

Geoprocessing mit Handheld-Laser-Scanner erfassten Point Cloud-Daten

Bachelorarbeit

Studiengang Informatik
OST – Ostschweizer Fachhochschule
Campus Rapperswil-Jona

Frühjahrssemester 2021

Autoren:	Denis Nauli, Nadine Sennhauser
Betreuer:	Stefan Keller
Weitere Beteiligte:	Joël Schwab
Experte:	Claude Eisenhut
Gegenleser:	Andreas Rinkel

Abstract Deutsch

Im Zentrum dieser Arbeit steht das Geoprocessing - d.h. Verwalten und Konvertieren - und das Geovisualisieren von Laser-SLAM-basierten Punktwolken im Aussenbereich. Punktwolken können fotogrammetrisch berechnet oder mit LiDAR-Sensoren direkt erfasst werden.

Für die Bereiche Archäologie und Denkmalpflege wurden drei Handlaserscanner (Leica BLK2GO, GeoSLAM ZEB-Revo RT und iPad Pro für den Consumer-Bereich) an drei unterschiedlichen Objekten getestet. Der Test der Handlaserscanner hat gezeigt, dass sie durchaus Vorteile gegenüber den fotogrammetrischen Verfahren haben, da sie auch bei schlechten Lichtverhältnissen schnelles Messen und das sofortige Einsehen der Messresultate ermöglichen. Die professionellen Geräte haben jedoch einen hohen Preis und deren Auswertungsprogramme sind unausgereift. Das iPad erreicht bei einem 10 m grossen Objekt nur eine Genauigkeit von 10-50 cm, während die beiden teureren Geräte bei gleicher Grösse lediglich 5 cm abweichen. Auch die Reichweite des iPads ist mit 4 m deutlich geringer als die der spezialisierten Geräte mit 15 m.

Neben dem Test der Handlaserscanner wurde ein Software-Prototyp für die Verwaltung von Punktwolken und deren einfachen Publikation im Web entwickelt. Das Resultat ermöglicht das Hochladen, Visualisieren (3D und 2D), sowie das Teilen von Punktwolken. Eine Export-Funktion erlaubt es zudem, die Punktwolke im LAZ- oder TIFF-Format herunterzuladen, damit sie in GIS-Programmen für Fortgeschrittene - wie QGIS oder CloudCompare - weiterverarbeitet werden kann.

Abstract Englisch

This work focuses on geoprocessing – i.e. managing and converting – and geovisualization of laser-SLAM-based point clouds in the outdoor area. Point clouds can be calculated photogrammetrically or directly detected with LiDAR sensors.

Three handheld laser scanners (Leica BLK2GO, GeoSLAM ZEB-Revo RT and iPad Pro for the consumer sector) were tested on three different objects for the fields of archaeology and heritage preservation. The handheld laser scanner tests have shown that they have advantages over photogrammetric methods, since they allow faster measurements even in poor light and the immediate visualisation of the measurement results. However, the professional devices are expensive and their evaluation programs are not as mature as needed. The iPad only achieves an accuracy of 10-50 cm on a 10 m object while the two more expensive devices differ only 5 cm at the same object size. Also the iPad's range of 4 m is significantly shorter than the specialized devices' range of 15 m.

In addition to testing the handheld laser scanners, a software prototype was developed for the management of point clouds and their easy publication on the web. The result allows you to upload, visualize (3D and 2D), and share point clouds. An export function also allows you to download the point cloud in LAZ or TIFF format for further processing in advanced GIS programs such as QGIS or CloudCompare.

Management Summary

Aufgabenstellung

Bei dieser Arbeit geht es um das Geoprocessing - d.h. Verwalten und Konvertieren - und um das Geovisualisieren von Laser-SLAM-basierten Punktwolken im Aussenbereich. Punktwolken sind Punktsammlungen, bei der jeder Punkt aus einer 3D-Koordinate und ggf. zusätzlichen Attributen wie Farbe, geometrischen Normalen, Aufnahmezeitpunkt oder weiteren besteht. Ein Beispiel dazu befindet sich in Abbildung 1. Punktwolken können fotogrammetrisch aus Bildern berechnet oder mit LiDAR-Sensoren direkt erfasst werden. Ein integrierter SLAM-Algorithmus ermöglicht es dem Handlaserscanner während des Messen von neuen Punkten sich über die bereits Bekannten zu positionieren.

Das erste Ziel der Arbeit war die Evaluation von drei Handlaserscannern für die Bereiche Archäologie und Denkmalpflege: Leica BLK2GO, GeoSLAM ZEB-Revo RT und iPad Pro für den Consumer-Bereich. Für die Verbesserung der Verwaltung von Punktwolken und deren einfachen Publikation im Web sollte zudem ein Software-Prototyp entwickelt werden. Damit die detailreichen Daten auch für weitere Zwecke verwendet werden können, wie namentlich zum Erfassen von Gebäudegrundrissen in OpenStreetMap, sollte zudem eine 2D-Ansicht der Daten ermöglicht werden.

