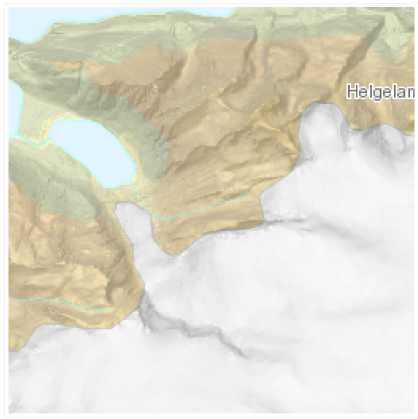
		
<p>OpenStreetMap (OpenStreetMap contributors 2015)</p>	<p>Österreichische Grundkarte (grau) (Stadt Wien et al. 2015)</p>	<p>Österreichische Grundkarte (Orthofoto) (Stadt Wien et al. 2015)</p>
		
<p>Südtirol (Grundkarte) (Autonome Provinz Bozen – Südtirol 2015)</p>	<p>Südtirol (Orthofoto) (Autonome Provinz Bozen – Südtirol 2015)</p>	<p>Grundkarte Norwegen 1:50.000 (Norwegian Mapping Authority 2014)</p>

Abbildung 2: Einzelne Kacheln der WMTS-Hintergrundlayer im Karteninterface

#### 6.4.2 Interaktiver Vektorlayer & Datendownload

Über dem Hintergrundlayer liegen zur Visualisierung der Ausdehnung der Punktwolke und als Zugang zum Datendownload die generalisierten Geometrien für Flugtrajektorien oder Kacheln. Der Einbau im Karteninterface erfolgt über das Leaflet-Plugin `Leaflet.omnivore` durch Laden so genannter GeoJSON-Objekte, die neben den Koordinaten der Linien- oder Polygonzüge auch ein Mindestmaß an Attributen zur Identifizierung der Geometrie beinhalten. Anhang 5 zeigt am Beispiel einer generalisierten Trajektorie der Hintereisferner-Befliegung vom 11.10.2001 die Syntax dieses GeoJSON-Objekts. `Leaflet.omnivore` liest dieses GeoJSON-Objekt ein und wandelt es in einen Leaflet-Layer um, der im Browser als Skalierbare Vektorgraphik (SVG) (W3C SVG Working Group 2011) visualisiert wird. Durch Klick auf die Geometrie der Trajektorie erscheint ein Pop-up Fenster mit Namen, Größe und Punktzahl der LAS-Datei sowie Hyperlinks zum Download der Punktwolke und originalen Flugtrajektorie, sofern vorhanden.

Das mit Hilfe von `Leaflet.Draw` implementierte Freihandtool im linken Bereich des Karteninterfaces erlaubt die Digitalisierung beliebiger Rechtecksausschnitte als Basis für den Download mehrere Flugstreifen oder Kacheln auf einmal. Dabei können entweder alle Streifen oder Kacheln, die das Rechteck schneiden, heruntergeladen werden oder nur jene Laserpunkte, die sich innerhalb des aufgezogenen Rechtecks befinden. In beiden Fällen wird zuerst die Liste der betroffenen LAS-Dateien über die Projekt-Eckdaten sowie Minima und Maxima der Rechteckskoordinaten ermittelt (siehe Listing 7).

*Listing 7: SQL-Befehl zum Filtern der LAS-Dateien in einem Rechtecksausschnitt*

```
SELECT gid,ptype,pname,cdate,cname,fname FROM view_meta
  WHERE ptype='als' AND pname='hef' AND cdate='011011' AND cname='hef01'
  AND ST_Intersects(
    ST_GeomFromText(
      'POLYGON((<Geographische Koordinaten des Rechtecks>))',
      4326
    ),hull
  )
```

Quelle: <https://github.com/openwebcc/ba/blob/master/www/lib/Laser/Util/web.py>

Die PostGIS-Funktion `ST_GeomFromText` wandelt die geographischen Koordinaten des Rechtecks in eine echte PostGIS-Geometrie um und verschneidet diese anschließend über `ST_Intersects` mit der generalisierten, konkaven Hülle im Datenbankview `view_meta`. Als Resultat dieser Operation steht eine Liste mit Pfaden zum Ablageort der betroffenen LAS-Dateien zur Verfügung. Für den Download der Dateien, die das Rechteck schneiden, müssen diese Dateien dann nur noch in den freigegebenen Downloadbereich des Servers über Unix Softlinks verlinkt werden:

```
ln -s /home/laser/rawdata/als/hef/011011_hef01/las/H_011011_04.las \
    /home/laser/download/als_hef_011011_hef01_H_011011_04.las
```

Zum Klippen der Laserpunkte am Rechteck ist ein weiterer Schritt nötig, denn die Filterung der Punkte erfolgt über das Programm `las2las` der `LAStools` und benötigt im Parameter `-keep_xy` die Koordinaten des Rechtecks in der Projektion der LAS-Datei. Listing 8 zeigt den SQL-Befehl zur Konvertierung der geographischen Koordinaten eines aufgezogenen Rechtecks in geographischer Projektion (EPSG:4326) in die Projektion der LAS-Datei (EPSG:25832).

Die PostGIS-Funktion `ST_Transform` erledigt dabei die Transformation zwischen den Projektionen, und `ST_Extent` liefert die Minima und Maxima der X-, Y-Koordinaten für den Parameter `-keep_xy`. Zur Beschleunigung der Filterung der Laserpunkte mit `las2las` werden die Pfade der betroffenen LAS-Dateien in eine temporäre Liste geschrieben und über den Parameter `-lof` (mnemotechnisch für "list of files") gewährleistet, dass nur diese LAS-Dateien bei der Filterung prozessiert werden. Den vollständigen Befehl zum Filtern der Punkte des Rechtecks im Screenshot des Karteninterface in Abbildung 1 zeigt Listing 9.