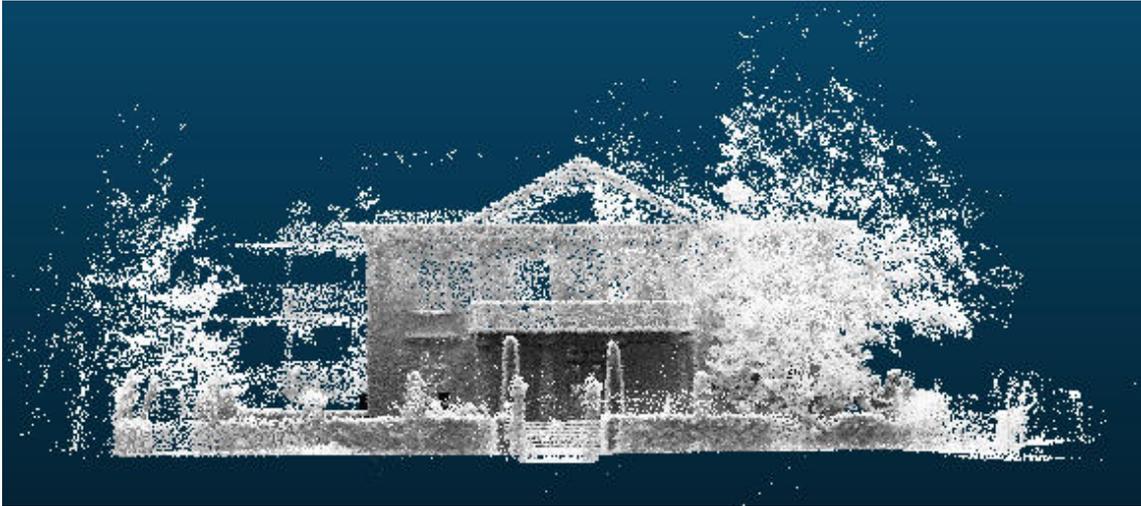


Abbildung 1: Punktwolke der Villa Aurum gemessen mit dem GeoSLAM ZEB Revo-RT

Vorgehen und Technologie

Der Einsatz der drei Handheld-Laserscanner wurde an drei unterschiedlichen Objekten getestet. Diese waren die Burgruine Weinfelden (TG), die Villa Aurum in Rapperswil (SG) und die Eisenplastik am OST Campus Rapperswil (SG). Bei den Messungen in Weinfelden und bei der Villa konnten die Punktwolken mit Referenzpunkten verglichen werden, um die Genauigkeit zu beurteilen.

Für die Erarbeitung der Webapplikation Point Cloud Browser wurde die Softwarebibliothek Django mit integriertem React-Framework gewählt, das die Viewer-Software Potree für die Darstellung der Punktwolken verwendet. Das Zusammenspiel der Komponenten wurde in Abbild 2 dargestellt. Die Punktwolken werden im komprimierten Format LAZ im Filesystem abgelegt. Über eine separate Verarbeitungs-Pipeline, die auch als Commandline Tool verwendet werden kann, können diese in ein für die Ansicht im Browser optimiertes Format, sowie in eine 2D-Ansicht konvertiert werden. Besonders grosse Punktwolken werden in angemessenen Chunks über HTTP an den Server übermittelt. Die Verarbeitungs-Pipeline Pointcloudpipeline basiert auf den Bibliotheken PDAL und untwine. Um eine einfache Installation zu ermöglichen, wurde der Point Cloud Browser dockerisiert und die Pointcloudpipeline auf PyPi veröffentlicht.

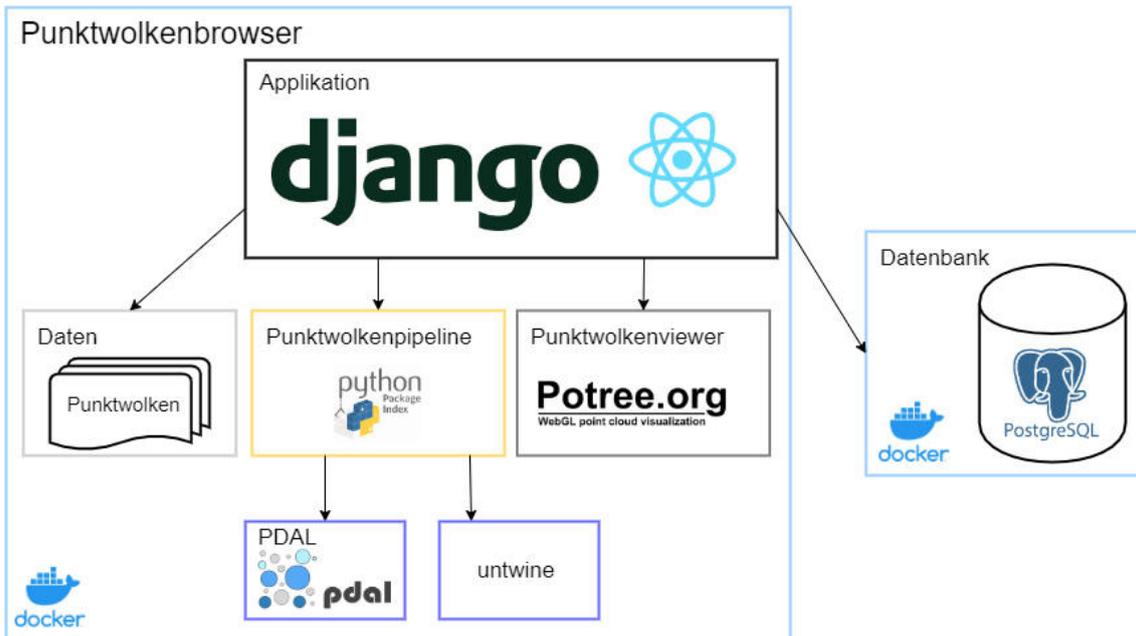


Abbildung 2: Overview Diagramm

Ergebnis

Die Tests der Handheld-Laserscanner haben gezeigt, dass sie durchaus Vorteile gegenüber den fotogrammetrischen Verfahren haben. So erlauben sie zum Beispiel auch Messungen bei schlechten Lichtverhältnissen. Im Vergleich zu Stativ mit Stop-and-Go sind sie schneller in der Datenerfassung und ermöglichen das sofortige Einsehen der Messresultate. Die professionellen Geräte haben jedoch einen hohen Preis und deren Auswertungsprogramme sind unausgereift. Das iPad erreicht bei einem 10 m grossen Objekt nur eine Genauigkeit von 10-50 cm, während die beiden teureren Handlaserscanner bei gleicher Grösse nur 5 cm abweichen. Auch die Reichweite des iPads ist mit 4 m deutlich geringer als die der spezialisierten Geräte mit ca. 15 m.

Der Point Cloud Browser ermöglicht das Hochladen, Visualisieren in 3D und 2D, sowie das Teilen von Punktwolken. Mit der Export-Funktion kann die Punktwolke im LAZ- oder TIFF-Format heruntergeladen werden, damit sie in GIS-Programmen für Fortgeschrittene - wie QGIS oder CloudCompare - weiterverarbeitet werden kann. Abbildung 3 zeigt die Benutzeroberfläche des fertigen Prototyps.



Abbildung 3: Benutzeroberfläche Point Cloud Browser

Ausblick

Der Point Cloud Browser ist ein lauffähiger Prototyp, der unter der BSD-2 Lizenz beliebig eingesetzt und weiterentwickelt werden kann. Auch die Pointcloudpipeline ist für alle verfügbar. Das Teilen der 3D Visualisierung ist nur eine der sinnvollen Erweiterungen, die empfehlenswert sind. Dabei könnten der Kamerawinkel, die Punktdichte und weitere Konfigurationen per Link geteilt werden. Auch eine Verbesserung der 2D Ansicht wäre nutzbringend, damit der Benutzer auch in dieser rotieren, zoomen und verschieben kann und vielleicht sogar die Farbskala selber bestimmen könnte. Auch die Eingabenbestimmung der Berechnungsparameter könnte je nach Benutzerwunsch in der 2D Ansicht ergänzt werden. Es gibt also noch einige Features, die nachgerüstet werden können. Der Point Cloud Browser birgt ein riesiges Potential, als vollständig freies Tool den gerade explodierenden Markt von 3D Daten als unkomplizierte Verwaltungs- und Visualisierungssoftware für Punktwolken zu bereichern.

Aufgabenstellung

SABA_FS21_Slot_2_Aufgabenstellung

Geoprocessing mit Handheld-Laser-Scanner erfassten Point Cloud-Daten

- Bachelorarbeit: **Nadine Sennhauser und Denis Nauli**, Frühlingssemester 2021
- Betreuer: Prof. Stefan Keller, Institut für Software / Abteilung Informatik, OST Campus Rapperswil, sowie Mitarbeiter am Institut für Software
- Externer Partner: (-)
- Weitere Beteiligte: Joël Schwab am Institut für Software.
- Organisatorisches (wöchentl. Meetings): Do. 10:30 Uhr
- Repository: Webapp [Pointcloudbrowser](#), Geoprocessing-Software [Pointcloudpipeline](#)
- Weitere wichtige Websites: tbd.
- Documentation: [Dokumentation](#),.

Hintergrund

Bei dieser Arbeit geht es um das Geoprozessieren - d.h. das Verwalten und Konvertieren - und um das Geovisualisieren, von Laser-SLAM-basierten Point Cloud-Daten im Aussenbereich (outdoor) mittels Open Source-Software-Komponenten. Ziel ist, 3D-Objekte sowie 2D-Basisdaten (Pläne) günstig und rasch im Web zu visualisieren.

Point Clouds (Punktwolken) sind eine Sammlung von Punkten. Jeder Punkt besteht aus einer Koordinate (X,Y,Z) plus zusätzliche Attribute. Punktwolken können aus einer Kombination von Fotografien erstellt werden. Genauer hingegen sind LiDAR-Sensoren kombiniert mit Farbkamera und vor allem mit SLAM (Simultaneous Localization and Mapping). Es sind dies namentlich Leica BLK2GO, GeoSLAM ZEB Revo RT sowie Consumer-Geräte wie iPad Pro.

Beispiele für entsprechende Datenprodukte sind archäologische Notgrabungen im Nahbereich, Denkmalschutz oder einfach Kartographierungen.

Diese neue Technologie erlaubt die vereinfachte, günstige Erfassung von 3D-Objekten, die einfache Publikation im Web (namentlich der Notgrabungen), sowie als Ausblick die hochaktuelle Nachführung von freien Geodatenbanken wie OpenStreetMap. Mit "Hochaktuell" gemeint ist einerseits, dass es sich um neuste Sensor-Technologie handelt und andererseits im Sinne einer Geodatenbank, die zeitnah aktualisiert werden kann.

Das Thema steht lose im Zusammenhang eines basisfinanzierten Forschungsprojekts "Laser-SLAM-Based Mobile Mapping mit SW-Integration und Visualisierung" des Instituts für Software am OST Campus Rapperswil und dem Amt für Archäologie des Kantons Thurgau. Es setzt Interesse voraus an der Software-Entwicklung mit 3D-Daten, Geodaten, Webapplikationen (Javascript), Python sowie C++ und JavaScript.

Ziele

- Erfahrungen sammeln anhand einigen realen Objekten (Burgstelle, Einzelgebäude, Eisenplastik) und dokumentieren zu LiDAR-Sensoren, Point Clouds und deren Geodatenverarbeitung (Zielgruppe: GIS-Spezialisten).
- Realisierung einer Webapplikation als Showcase/Prototyp zur Visualisierung von Outdoor Point Clouds inkl. effizienter Datenverwaltung.
- Realisierung als Showcase/Prototyp einer Geoprozessierungs-Pipeline (in Python) zur Konversion von 3D-Point-Clouds nach 2D-Vektordaten.

Aufgaben

Diese Arbeit umfasst folgende Arbeiten:

1. Entsprechende Technologien evaluieren, kennenlernen (v.a. Leica BLK2GO, GeoSLAM ZEB Revo RT, iPad Pro) und dokumentieren (u.a. von Potree (Open Source) vs. Cesium ("Halb" Open Source)).
Visualisierung von Point Cloud-Daten über API in Webapplikationen anhand ausgewählter, bestehender Open-Data-Daten sowie selber erfasster realer Objekte,
2. Konversion von 3D nach 2D als Geoprozessierungs-Pipeline, Commandline Interface (z.B. mit PDAL Open Source, CloudCompare, etc).
3. Datenverwaltung von massiven Point Cloud-Datensätzen (z.B. mit Untwine/Entwine Open Source) inkl. Architektur-Evaluation (Verwaltung als Punkt- oder Mesh-Strukturen?)
Anforderungen ans Team

Lieferobjekte / Deliverables

- Dokumentation der Evaluationen, Messkampagnen und Erfahrungen mit Point Cloud-Technologien.
- Webapplikation (Prototyp, Open Source) zur Visualisierung von Point Clouds nach vorgängiger Evaluation (Potree vs. Cesium vs. NN), inkl. Realisierung einer entsprechenden effizienten Point Cloud-Datenverwaltung (Prototyp, Open Source).
- Geoprozessierungs-Pipeline (Prototyp, Open Source) zur Konversion von 3D-Point-Clouds nach 2D-Vektordaten mittels obiger eigener Beispiele.