*Listing 8: SQL-Befehl zur Ermittlung der Rechtecks-Koordinaten in der Projektion der LAS-Datei*

```
SELECT ST_Extent(  
    ST_Transform(  
        ST_GeomFromText(  
            'POLYGON((<Geographische Koordinaten des Rechtecks>))',  
            4326  
        ),  
        25832  
    )  
) FROM view_meta WHERE gid=1000
```

Quelle: <https://github.com/openwebcc/ba/blob/master/www/lib/Laser/Util/web.py>

*Listing 9: las2las-Befehl zur Filterung der Laserpunkte in einem Rechteck*

```
las2las -merged \  
-keep_xy 630236 5183913 634796 5186493 \  
-lof /home/laser/download/lasfiles.txt \  
-o /home/laser/download/clipped.las
```

Quelle: <http://geographie.uibk.ac.at/lidar/app.py/points?>

*cid=als:hef:011011:hef01;extent=10.7072925,46.7960247,10.7663440,46.8183487*

Gemeinsam mit der Beschränkung auf betroffene LAS-Dateien beim las2las Aufruf über den Parameter -lof tragen die mit lasindex generierten Indizes der LAS-Dateien zur Beschleunigung der Filterung bei. Im vorliegenden Beispiel können so in weniger als drei Sekunden zirka 9,5 Millionen Laserpunkte in eine 255 MB große LAS-Datei mit gefilterten Laserpunkte innerhalb des Rechtecks geschrieben werden.

Das Karteninterface wird schließlich noch durch die Eckdaten der Befliegung wie Anzahl und Größe der LAS-Dateien, Anzahl der Laserpunkte, durchschnittliche Punktdichte pro m<sup>2</sup>, den verwendeten Sensor, EPSG-Projektion, Flugdatum und dem Befliegungsbericht als PDF, sofern vorhanden, ergänzt. Ein Auswahlménü mit allen verfügbaren Flugstreifen / Kacheln ermöglicht den direkten Zugang zu den detaillierten Metadaten auf einer weiteren Web-Seite, die das Metadatenblatt, die JSON-Notation der Attribute in der Datenbank sowie Downloadlinks zur LAS-Datei und der Originaltrajektorie enthält.

## 7 Ergebnisse und Fazit

Der im Rahmen dieser Arbeit entwickelte Workflow zur Verwaltung und Verteilung von LiDAR-Daten orientiert sich an bereits bestehenden Ansätzen und versucht, diese in neuer Form zu kombinieren und als *Proof of concept* unter Einsatz von Open Source Software zu implementieren. Die dabei gewonnenen Erkenntnisse seien an dieser Stelle als Reflexion der Konzeptions- und Implementierungsphase noch einmal mit ihren wichtigsten Aussagen festgehalten.

Die Ablagestruktur für ALS-Daten hat sich als flexibel genug erwiesen, um heterogene LiDAR-Datensätze

mit einem Mindestmaß an Dokumentation aufzunehmen. Sowohl Flugstreifen-Daten der multitemporalen Datenserie am Hintereisferner, im Kaunertal, im Montafon und im Vinschgau als auch gekachelte Daten des Engabreen Gletschers in Norwegen oder das klassifizierte Laserhöhenmodell für Vorarlberg konnten so in das neue Ablageschema überführt werden.

Die Wahl des LAS-Formats als primäres Speicherformat für Rohdaten gewährleistet dabei eine kompakte Ablage der Originalpunktwolken und ermöglicht mit Hilfe der LAS-Tools die effiziente, skriptgesteuerte Ableitung von Metadaten und generalisierten Geometrien zur weiteren Verwendung im Web-Interface. Die Auslagerung ressourcen- und zeitintensiver Prozessierungsschritte in Tagesrandbereiche über automatisierte Skripts hat sich vor allem bei der Migration der unsortierten Rohdaten und der Ableitung von Metadaten als vorteilhaft erwiesen.

Die Datenbankstruktur mit einer zentralen Metadatentabelle und einem Datenbankview für Abfragen aus dem Web-Interface erfüllt die Mindestanforderungen zur Ablage der Metadaten und Geometrien der generalisierten konkaven Hüllen und Trajektorien, wie sie im Web-Interface benötigt werden. Der Einsatz von PostgreSQL mit PostGIS als Datenbank *back end* stellt dabei alle nötigen Funktionalitäten zum Handling von Attributdaten und räumlichen Daten in verschiedensten Projektionen zur Verfügung.

Die Entscheidung, die Punktwolkendaten als LAS-Dateien im Dateisystem zu belassen, führt zu einer platzsparenden Datenablage ohne den Overhead der Datenduplizierung beim Import in eine Datenbank. Gleichzeitig gibt es kaum Einschränkungen bei Abfragen auf den Datenbestand, denn räumliche Abfragen lassen sich mit Hilfe der LAS-Indexdateien zügig ausführen.

Die bei der Visualisierung im Web-Interface eingesetzten WMTS-Hintergrundlayer erlauben das Einbinden von Kartendaten beliebiger Anbieter und lassen sich so auch auf andere Regionen erweitern. Die Wahl von OpenStreetMap als Basishintergrund stellt dabei sicher, dass für alle Gebiete ein Mindestmaß an räumlicher Information verfügbar ist.

Bei der Visualisierung der Punktwolkenausdehnung im Web-Interface bieten die generalisierten Trajektorien und konkaven Hüllen nicht nur einen raschen Überblick über die Abdeckung der einzelnen Flugstreifen oder Kacheln, sondern ermöglichen auch den direkten Zugang zu deren Metadaten. Das Freihandtool zum Datendownload nach Rechtecksausschnitten erlaubt eine rasche Auswahl der gewünschten Daten, ist aber derzeit noch auf Streifen-, bzw. Kacheldaten beschränkt. Originaltrajektorien können deshalb nur einzeln als Textdateien heruntergeladen werden.

## **8 Ausblick**

In puncto Rohdatenablage ergeben sich offene Fragen bei der weiteren Reduzierung des Speicherbedarfs bei stetig wachsenden Datenmengen. Die Verwendung von komprimierten LAS-Dateien zur weiteren Reduktion von Speicherplatz am Server wäre wünschenswert, scheitert derzeit aber noch am Fehlen eines offenen Standards zur Komprimierung von LAS-Dateien.

Die Erweiterung der Möglichkeiten zur Filterung der Punktwolken beim Datendownload wäre ebenso wünschenswert. Über zusätzliche Datenbankviews könnten auch Histogramme von klassifizierten Daten oder Minimum- und Maximumwerte zur Auswahl von Punktwolkendaten im Web-Interface herangezogen werden. Damit wären Abfragen auf Punkte nach Klassifizierung, Punkte nach Intensitätswerten oder Echo-Eigenschaften möglich. Bei Originaltrajektorien wäre die Konvertierung in ein GIS-taugliches Format (z.B. ESRI *Shapefiles*) ebenso wünschenswert wie die Implementierung des Downloads nach Ausschnitten analog zu den Flugstreifen und Kacheln im Freihandtool.

Der gesamte Downloadprozess ist derzeit noch nicht auf einen öffentlichen Zugang ausgelegt. Resultate sind nur im Intranet des Instituts für Geographie der Leopold-Franzens-Universität verfügbar und müssten in eine öffentliche Dateiablage überführt werden. In diesem Zusammenhang gilt es natürlich auch Zugriffsrechte zu klären. Als erster Beitrag für die Wissenschaftsgemeinde könnte so z.B. die multitemporale Serie des Hintereisferners als Open Data frei zur Verfügung gestellt werden. Im Rahmen der *Open Access Policy* für vom Österreichischen Wissenschaftsfonds (FWF) geförderte Projekte (FWF Der Wissenschaftsfonds 2015), könnten auch die Daten zukünftiger Forschungsprojekte, wie vom FWF gefordert, der Öffentlichkeit über das LiDAR-Daten Repository zugänglich gemacht werden.

Auf konzeptioneller Ebene steht aber vor allem die Frage nach den Integrationsmöglichkeiten von TLS-Daten in das bestehende System im Vordergrund. Der am Institut für Geographie der Leopold-Franzens-Universität Innsbruck verfügbar RIEGL VZ-6000 Laserscanner wird im Rahmen von Forschungsprojekten schon jetzt intensiv genutzt. Mit der geplanten Fortführung der multitemporalen Serie des Hintereisferners durch Fixinstallation eines zweiten Scanners gleicher Bauart zum laufenden Monitoring des Gletschers wird sich das Datenvolumen registrierter TLS-Punktwolken rapide vergrößern. Eine Integration in das bestehende System wäre demnach wünschenswert.

## 9 Zusammenfassung

Airborne Laserscanning hat sich zu einem leistungsfähigen und operationell einsetzbaren Standardverfahren in den Naturwissenschaften entwickelt. Die dabei entstehenden, räumlich und zeitlich immer höher aufgelösten Datensätze eröffnen neue Möglichkeiten der Forschung mit LiDAR-Daten, einem Forschungsfeld, das auch am Institut für Geographie der Leopold-Franzens-Universität Innsbruck eine lange Tradition besitzt. Ausgehend von ersten Untersuchungen zur Eignung von Laserscanning als Instrument zum Monitoring von Gletschern zu Beginn der Zweitausender-Jahre, erweiterte sich der Fokus bald auf geomorphologische Fragestellung bei Blockgletschern und Permafrostgebieten.

Der im Rahmen dieser Forschungsprojekte entstandene, umfangreiche Pool an ALS-Datensätzen umfasst neben einer weltweit einzigartigen, multitemporalen ALS-Daten Serie für den Hintereisferner mit 23 Befliegungen im Zeitraum 2001 bis 2013, auch zahlreiche weitere Datensätze im Bereich Ötztal, Kaunertal, Vorarlberg, Vinschgau und Norwegen. Alles in allem beläuft sich der unsortierte Rohdatenbestand auf zirka

3,9 TB, die es neu zu organisieren und für Projekte am Institut für Geographie der Leopold-Franzens-Universität Innsbruck zugänglich zu machen gilt. Die Konzeption des Workflows zur Verwaltung und Verteilung der LiDAR-Daten orientiert sich dabei an bereits bestehenden Ansätzen und versucht, diese in neuer Form zu kombinieren und als *Proof of concept* unter Einsatz von Open Source Software zu implementieren.

Am Beginn des Workflows steht die Migration der unsortierten Rohdaten in ein neues Ablageschema, das Platz für Punktwolkendaten, Trajektoriendaten der Flugstreifen sowie Dokumentation in Form von Befliegungsberichten bietet. Zentrales Format für die Ablage der Flugstreifen oder Kacheln ist dabei das LAS-Format, ein von der American Society for Photogrammetry & Remote Sensing zum Austausch von dreidimensionalen Laserscanning-Daten geschaffener offener Standard.

Mit Hilfe der LAStools, einer Softwaresammlung von Martin Isenburg, können in einem zweiten Schritt Metadaten aus den Punktwolken und Trajektorien, sofern vorhanden, extrahiert und in eine PostgreSQL / PostGIS Datenbank überführt werden. Das Datenbankschema bietet Platz für Eckdaten der Befliegungen (Projektname, Befliegungszeitpunkt, Anzahl und Größe der LAS-Dateien, etc.), Punktwolken-Attribute (Anzahl der Laserpunkte, durchschnittliche Punktdichte pro m<sup>2</sup>, verwendeter Sensor, Projektion, Flugdatum, Minima und Maxima von Koordinaten, gemessenen Intensitäten, etc.) sowie generalisierte Geometrien der Flugtrajektorien und konkaven Hülle der Punktwolkenausdehnung als Basis zur späteren Visualisierung im Web-Interface.

Der webbasierte Zugang zur Anzeige der Metadaten aus der Datenbank und dem Download der Rohdaten aus der Dateiablage erfolgt mit Hilfe des HTML5-Standards, der Javascript Kartenbibliothek Leaflet zur Visualisierung und Interaktion im Karteninterface sowie einer Pythonbibliothek als Schnittstelle zur Datenbank und zur Generierung des HTML5-Quelltextes. Das Karteninterface als zentrale Komponente besteht aus frei wählbaren Hintergrundkarten (z.B. OpenStreetMap, Österreichische Grundkarte, Grundkarte Südtirol) sowie den generalisierten Trajektorien bei Flugstreifen bzw. konkaven Hüllen bei gekachelten Daten zur Visualisierung der Punktwolkenausdehnung und als Zugang zu den Metadaten. Ein zusätzliches Freihandtool erlaubt den Download von Flugstreifen oder gefilterten Punkten über frei wählbare Rechtecksausschnitte. Die Bereitstellung der Daten selbst erfolgt im Intranet des Instituts für berechnete Benutzerinnen und Benutzer.