Bibliographie (Literaturverzeichnis)

- [1] <https://maps.zh.ch/s/w73btqyy>
- [2] <https://www.pixel8.earth/>
- [3] <https://pixel8earth.medium.com/the-case-for-affordable-and-open-3d-mapping-to-accelerate-augmented-reality-and-autonomy-bc782d4134ea>
- [4] <https://pixel8earth.medium.com/deploying-the-lowest-earth-orbit-satellite-to-edit-openstreetmap-cca4676052ed>

Inhaltsverzeichnis

Kapitel I	Technischer Bericht	13
1	Einführung	13
1.1	Problemstellung	13
1.2	Vision	14
1.3	Ziele und Unterziele	14
1.4	Vorgehen	15
1.5	Einleitung in die Thematik	16
2	Stand der Technik	19
2.1	Punktwolken Desktop Programme	19
2.2	Webbasierte Punktwolken Viewer	20
2.3	Punktwolken Verwaltung	22
2.4	Punktwolken Verarbeitung	25
3	Evaluation	25
3.1	Kriterien	25
3.2	Evaluation der iPad Apps	26
3.3	Evaluation Punktwolke vs Meshes	27
3.4	Evaluation von Punktwolken Viewer	28
3.5	Evaluation von Punktwolken Verwaltung	31
3.6	Evaluation von Punktwolken Verarbeitung	33
3.7	Evaluation Technologie Punktwolkenbrowser	36
3.8	Evaluation Technologie Punktwolkenpipeline	40
4	Erfahrungsbericht Handlaserscanner	40
4.1	Messobjekte	40
4.2	Leica BLK2GO	43
4.3	GeoSLAM ZEB-Revo RT	45

4.4	iPad Pro	47
4.5	Structure from Motion Referenzmessung	48
4.6	Gegenüberstellung der Handlaserscanner	49
4.7	Handlaserscanner vs Structure from Motion	51
4.8	Vorgehen beim Vergleich	54
4.9	Schlussfolgerung	55
5	Umsetzungskonzept	56
5.1	Aufbau	56
5.2	Punktwolken Hoch- und Herunterladen	57
5.3	Punktwolken Ansicht	58
5.4	2D-Ansicht und Export	59
6	Schlussfolgerung	61
6.1	Zielerreichung	61
6.2	Weiterentwicklung	62
6.3	Wir sagen Danke	64
Kapitel II Projektdokumentation		65
1	Vision	65
2	Anforderungsspezifikation	65
2.1	Requirements	65
2.2	Benutzer-Analyse	68
2.3	Aktoren	72
2.4	Use Cases	74
2.5	User Stories	76
2.6	Nichtfunktionale Anforderungen	77
3	Design	81
3.1	Architektur	81
3.2	Wichtige Abläufe	85
3.3	UI Design	91
4	Testing	93
4.1	Automatische Testverfahren	93
4.2	Manuelle Testverfahren	93
5	Resultate und Weiterentwicklung	94

5.1	Resultat	94
5.2	Weiteres Vorgehen	96
6	Projektmanagement	96
6.1	Stakeholder	96
6.2	Zeitplan	97
6.3	Risiken	98
7	Projektmonitoring	100
7.1	Ereignisse	100
7.2	Soll-Ist-Zeit-Vergleich	102
7.3	Codestatistik	103
7.4	Fazit	104
8	Softwaredokumentation	105
8.1	Installationsanleitung und Benutzerhandbuch	105
8.2	Schnittstellendokumentation	105
Anhang		106
1	Evaluation	107
1.1	Recherche Viewer	107
1.2	Recherche Speicher	118
1.3	Recherche Prozessierung	129
2	Scanner Vergleich	138
3	2D Berechnungen	169
4	Qualitätsmanagement	179
4.1	Qualitätsmassnahmen	179
4.2	Systemtests	182
4.3	Usability Test	189
4.4	Performance Test	210
5	Risiken	214
6	Organisation	224
6.1	Besprechungsnotizen	224
6.2	Konstruktionsplan	230
7	UI-Entwurf	233
8	Architektur	238

9	Pointcloudbrowser Manual	240
10	Pointcloudpipeline Manual	248
11	Glossar	253
12	Quellenverzeichnis	255
13	Codeverzeichnis	257
14	Abbildungsverzeichnis	258
15	Tabellenverzeichnis	260

Kapitel I

Technischer Bericht

1 Einführung

1.1 Problemstellung

Punktwolken sind Sammlungen von Punkten. Jeder Punkt besteht aus einer Koordinate (X,Y,Z) und ggf. zusätzlichen Attributen. Die hochaktuelle, auf LiDAR und SLAM basierende Technologie der Handlaser-scanner erlaubt ein schnelles und effizientes 3D-Erfassen von Objekten, resultierend in Punktwolken.

Die effizientere Erfassung, Verwaltung und Präsentation solcher 3D-Punktwolken verspricht einen Mehrwert für die Archäologie und die Denkmalpflege und genau da setzt dieses Projekt an. Die Archäologie wendet zurzeit vielerorts fotogrammetrische Strategien zur 3D-Datengewinnung an. Dieses Verfahren ist zeitintensiv und stark von den Lichtverhältnissen abhängig. Auch für die Nachführung der OpenStreetMap-Geodatenbank wäre ein einfaches Verfahren zur Erfassung der Umwelt und der Ableitung von 2D-Vektordaten sinnvoll. In naher Zukunft werden vermutlich viele unserer mobilen Geräte mit einem LiDAR-Sensor ausgestattet sein. So kann jeder, zum Beispiel auf seinem täglichen Weg zur Arbeit, die Umgebung beim Vorbeilaufen erfassen und diese Daten als Grundlage für die Nachführung der OpenStreetMap-Geodatenbank verwenden.

Zurzeit haben die Archäologen und Denkmalpfleger sich nicht abschliessend geeinigt, wie

die 3D-Daten letzten Endes gespeichert, beziehungsweise archiviert werden sollen. Dabei ist noch zu klären, wie die Punktwolken effizient gespeichert werden können. Es gibt noch keine kostenlosen oder vollständig quellenöffentliche Tools, mit denen Punktwolken auf einem Server gespeichert und zusätzlich visualisiert und geteilt werden können. Oftmals liegen die Punktwolken lokal beim Ersteller und müssen zum Teilen via Fileshare zum Endbenutzer gesendet werden. Der Nachteil davon ist sicher der aufwändige Übermittlungsprozess und die Gefahr, dass solch grosse Dateien plötzlich mehrfach in unterschiedlichen Versionen auf den Computern vorhanden sind.

Das Thema steht lose im Zusammenhang mit dem basisfinanzierten Forschungsprojekt Laser-SLAM-Based Mobile Mapping mit SW-Integration und Visualisierung des Instituts für Software am OST Campus Rapperswil und dem Amt für Archäologie des Kanton Thurgau.

1.2 Vision

In dieser Bachelorarbeit wird über den geeigneten Einsatz von verschiedenen Handlaser-scanner im Outdoorbereich aufgeklärt. Die Vor- und Nachteile der neuen Technologie werden im Vergleich zu den bestehenden Verfahren geprüft und die Genauigkeit ausgewertet.