Dem Open Source Gedanken folgend sind auch alle Skripts und HTML-Seiten, die im Rahmen der Implementierung der einzelnen Phasen des Workflows zur Verwaltung und Verteilung der LiDAR-Daten erstellt wurden, unter <https://github.com/openwebcc/ba> in einem öffentlichen Quellcode-Repository abgelegt. Die Web-Applikation selbst ist unter der Adresse <http://geographie.uibk.ac.at/lidar/app.py> im Internet verfügbar.



## 10 Summary

This paper presents the design and implementation of an Open source based work flow for management and dissemination of airborne lidar data that has been collected as part of research projects carried out at the Department of Geography, University of Innsbruck. Based on research of current approaches to deal with lidar data management and dissemination, it combines identified solutions to design and implement a work flow consisting of three distinct phases. In the first phase, existing raw data for point clouds is converted to the LAS format and migrated to a unitized directory structure along with trajectory data and flight documentation if available. In the second phase, metadata is extracted from these LAS files and trajectories and stored in a PostgreSQL / PostGIS database along with generalized geometries for flight-paths and concave hulls of the 2D-pointcloud extent. The database schema consists of a single metadata table and an additional database view to provide easy access to attribute data and geometries from within the HTML5 standard conforming web interface. This web interface is implemented in phase three. It provides a Javascript based visualization with Leaflet to show the pointcloud extent using the generalized flight paths or hulls as vectorlayer on top of a background image layer displaying Web Map Tile Services of Openstreetmap or the Austrian basemap. Detailed metadata can be accessed by clicking on the interactive vector layer. In addition to that, a download utility is implemented that allows to select pointcloud data intersecting a digitized rectangle or to download data clipped to that rectangle. Download results are presented to authorized members of projects at the department. Following the spirit of Open source all scripts and web pages created during the implementation phase are provided at <https://github.com/openwebcc/ba>, the web interface is located at <http://geographie.uibk.ac.at/lidar/app.py>.

## 11 Literaturverzeichnis

Actueel Hoogtebestand Nederland (2015): AHN2 pointcloud viewer. Internet: <http://ahn2.pointclouds.nl> (Abruf 12.1.2016).

Agafonkin, V. (2105): Leaflet - an open-source JavaScript library for mobile-friendly interactive maps. Internet: <http://leafletjs.com> (Abruf 15.12.2015).

Aho, A.V., Kernighan, B.W. & Weinberger, P.J. (1987): The AWK Programming Language. Addison-Wesley Longman, Amsterdam (1987).

Apache Software Foundation (2015): Apache HTTP Server Project. Internet: <http://httpd.apache.org> (Abruf 15.12.2015).

ASPRS - The American Society for Photogrammetry & Remote Sensing (2013): LAS 1.4 Specification Version 1.4 -R13, 15 July 2013. Internet: [http://www.asprs.org/a/society/committees/standards/LAS\\_1\\_4\\_r13.pdf](http://www.asprs.org/a/society/committees/standards/LAS_1_4_r13.pdf) (Abruf 9.12.2015).

Autonome Provinz Bozen – Südtirol (2009): Geförderte Projekte im Rahmen des 1. Wettbewerbes für Projekte im Bereich der Wissenschaftlichen Forschung. Internet: <http://www.provinz.bz.it/bildungsfoerderung/forschung/1394.asp> (Abruf 5.1.2016).

Autonome Provinz Bozen – Südtirol (2015): Abteilung Informationstechnik - Geoportal-Dienste. Internet: <http://www.provinz.bz.it/informatik/kartografie/Geoportal.asp> (Abruf 8.1.2016).

Boehm, J. & Liu, K. (2015): NoSQL for storage and retrieval of large lidar data collections. In: ISPRS Geospatial Week 2015 (Volume XL-3/W3). La Grande Motte, France. S. 577-582.

Butler, H. & Gerlek, M. (2015): PDAL - Point Data Abstraction Library. Internet: <http://www.pdal.io> (Abruf 12.1.2016).

Butler, H., Schmidt, C., Springmeyer, D. & Livni, J. (2015): spatialreference.org. Internet: <http://spatialreference.org> (Abruf 22.12.2015).

Conrad, O., Bechtel, B., Bock, M., Dietrich, H., Fischer, E., Gerlitz, L., Wehberg, J., Wichmann, V., & Böhner, J. (2015): System for Automated Geoscientific Analyses (SAGA) v. 2.1.4. Geosci. Model Dev., 8, 1991-2007. S. 1991 - 2007. Internet: <http://www.geosci-model-dev.net/8/1991/2015/gmd-8-1991-2015.pdf> (Abruf 11.1.2016).

Cura, R., Perret, J. & Nicolas Paparoditis, N. (2015): Point Cloud Server (PCS): point clouds in-base management and processing. In: ISPRS Geospatial Week 2015 (Volume XL-3/W3). La Grande Motte, France. S. 531 - 539.

Ecma International (2013): Standard ECMA-404 - The JSON Data Interchange Standard (October 2013). Internet: <http://www.ecma-international.org/publications/files/ECMA-ST/ECMA-404.pdf> (Abruf 13.12.2015).

Ecma International (2015): Standard ECMA-262 - ECMAScript 2015 Language Specification. Internet: <http://www.ecma-international.org/publications/files/ECMA-ST/Ecma-262.pdf> (Abruf 15.12.2015).

Förster, K. (2015): GitHub-Repository für Skripts und HTML-Seiten die im Rahmen der Bachelorarbeit erstellt wurden. Internet: <https://github.com/openwebcc/ba> (Abruf 6.1.2016)

Free Software Foundation (1999): GNU Lesser General Public License, Version 2.1. Internet: <http://www.gnu.org/licenses/old-licenses/lgpl-2.1.en.html> (Abruf 10.12.2015).

FWF Der Wissenschaftsfonds (2015): Open Access Policy für vom FWF geförderte Projekte. Internet: <https://www.fwf.ac.at/de/forschungsfoerderung/open-access-policy> (Abruf 15.1.2016).