Die Daten einer Messung sollen sinnvoll verwaltet, verarbeitet, visualisiert und geteilt werden können, um einen Mehrwert für die Nutzer zu generieren. Dazu wird in dieser Arbeit ein Prototyp zur Verwaltung und Darstellung von Punktwolken erstellt, welcher die Daten zentral speichert, sodass sie direkt angeschaut und geteilt werden können. Eine Verarbeitungspipeline, um 3D-Punktwolken zu 2D-Ansichten zu verarbeiten, wird als Commandlinetool ebenfalls im Rahmen dieser Arbeit entwickelt.

1.3 Ziele und Unterziele

Der Auftrag definiert 3 Ziele:

Dokumentation

Erfahrungen zu Handlaser-scanner, Punktwolken und deren Geodatenverarbeitung und Verwaltung sammeln und für die Zielgruppe GIS-Spezialisten dokumentieren. In diesem Ziel ist auch das Testen der drei Geräte (Leica BLK2Go, GeoSLAM Revo-RT und iPad Pro) und die Analyse bestehender Punktwolkenviewer und Verarbeitungstools enthalten.

Webapplikation

Realisierung einer Webapplikation als Prototyp zur Visualisierung von Outdoor-Punktwolken mit einer effizienten Datenverwaltung.

Geoprozessierungs-Pipeline

Realisierung einer Geoprozessierungs-Pipeline in Python als Prototyp zur Konvertierung von 3D-Punktwolken zu 2D-Vektordaten.

Die erarbeiteten Requirements zu den drei Zielen finden Sie im Teil 2 unter Requirements.

1.4 Vorgehen

Als erster Schritt wurde viel zum aktuellen Stand der Technik recherchiert und Wissen über Begrifflichkeiten rund um Punktwolken angeeignet. Schnell wurde bemerkt, dass viele Faktoren berücksichtigt werden müssen, um eine brauchbare Anwendung zum Verwalten und Publizieren der Punktwolken zu erstellen.

Damit die 3D-Daten schnell und genau erfasst werden können, wurden moderne Handlaser-scanner getestet. Die Messresultate wurden miteinander verglichen und entsprechend dokumentiert. Anschliessend wurde ermittelt, welche Dateiformate für die Langzeitspeicherung einer grossen Anzahl Punktwolken bestmöglichst geeignet sind und wie die Punktwolken im Web präsentiert werden können.

Darauf aufbauend wurde ein Commandline Tool aus bestehenden Tools zur Punktwolkenverarbeitung in Python erstellt, welches dem Nutzer die Handhabung mit diesen erleichtern sollte. Zeitgleich entstand eine Webapplikation, in welcher die Punktwolken gesammelt und dargestellt werden können.

1.5 Einleitung in die Thematik

1.5.1 Punktwolken

Punktwolken sind Sammlungen von Punkten, welche mit einer X, Y und Z Koordinate versehen sind. Tausende bis Millionen oder sogar Milliarden von Punkten ergeben eine sogenannte Punktwolke, welche eine detaillierte 3D-Oberflächen abbilden kann. Zu den Punkten können zusätzlich Attribute, wie z.B. geometrische Normalen, Farbwerte, Aufnahmezeitpunkt oder Messgenauigkeit erfasst sein.[1]

Eine Punktwolke entsteht typischerweise auf direktem Wege durch einen Laserscan mit LiDAR-Technologie oder durch ein fotogrammetrisches Verfahren.

Eines der grössten Probleme mit den Punktwolken ist die enorme Datenmenge. Obwohl meist nur die Koordinaten und Farbwerte gespeichert werden, ist durch die hohe Punktdichte eine sehr grosse Datenmenge zu erwarten.[2] Deshalb ist die Darstellung und Verwaltung von 3D-Daten knifflig. Für die Verwaltung und Vorbereitung der Darstellung gibt es bereits etliche Dateiformate.[3]

3D-Punktwolken werden immer relevanter und die Erfassungsgeräte immer handlicher und kosteneffizienter. Zum Beispiel sind iPad Pro und iPhone 12 bereits mit Laserscanner ausgerüstet. Auch in der autonomen Fahrindustrie werden LiDAR Sensoren für die Orientierung und Umgebungserkennung eingesetzt. Da in diesem Gebiet fleissig geforscht wird, kann angenommen werden, dass wir in Zukunft standardmässig LiDAR Sensoren in unseren Handys finden werden. Dies ermöglicht einer riesigen Community den Zugang zur heute eher teuren Erfassung von Punktwolken mit Laserscannern.

1.5.2 Mesh

Ein direkter Konkurrent zur Punktwolke ist das Mesh-Modell, bei welchem die Punkte in ein Vektormodell umgewandelt wurden. Drei Punkte bilden jeweils eine Fläche. Mit Mesh bezeichnet man in der Computergrafik ein Polygonnetz zur geometrischen Beschreibung von Oberflächen.[7] Dadurch ist die Oberfläche als Fläche sichtbar und das Modell noch schneller interpretierbar.

Der Nachteil am Mesh ist allerdings die aufwändige Vorarbeit.

1.5.3 LiDAR[4]

Mit Hilfe der LiDAR-Methode (Light detection and ranging) wird mit Laserstrahlen die Oberfläche eines Bereiches abgescannt. Durch die Reflektion eines Laserstrahls auf der Oberfläche wird der Vektor des Punktes im drei dimensionalen Raum bestimmt. Mehrere Laserimpulse ergeben so ein 3D-Abbild einer Oberfläche. Ein gutes Gerät misst pro Sekunde mehrere Hunderttausend solcher Punkte.

Die LiDAR-Methode kennt drei verschiedene Systeme:

- **Puls-basiert**

Das System sendet einen Laserimpuls aus und misst die Zeit, die der Lichtimpuls braucht, um vom Sender zur Oberfläche zu gelangen und durch die Reflektion am Objekt vom Sensor wieder eingefangen wird.

- **Phasen-basiert**

Phasen-basierte LiDAR-Systeme messen die Distanz, indem sie die Phase eines modulierten Laserstrahls verwenden, um die Distanz als Bruchteil der Wellenlänge des modulierten Signals zu berechnen. Dieses Verfahren ist sehr präzise, benötigt aber sehr viel Energie und wird deshalb meist nur in Innenräumen verwendet.