Galos, S.P., Klug, C., Prinz, R., Rieg, L., Dinale, R., Sailer, R. & Kaser, G. (2015): Recent glacier changes and related contribution potential to river discharge in the Vinschgau / Val Venosta, Italian Alps. In: *Geografia Fisica e Dinamica Quaternaria* Vol 38 (2), 2015. S. 143-154.

Geist, T., Elvehøy, H., Jackson, M. & Stötter, J. (2005a): Investigations on intra-annual elevation changes using multi-temporal airborne laser scanning data: case study Engabreen, Norway. In: *Annals of Glaciology* 42, 2005, S. 195-201.

Geist, T., Heller, A., & Stötter, J.(2004): Digitale Geländemodelle aus Airborne Laser Scanning Daten und ihre Anwendung für glaziologische Fragestellungen. In: Strobl, J., Blaschke, T. & Griesebner, G. (Hrsg.): *Angewandte Geoinformatik 2004, Beiträge zum 16. AGIT-Symposium, Salzburg 2004*, S. 163-171.

Global Positioning Systems Directorate (2013): Interface Specification IS-GPS-200H. Sep 24, 2013. Internet: <http://www.gps.gov/technical/icwg/IS-GPS-200H.pdf> (Abruf 10.12.2015).

GRASS Development Team (2002): GRASS 5.0 Users Manual. ITC-irst, Trento, Italy. Internet: [http://grass.osgeo.org/gdp/html\\_grass5](http://grass.osgeo.org/gdp/html_grass5) (Abruf 11.1.2016).

Helfricht, K., Lehning, M., Sailer, R. & Kuhn, M. (2015): Local Extremes in the Lidar-Derived Snow Cover of Alpine Glaciers. In: *Geografiska Annaler: Series A: Physical Geography, Volume 97*. S. 721–736.

Höfle, B. (2005): Entwicklung eines Informationssystems für Laserscannerdaten mit open source Software. Diplomarbeit am Institut für Geographie, Universität Innsbruck, Österreich.

Höfle, B., Rutzinger, M., Geist, T. & Stötter, J. (2006): Using airborne laser scanning data in urban data management - set up of a flexible information system with open source components. In: *Proceedings UDMS 2006: Urban Data Management Symposium. Aalborg, Denmark*. Internet: [http://wl27www658.webland.ch/bh/docs/papers/2006/Hoefle\\_et\\_al\\_UDMS\\_2006.pdf](http://wl27www658.webland.ch/bh/docs/papers/2006/Hoefle_et_al_UDMS_2006.pdf) (Abruf 11.1.2016).

Höfle, B. & Pfeifer, N. (2007): Correction of laser scanning intensity data: Data and model-driven approaches. In: *ISPRS Journal of Photogrammetry & Remote Sensing* 62 (2007). S. 415-433.

IOGP (International Association of Oil and Gas Producers) - Geomatics Committee (2015a). Internet: <http://www.iogp.org/geomatics-committee#251197-history> (Abruf 31.13.2015).

IOGP (International Association of Oil and Gas Producers) (2015b): EPSG Geodetic Parameter Dataset. Internet: <http://www.epsg-registry.org> (Abruf 22.12.2015).

Isenburg, M. (2012): LASindex - simple spatial indexing of LiDAR data. European LiDAR Mapping Forum 2012, Salzburg. Internet: <http://rapidlasso.com/2012/12/03/lasindex-spatial-indexing-of-lidar-data> (Abruf 10.1.2016)

Isenburg, M. (2015a): LASTools. Internet: <http://www.cs.unc.edu/~isenburg/lastools> (Abruf 28.12.2015).

Isenburg, M. (2015b): LASTools License Agreement. Internet: <https://www.cs.unc.edu/~isenburg/lastools/download/LICENSE.txt> (Abruf 10.12.2015).

Klug, C., Bollmann, E., Rieg, L., Sproß, M., Sailer, R. & Stötter, J. (2014): Detecting and Quantifying Area Wide Permafrost Change. In: Rutzing, M., Heinrich, K., Borsdorf, A. & Stötter, J. (Hrsg.): permafrost - Austrian Permafrost Research Initiative. Final Report Vol. 6, pp. 108. Verlag der Österreichischen Akademie der Wissenschaften. S. 57-88.

Khronos Group (2014): WebGL Specification, Version 1.0.3, 27 October 2014. Internet: <https://www.khronos.org/registry/webgl/specs/1.0> (Abruf 29.12.2015).

Krishnan, S., Crosby, C., Nandigam, V., Phan, M., Cowart, C., Baru, C. & Arrowsmith, R. (2011): OpenTopography: a services oriented architecture for community access to LIDAR topography. In: Proceedings of the 2nd International Conference on Computing for Geospatial Research & Applications, COM.Geo '11, Washington, DC, USA, May 23 - 25, 2011, Article No. 7.

Laserdata GmbH (2015): Management and Analysis of LiDAR Data. Internet: <http://laserdata.at> (Abruf 12.1.2016).

Lewis, P., Mc Elhinney, C.-P. & McCarthy, T. (2012): LiDAR data management pipeline; from spatial database population to web-application visualization. In: Proceedings of the 3rd International Conference on Computing for Geospatial Research and Applications, COM.Geo 2012, Washington, DC, USA, July 1-3, 2012, Article No. 16.

MacWright, Tom (2015): leaflet-omnivore - universal format parser for Leaflet & Mapbox.js. Internet: <https://github.com/mapbox/leaflet-omnivore> (Abruf 15.12.2015).

Martinez-Rubi, O., Verhoeven, S., Van Meersbergen, M., Schütz, M., Van Oosterom, P., Gonçalves, R. & Tijssen, T. (2015): Taming the beast: Free and open-source massive point cloud web visualization. Capturing Reality Conference Salzburg, 23.-25. November 2015. Internet: [http://www.gdmc.nl:8080/mpc/documents/papers/capturingreality2015-taming-the-beast-free-and-open-source-massive-point-cloud-web-visualization/at\\_download/file](http://www.gdmc.nl:8080/mpc/documents/papers/capturingreality2015-taming-the-beast-free-and-open-source-massive-point-cloud-web-visualization/at_download/file) (Abruf 14.1.2016).

Martinez-Rubi, O., Tijssen, T. P. M., Van Oosterom, P. J. M., Ivanova, M. & Verbree, E. (2014): Massive point clouds for eSciences - Mini-benchmark: description. Internet:

[http://www.gdmc.nl:8080/mpc/documents/reports/mini-benchmark-description/at\\_download/file](http://www.gdmc.nl:8080/mpc/documents/reports/mini-benchmark-description/at_download/file) (Abruf 30.12.2015).

Melton, J. (2003): Chapter 10 - SQL Multimedia and Application Packages. In: Advanced SQL:1999 - Understanding Object-Relational and Other Advanced Features. The Morgan Kaufmann Series in Data Management Systems, edited by Jim Melton, Morgan Kaufmann, San Francisco, 2003. S. 411-443.

MongoDB Inc. (2015a): MongoDB. Internet: <https://www.mongodb.org> (Abruf 12.1.2016).

MongoDB Inc. (2015b): MongoDB - GridFS — MongoDB Manual 3.2. Internet: <https://docs.mongodb.org/manual/core/gridfs> (Abruf 12.1.2016).

Netherlands eScience Center (2015): Massive Point Clouds for eSciences. Internet: <http://www.gdmc.nl:8080/mpc> (Abruf 12.1.2016).

Norwegian Mapping Authority - Kartverket (2014): Open and free Topographical land data in scale 1:50 000. Internet: <http://kartverket.no/en/Maps--Nautical-Charts/Gratis-kartdata/Open-and-Free-geospatial-data-from-Norway> (Abruf 15.12.2015).

Open Geospatial Consortium (OGC) (2010a): OpenGIS Implementation Specification for Geographic information - Simple feature access - Part 1: Common architecture. Internet: [http://portal.opengeospatial.org/files/?artifact\\_id=25355](http://portal.opengeospatial.org/files/?artifact_id=25355) (Abruf 13.12.2015).

Open Geospatial Consortium (OGC) (2010b): OpenGIS Implementation Specification for Geographic information - Simple feature access - Part 2: SQL option. Internet: [http://portal.opengeospatial.org/files/?artifact\\_id=25354](http://portal.opengeospatial.org/files/?artifact_id=25354) (Abruf 28.12.2015).

Open Geospatial Consortium (OGC) (2013): OGC Web Map Tile Service (WMTS) Simple Profile. Internet: <http://www.opengeospatial.org/standards/wmts> (Abruf 15.12.2015).

OpenStreetMap contributors (2015): OpenStreetMap. Internet: <https://www.openstreetmap.org> (Abruf 15.12.2015).

OpenTopography Facility (2015a): OpenTopography - High-Resolution Topography Data and Tools. Internet: <http://www.opentopography.org> (Abruf 12.1.2016).

OpenTopography Facility (2015b): OpenTopography - High-Resolution Topography Data and Tools - Data. Internet: <http://opentopo.sdsc.edu/lidar> (Abruf 12.1.2016).

OpenTopography Facility (2015c): OpenTopography - High-Resolution Topography Data and Tools - MyOpenTopo. Internet: <http://opentopo.sdsc.edu/login> (Abruf 12.1.2016).

OpenTopography Facility (2015d): OpenTopography - High-Resolution Topography Data and Tools - OpenTopography for Developers. Internet: <http://www.opentopography.org/developers> (Abruf 12.1.2016).

Petrini-Monteferri, F., Wichmann, V., Georges, C., Mantovani, D. & J. Stötter (2009): Erweiterung der GIS-

Software SAGA zur Verarbeitung von Laserscanning-Daten der Autonomen Provinz Bozen - Südtirol. In: Strobl, J., Blaschke T. & G. Griesebner (Hrsg.): Angewandte Geoinformatik 2009. Beiträge zum 21. AGIT-Symposium, Salzburg 2009. S. 618-623.

PostgreSQL Global Development Group (2015): PostgreSQL. Internet : <http://www.postgresql.org> (Abruf 13.12.2015).

PostGIS Project Steering Committee (2015a): PostGIS - Spatial and Geographic objects for PostgreSQL. Internet: <http://postgis.net> (Abruf 12.1.2016).

PostGIS Project Steering Committee (2015b): PostGIS - Spatial and Geographic objects for PostgreSQL - Chapter 8. PostGIS Reference. Internet: <http://postgis.net/docs/manual-2.1/reference.html> (Abruf 13.12.2015).

Python Software Foundation (2015): TkInter - Python Wiki. Internet: <https://wiki.python.org/moin/TkInter> (Abruf: 11.1.2016).

Ramsey, P. (2015): Pointcloud - A PostgreSQL extension for storing point cloud (LIDAR) data. Internet: <https://github.com/pgpointcloud/pointcloud> (Abruf 7.1.2016).

Ramsey, P. (2013): LIDAR in PostgreSQL with PointCloud. FOSS4G North America Conference, Minneapolis 2013. Internet: <http://boundlessgeo.com/wp-content/uploads/2013/05/foss4gna2013-pointcloud.pdf> (Abruf 15.12.2015).

Rieg, L., Wichmann, V., Rutzinger, M., Sailer, R., Geist, T. & Stötter, J. (2014): Data infrastructure for multitemporal airborne LiDAR point cloud analysis - Examples from physical geography in high mountain environments. In: Computers, Environment and Urban Systems 45, 2014. S. 137-146.

Rutzinger, M., Bremer, M., Sailer, R. & Stötter, J. (2013): Erfassung von räumlichen Daten in multiplen Dimensionen – topographisches LiDAR. In: Dander, V., Gründhammer, V., Ortner, H., Pfurtscheller, D. & Rizzolli, M.: Medienräume: Materialität und Regionalität, Innsbruck University Press. S. 209-221.

Sailer, R. (2013): C4AUSTRIA (Climate change consequences for the cryosphere) - Publizierbarer Endbericht. Internet: <https://www.klimafonds.gv.at/assets/Uploads/Projektberichte/ACRP-2008/20130627C4AUSTRIAEnderberichtRudolf-Sailer.pdf> (Abruf 4.1.2016).