- **Geiger-Modus**

Dieses System wird auch Photonenzählsystem genannt und arbeitet mit extrem empfindlichen Detektoren, die durch ein einzelnes Photon ausgelöst werden können. Sie sind dadurch für höhere Höhenlagen geeignet. Geiger-Modus-Systeme leiden jedoch unter sehr hohen Geräuschpegeln und anderen Betriebsbeschränkungen, was sie wesentlich seltener macht als Linear-Modus-Systeme.

Die Messmethoden mit den verschiedenen Geräten lässt sich noch einmal in drei Kategorien unterteilen:

- **Terrestrial LiDAR Scanning (TLS)**

Der Scanvorgang erfolgt mit einem stationären LiDAR-Sensor. Heutzutage weisen

diese Geräte die höchste Genauigkeit auf und sind in einem Bereich von bis zu mehreren Hundert Metern anwendbar. Die Vorbereitung und Installation der Geräte ist aber sehr zeitaufwändig.

- **Airborne LiDAR scanning (ALS)**

Der LiDAR-Sensor wird dabei an einem Flugobjekt befestigt und misst aus der Luft. Mit diesen Geräten wird meist eine Genauigkeit von 0.2 bis 1 m erreicht.[5] Grosse Flächen können so einfach gescannt werden. Der Nachteil ist allerdings, dass z.B. die Hausfassaden nicht abgebildet werden können, sondern nur die Aufsicht.

- **Mobile LiDAR scanning (MLS)**

Die Messung erfolgt vom Boden aus, aber nicht von einer fixen Station sondern einem fahrenden Auto oder einem bewegbaren Handlaserscanner. Diese Geräte zeichnen sich durch ihre schnelle und einfache Erfassung von Punktwolken aus.

1.5.4 Fotogrammetrie

Zur Erfassung von 3D-Punktwolken werden häufig fotogrammetrische Verfahren angewendet. Im Gegensatz zur laserbasierten Aufnahme werden dabei Bilder mit einer Standardfotokamera aufgenommen. Diese Bilder werden mit entsprechender Software zu 3D-Punktwolken verarbeitet und die Abbildungsgeometrie zum Zeitpunkt der Aufnahme aus den Bildern rekonstruiert.[8] Dazu werden Bilder mit hoher Redundanz benötigt. Bei der Orientierung der Bilder kommt z.B. ein Kalman-Filter oder ein Verfahren der Computer Vision wie Structure-from-Motion (heute üblich) zum Einsatz. Kennt man die Position, Ausrichtung und die innere Orientierung der Kamera, kann jeder Strahl im Raum beschrieben werden. Durch Verwendung von mindestens zweier verschiedener Aufnahmepositionen, kann über den Schnittpunkt der Strahlen jeder Objektpunkt dreidimensional berechnet werden, wie in Abbildung I.1 ersichtlich. Dabei kommen in der Regel noch Verfahren der Ausgleichsrechnung zum Einsatz.[2]

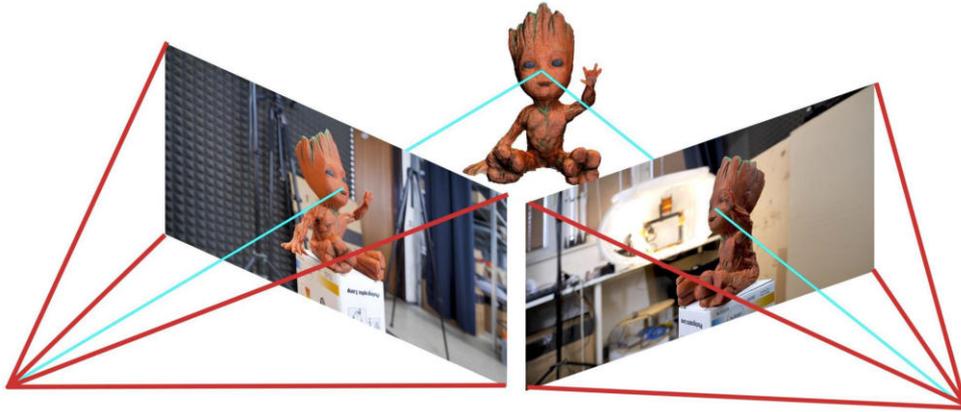


Abbildung I.1: Beispiel Fotogrammetrie [19]

Die Aufnahmeverfahren werden unterteilt in Luftbild- und Satellitenfotogrammetrie, terrestrische Fotogrammetrie und Nahbereichsfotogrammetrie.[8]

1.5.5 SLAM

SLAM ist ein Akronym für "Simultaneous Localization and Mapping". Dabei wird das Lokalisieren und Kartieren der Umgebung zeitgleich vorgenommen. Somit wird nicht nur ein Modell der Umgebung erstellt, sondern auch die eigene Position in diesem bestimmt. SLAM kann mit laserbasierten Messmethoden oder fotogrammetrischen Methoden verwendet werden.[9]

2 Stand der Technik

2.1 Punktwolken Desktop Programme

Die grössten Opensource Programme, welche mit Punktwolken umgehen können, sind CloudCompare, MeshLab und seit neuestem QGIS.

CloudCompare ist ein Programm, welches ausschliesslich für Punktwolken und Meshes verwendet werden kann. Entsprechend werden zahlreiche Datenformate unterstützt. CloudCompare bietet sehr viele spezifische Funktionen. Punktwolken können zum Bei-

spiel visualisiert, zugeschnitten, aber auch zueinander referenziert werden. Dank der breiten Unterstützung verschiedener Dateiformate wird es häufig auch einfach zum Konvertieren vom einen ins andere Format verwendet. CloudCompare ist aber weniger stark in der Bearbeitung von Meshes. Die PDF-Dokumentation zu CloudCompare ist leider nicht mehr ganz aktuell, dennoch findet man sich mit Onlinehilfen und der Dokumentation schnell in der Software zurecht.

MeshLab ist ein System, um Meshes zu prozessieren und editieren. Es bietet Funktionen, um Punktwolken in Meshes zu überführen und diese anschliessend zu verfeinern. Daneben ist es spezialisiert auf die Aufbereitung von Modellen für den-3D Druck.