Sailer, R., Bollmann, E., Hoinkes, S., Rieg, L., Sproß, M., & Stötter, J. (2012): Quantification of geomorphodynamics in glaciated and recently deglaciated terrain based on airborne laser scanning data. In: Geografiska Annaler Series A: Physical Geography, Vol 94. S. 17–32.

Sailer, R., Stötter, J., Bollmann, E., Fritzmann, P., Georges, C. & Geist, T. (2010b): ALS-X – Kombination und Bewertung von flugzeuggestützten Laserscannerdaten und TerraSAR-X Daten für glaziologische und schneehydrologische Fragestellungen. In: Innsbrucker Geographische Gesellschaft. (2010). Innsbrucker Jahresbericht 2008-2010, S.174-187.

- Samba Team (2015). samba - Opening Windows to a Wider World: Internet: <http://samba.org> (Abruf 6.1.2016).
- Samberg, A. (2007): An implementation of the ASPRS LAS standard. In: IAPRS Volume XXXVI, Part 3 / W52, 2007, S. 363-372.
- Sánchez Ortega, I. (2015): Leaflet.draw - Vector drawing and editing plugin for Leaflet. Internet: <https://github.com/Leaflet/Leaflet.draw> (Abruf 15.12.2015).
- Schütz, M. (2015): Potree | WebGL pointcloud renderer. Internet: <http://potree.org> (Abruf 12.1.2016).
- Shan, J. & Toth, C.K. (Hrsg.) (2008): Topographic Laser Ranging and Scanning: Principles and Processing. Boca Raton, Florida: CRC Press.
- Stadt Wien und Österreichische Länder bzw. Ämter der Landesregierung (2015): basemap.at - die Verwaltungsgrundkarte von Österreich, Internet: <http://basemap.at> (Abruf 15.12.2015).
- Suijker, P. M., Alkemade, I., Kodde, M. P. & Nonhebel, A.E. (2014): User requirements Massive Point Clouds for eSciences (WP1). Internet: [http://www.gdmc.nl:8080/mpc/documents/reports/user-requirements-massive-point-clouds-for-esciences/at\\_download/file](http://www.gdmc.nl:8080/mpc/documents/reports/user-requirements-massive-point-clouds-for-esciences/at_download/file) (Abruf 30.12.2015).
- The R Foundation (2015): The R Project for Statistical Computing. Internet: <https://www.r-project.org> (Abruf 11.1.2016).
- TopScan - Gesellschaft zur Erfassung topographischer Information mbH (2003): Projektbericht zu den Laserscannermessungen am Hintereisferner (Österreich) (11.10.2001 bis 26.09.2003).
- Van Oosterom, P., Martinez-Rubi, O., Ivanova, M., Horhammer, M., Geringer, D., Ravada, S., Tijssen, T., Kodde, M. & Gonçalves, R. (2015): Massive point cloud data management: design, implementation and execution of a point cloud benchmark. In: Computer & Graphics Volume 49, Juni 2015, S. 92-125.
- Vosselman, G. & Mass, H.G. (Hrsg.) (2010): Airborne and Terrestrial Laser Scanning - 1st Edition. Caithness: Whittles Publishing.
- W3C SVG Working Group (2011): Scalable Vector Graphics (SVG) 1.1 (Second Edition) - W3C Recommendation 16 August 2011. Internet: <http://www.w3.org/TR/SVG11> (Abruf 10.1.2016).
- WHATWG (Web Hypertext Application Technology Working Group) (2015): HTML - Living Standard. Internet: <https://html.spec.whatwg.org/multipage> (Abruf 15.12.2015).
- Wine - Run Windows applications on Linux, BSD, Solaris and Mac OS X.. Internet: <https://www.winehq.org> (Abruf 13.12.2015).

## 12 Anhang

Anhang 1: Shell-Skript zur Datenmigration der Hintereisferner Befliegung vom 11.10.2001

```
#!/bin/sh
#
# Datenmigration: Befliegung Hintereisferner, 11.10.2001
#

# Ordnerstruktur erstellen
mkdir -pv /home/laser/rawdata/als/hef/011011_hef01/{asc,las,bet,doc,meta}

# Rohdaten und Dokumentation kopieren
cd /home/laser/rawdata/als/hef/011011_hef01
cp -avu /mnt/netappa/Rohdaten/hef/hef01_011011/str/ala/*.ala ./asc/
cp -avu /mnt/netappa/Rohdaten/hef/hef01_011011/str/H_011011.bet ./bet/
cp -avu /mnt/netappa/Rohdaten/hef/hef01_011011/str/H_011011.dgn ./bet/
cp -avu /mnt/netappa/Rohdaten/hef/*Befliegingsbericht*.pdf ./doc/report.pdf

# ASCII Rohdaten bereinigen und nach LAS konvertieren
cd /home/laser/rawdata/als/hef/011011_hef01/asc/
for ALA in `ls *.ala`
do
    echo "entferne 32 bei x-Koordinaten in $ALA ..."
    awk '{gsub(/^32/, "", $2); print}' $ALA > $ALA.tmp
    mv $ALA.tmp $ALA

    LAS=`echo $ALA | sed s/\.[^\.]*$/.las/`
    echo "konvertiere nach ../las/$LAS ..."
    txt2las -i $ALA \
        -o ../las/$LAS \
        -parse txyzirn \
        -reoffset 0 0 0 \
        -rescale 0.01 0.01 0.01 \
        -epsg 25832 \
        -set_file_creation 284 2001 \
        -set_system_identifizier "ALTM 1225"
done

# Koordinaten der Trajektorie(n) bereinigen
for BET in `find /home/laser/rawdata/als/hef/011011_hef01/bet/ -name *.bet`
do
    echo "entferne 32 bei x-Koordinaten in $BET ..."
    awk '{gsub(/^32/, "", $2); print}' $BET > $BET.tmp
    mv $BET.tmp $BET
done
```

Quelle: [https://github.com/openwebcc/ba/blob/master/maintenance/rawdata/als/hef/011011\\_hef01.sh](https://github.com/openwebcc/ba/blob/master/maintenance/rawdata/als/hef/011011_hef01.sh)