QGIS ist ein professionelles GIS-Programm, das bisher vor allem im 2D Bereich eingesetzt wurde. Seit der Version 3.18 unterstützt es auch Punktwolken. Die Punktwolken können als EPT importiert werden und über einen speziellen Viewer in 3D betrachtet werden. Dies ist aber zurzeit noch nicht so schnell und intuitiv wie bei CloudCompare. Zum Beispiel ist die Navigation noch sehr umständlich, da nur um den Nullpunkt gedreht werden kann. Da es sich dabei um ein enterprise GIS-Programm handelt, ist es für GIS Spezialisten, die damit bereits arbeiten, empfehlenswert. Für den alleinigen Zweck der Betrachtung einer Punktwolke ist QGIS für einen Anfänger zu umfangreich.

2.2 Webbasierte Punktwolken Viewer

Das grösste Problem bei der Darstellung von Punktwolken ist die Datenmenge, die über das Web gesendet und schlussendlich in einem Browser angezeigt werden soll. Deshalb gibt es in dem Bereich auch bereits viele Ansätze und Forschungsarbeiten.

Opensource Viewer:

- **Potree** (<https://github.com/potree/potree>)

Potree ist Punktwolkenviewer, mit dem man nicht nur seine Visualisierung anpassen, sondern auch Messungen verschiedenster Art durchführen kann. Punktwolken werden dabei beim Bewegen Stück für Stück geladen, sodass nicht die gesamte Datenmenge auf einmal gestreamt werden muss. Potree ist vollständig Opensour-

ce.

- **Cesium** (<https://cesium.com/>)

In Cesium werden die Punktwolken auf einem 3D-Globus abgebildet. So können die Punktwolken am exakten Ort auf der Karte angezeigt werden. CesiumJS ist Opensource, während die Plattform Cesium Ion Closedsource ist.

- **WebODM** (<https://github.com/OpenDroneMap/WebODM>)

WebODM ist die Webversion von Open Drone Map und ist Opensource. Sie ist für die Messung mit einer Drohne ausgelegt und ermöglicht es Karten, Punktwolken und Oberflächenmodelle aus Luftbildern zu berechnen. Für die 3D-Visualisierung von Punktwolken integriert WebODM Potree. Daneben bietet es viele Fotogrammetriefunktionen, um aus den Drohnenbildern die Daten zu gewinnen.

- **iTowns** (<https://github.com/iTowns/itowns>)

iTowns ist ein Three.js-basiertes JS/WebGL-Framework für die 3D-Visualisierung von Geodaten in unterschiedlichen Formaten (3D tiles, DOM, OGC Standarts, Punktwolken, GeoJSON, Gpx, KML, Collada, GLTF OBJ und weitere)

- **Plasio** (<https://github.com/verma/plasio>)

In Plasio kann der Benutzer seine Punktwolke in LAS oder LAZ auswählen und im Browser visualisieren lassen. Die letzte Änderung am Tool ist allerdings schon 4 Jahre her.

- **AHN pointcloud viewer** (<https://github.com/NLeSC/ahn-pointcloud-viewer-ws>)

Der AHN pointcloud Viewer eignet sich für Punktwolken der Grösse eines ganzen Landes. Leider kann dieser Viewer nur in Chrome und Microsoft Edge verwendet werden. Auch hier ist die letzte Änderung schon 4 Jahre her.

Interessante Closedsource Viewer:

- **udStream von Euclidean** (<https://www.euclidean.com/udstream/>)

Beschreibung: UdStream verspricht trotz Milliarden von Punkten flüssige Bewegungen bei voller Punktdichte. Sie verwenden dabei nicht den Ansatz, Teilmengen in den Hauptspeicher zu laden, sondern haben einen optimal indexierten Zugriff auf

die Daten im Endspeicher entwickelt. Die Webseite ist allerdings ziemlich schlecht (tote Verlinkungen, schlechte Usability, unverständliche Navigation).

- **PointSharePlus** (<https://pointshareplus.com/>)

Beschreibung: Der PointSharePlus ist ein kostenpflichtiger Pointcloud Viewer, der unsere Requirements nahezu vollständig erfüllen würde. Er wirbt mit dem Speichern, Ansehen, Analysieren und Teilen von 3D Modellen in den Formaten OBJ, PLY, Revit, Navisworks, IFC und FBX.[10] Er könnte dadurch in Zukunft als eine kostenpflichtige Alternative zu unserem Tool verwendet werden.

Genauere Informationen zu den Viewern sind im Anhang 1.1 hinterlegt. Eine Evaluation der Opensource Viewer ist zudem im Kapitel I.3.4 beschrieben.

Viele Viewer, darunter auch Cesium und der Potree, versuchen durch optimiertes Segmentieren der Punktwolke eine Ansicht zu generieren. Insbesondere bei der Bewegung wird die Punktdichte dabei reduziert. Entweder erscheinen die ausgedünnten Punkte dann grösser oder man sieht relativ grosse freie Flächen zwischen den Punkten.

Fast alle der bestehenden Opensource Tools erlauben das Anzeigen von Daten, die auf dem Rechner abgelegt sind. Die kommerziellen Closedsource-Tools sind darauf ausgelegt, Daten von zentralen Servern oder aus der Cloud zu visualisieren. Der Vorteil eines solchen Systems ist es, dass nicht jeder eine Kopie der riesigen Punktwolkendaten bei sich lokal braucht.

2.3 Punktwolken Verwaltung

Die Datenverwaltung ist aufgrund der oben beschriebenen Datenmenge eine Herausforderung. Zum einen möchte man grosse Mengen an 3D Rohdaten speichern, braucht aber für die Darstellung vorprozessierte Segmente. Das Problem wurde bereits von vielen angegangen und alleine CloudCompare unterstützt 30 verschiedene Datenformate.