Anhang 2: Metadaten für einen Streifen der Hintereisferner Befliegung vom 11.10.2001, generiert mit lasinfo

reporting all LAS header entries:

```
file signature:      'LASF'
file source ID:      0
global_encoding:     0
project ID GUID data 1-4: 00000000-0000-0000-0000-000000000000
version major.minor: 1.2
system identifier:    'ALTM 1225'
generating software:  'txt2las (version 150925)'
file creation day/year: 333/2011
header size:         227
offset to point data: 321
number var. length records: 1
point data format:    1
point data record length: 28
number of point records: 1763453
number of points by return: 996984 766469 0 0 0
scale factor x y z:   0.01 0.01 0.01
offset x y z:         0 0 0
min x y z:            634838.45 5182938.26 2276.04
max x y z:            640438.48 5187924.48 3333.11
```

variable length header record 1 of 1:

```
reserved            43707
user ID              'LASF_Projection'
record ID            34735
length after header  40
description           'by LASTools of rapidlasso GmbH'
```

GeoKeyDirectoryTag version 1.1.0 number of keys 4

```
key 1024 tiff_tag_location 0 count 1 value_offset 1 - GTModelTypeGeoKey:
ModelTypeProjected
key 3072 tiff_tag_location 0 count 1 value_offset 25832 -
ProjectedCSTypeGeoKey: UTM 32 northern hemisphere
key 3076 tiff_tag_location 0 count 1 value_offset 9001 -
ProjLinearUnitsGeoKey: Linear_Meter
key 4099 tiff_tag_location 0 count 1 value_offset 9001 -
VerticalUnitsGeoKey: Linear_Meter
```

reporting minimum and maximum for all LAS point record entries ...

```
X            63483845    64043848
Y            518293826    518792448
Z             227604      333311
intensity      5         704
return_number   1         2
number_of_returns 1         2
edge_of_flight_line 0         0
scan_direction_flag 0         0
classification  0         0
scan_angle_rank 0         0
user_data       0         0
point_source_ID 0         0
gps_time 380600.732777 380694.999970
number of first returns: 996984
```

```

number of intermediate returns: 0
number of last returns:      996984
number of single returns:    230515
covered area in square meters/kilometers: 2854592/2.85
point density: all returns 0.62 last only 0.35 (per square meter)
      spacing: all returns 1.27 last only 1.69 (in meters)
overview over number of returns of given pulse: 230515 1532938 0 0 0 0 0
histogram of classification of points:
      1763453  never classified (0)

```

Quelle: `/home/laser/rawdata/als/hef/011011_hef01/las/H_011011_01.las`

Anhang 3: Zusätzliche Metadaten bei klassifizierten Daten am Beispiel einer Kachel des vogis-Datensatzes im Raum Bregenz, generiert mit lasinfo

```

histogram of classification of points:
      17491  never classified (0)
      1381382  unclassified (1)
      75370257  ground (2)
      5405243  low vegetation (3)
      2921713  medium vegetation (4)
      2321032  high vegetation (5)
      11623012  building (6)
      76147  noise (7)
      320222  keypoint (8)
      606199  water (9)
      191634  rail (10)
      26489  overlap (12)
      7106  tower (15)
      3286  wire connector (16)

```

Quelle: `/home/laser/rawdata/als/vogis/111018_vb11/las/2011_11275202.laz`

#### Anhang 4: JSON-Repräsentation der Attribute in der Spalte info der Metadatentabelle meta

```
{
  "file_doy": 284,
  "file_gpsweek": 1135,
  "file_name": "H_011011_02.las",
  "file_size": 95025265,
  "file_year": 2001,
  "global_encoding": 0,
  "maximum": {
    "classification": 0.0,
    "edge_of_flight_line": 0.0,
    "gps_time": 380977.574412,
    "intensity": 736.0,
    "number_of_returns": 2.0,
    "point_source_ID": 0.0,
    "return_number": 2.0,
    "scan_angle_rank": 0.0,
    "scan_direction_flag": 0.0,
    "user_data": 0.0,
    "x": 640441.43,
    "y": 5188133.42,
    "z": 3323.05
  },
  "minimum": {
    "classification": 0.0,
    "edge_of_flight_line": 0.0,
    "gps_time": 380865.258534,
    "intensity": 5.0,
    "number_of_returns": 1.0,
    "point_source_ID": 0.0,
    "return_number": 1.0,
    "scan_angle_rank": 0.0,
    "scan_direction_flag": 0.0,
    "user_data": 0.0,
    "x": 633890.94,
    "y": 5182428.64,
    "z": 2276.04
  },
  "point_area": 4801444.0,
  "point_density": 0.71,
  "point_format": 1,
  "point_spacing": 1.19,
  "points": 3393748,
  "points_by_return": [
    1905915,
    1487833,
    0,
    0,
    0
  ],
  "projection": "UTM 32 northern hemisphere",
  "srid": 25832,
  "system_identifier": "ALTM 1225"
}
```

Quelle: eigene Datenbankabfrage

Anhang 5: GeoJSON-Objekt für die generalisierte Trajektorie des ersten Flugstreifens der Hintereisferner-Befliegung vom 11.10.2001 mit Koordinaten und Metadaten

```
geojson = {
  "type": "FeatureCollection",
  "crs": {
    "type": "name",
    "properties": {
      "name": "EPSG:4326"
    }
  },
  "features": [
    {
      "type": "Feature",
      "properties": {
        "gid": 1000,
        "fname": "H_011011_01.las",
        "fsize": 49.4,
        "points": 1763453,
        "has_traj": true
      },
      "geometry": {
        "type": "LineString",
        "coordinates": [
          [10.8390514, 46.8265305],
          [10.8086639, 46.8077375],
          [10.7731786, 46.7847364]
        ]
      }
    }
  ]
}
```

Quelle: <http://geographie.uibk.ac.at/lidar/app.py/geom?cid=als:hef:011011:hef01>

### **Eidesstattliche Erklärung**

Ich erkläre hiermit an Eides statt durch meine eigenhändige Unterschrift, dass ich die vorliegende Arbeit selbständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel verwendet habe. Alle Stellen, die wörtlich oder inhaltlich den angegebenen Quellen entnommen wurden, sind als solche kenntlich gemacht.

Ich erkläre mich mit der Archivierung der vorliegenden Bachelorarbeit einverstanden.

15.1.2016

---

Datum

---

Unterschrift