Die Datenformate lassen sich in verschiedene Gruppen unterteilen:

- **Strukturiert/Unstrukturiert:**

Dabei haben die Daten eine feste Struktur, wie ein festes Raster. Solche Punktdaten werden mit FARO-, Leica HDS- oder ZF-Scanner erzeugt. Sie erleichtern das Abfragen und Verarbeiten. Das kann man sich vorstellen wie das Suchen nach einer Zahl in einer sortierten Liste.[11] Beispiel: Suchen wir die Fünf und sind bei einer Sechs gelandet, können wir die Suche beenden, da nachher nur noch grössere Zahlen kommen. Bei unstrukturierten Daten werden die einzelnen Punkte der Wolke ohne gemeinsamen räumlichen Beziehungen erfasst.[12] Handlaser-scanner erzeugen häufig solche Punktwolken. Algorithmen müssen in unstrukturierten Daten immer alle Daten durchlaufen, was sehr zeitaufwändig ist.[11]

- **Registriert:**

Die Punktwolkenregistrierung findet oftmals Anwendung in den Bereichen autonomes Fahren, Bewegungsschätzung und 3D-Rekonstruktion, Objekterkennung, simultane Lokalisierung und Kartierung (SLAM) und medizinische Bildgebung. Bei registrierten Punktwolken verwenden mindestens zwei Punktwolken ein gemeinsames Bezugssystem. Das Georeferenzieren ähnelt dem Registrieren.[13]

- **Geordnet:**

Sowohl in organisierten als auch in nicht organisierten Punktwolken werden drei Koordinatenwerte für jeden Punkt bereitgestellt. Das Speicherlayout von organisierten Punktwolken ist das eines 2D-Arrays. Das Speicherlayout der Punkte hängt dabei vom räumlichen Layout ab, das durch die XYZ-Werte dargestellt wird.[14]

In einer Studie mit dem Titel "Massive point cloud data management: design, implementation and execution of a point cloud benchmark" wurden verschiedene Verwaltungsmethoden verglichen. Das Endrésümé der Arbeit zeigt, dass es keine Lösung gibt, die für alle Probleme optimal ist. Sie kommen zum Schluss, dass alle ihre getesteten Varianten (PostgreSQL flat table und blocks, Oracle flat table, blocks und Exadata, MonetDB, LA-Stools) Vor- und Nachteile haben und bei jedem Anwendungsfall individuell entschieden werden muss. Diese Arbeit beinhaltet vorwiegend Vergleiche im Initialisieren, Abfragen und Speichern der Daten.[15]

Da die grossen Punktwolken nicht effizient in den Arbeitsspeicher geladen werden können, werden die Dateien für die Darstellung häufig in speziell strukturierte Octree-Formate

überführt. Die Octreestruktur wird z.B. zum Laden und Anzeigen der riesigen Datenmen- gen gebraucht. Damit sind kleine und bis zu terabyte-grosse Punktdatensätze verlustfrei indexiert. Es gibt Unterschiede zwischen den einzelnen Octree-Formaten, im Grundprin- zip sind sie aber gleich. Sie haben immer eine Datei, die die Metadaten hält, eine mit dem Beschrieb der hierarchischen Struktur der binären Punktdaten und die binären Punktda- ten selber. Die Punktwolken werden dabei in Voxels abgelegt, wobei es mindestens einen Wurzelvoxel gibt. Jeder Voxel kann bis zu acht Childs besitzen. Es gibt keine maximale Auflösung, sondern es wird so häufig geteilt, bis es keine weiteren Punkte mehr gibt. Im Gegensatz zum Raster-Kachelungsverfahren werden die Kacheln hier nicht ausgetauscht, sondern mit den Punkten der nächsten Kachel ergänzt.[16] Damit kann die Punktdichte je nach Zoomstufe und Entfernung variieren und dadurch das Laden von Teildaten in den Arbeitsspeicher ermöglicht werden (Abbildung I.2).

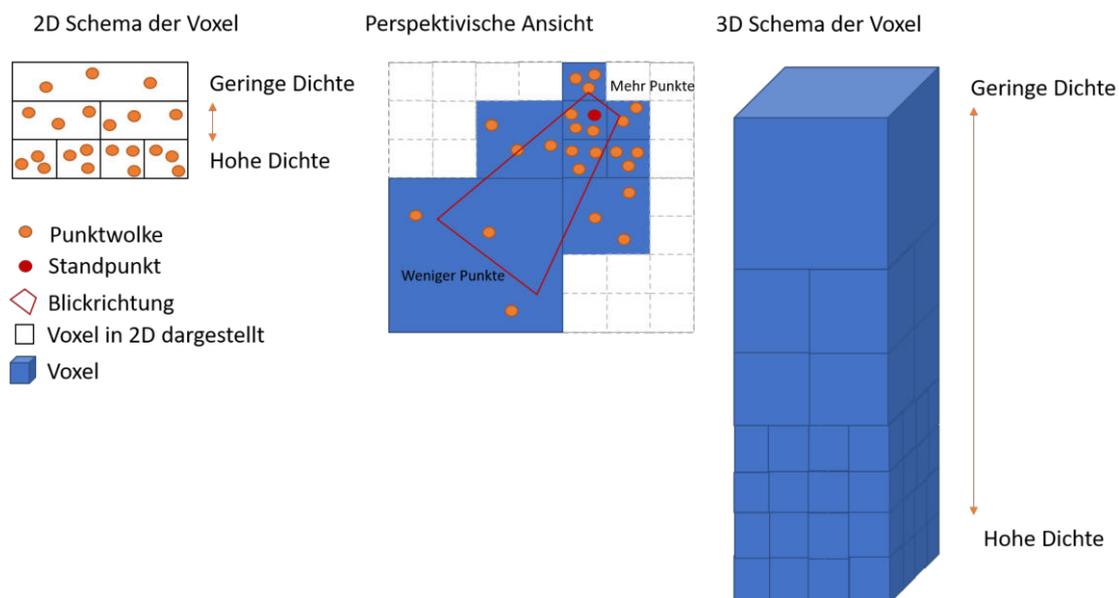


Abbildung I.2: Visualisierung der Octree Struktur

Weiter gibt es Datenbankerweiterungen für Oracle und PostgreSQL, um die Punktwol- ken effizient zu speichern und darauf zuzugreifen. Aber wie auch bei der Verwaltung von Rasterdaten empfiehlt es sich nur eine Enterprise-Geodatenbank zu hosten, wenn viele Benutzer mit Schreibrechten Zugriff auf die Daten benötigen. Ansonsten wird ein herkömmlicher ordnerbasierter Ansatz empfohlen.

Weitere Informationen befinden sich im Anhang 1.2 und im Kapitel I.3.5.

2.4 Punktwolken Verarbeitung

Bei der Verarbeitung von Punktwolken kommt man kaum um das C++ basierte Tool PDAL. Mit PDAL kann man Pipelines zur Bearbeitung und Umwandlung von Punktwolken definieren. Diese Pipelines werden vielseitig in Programmen eingesetzt und können mit wenig Aufwand über eine JSON-Datei erstellt werden. Zu diesem Tool gibt es zudem ein Python Import und viel aktuelle und hilfreiche Dokumentationen zur Unterstützung.

Auch PCL ist ein weitverbreitetes Tool um Punktwolken zu bearbeiten. Es zeichnet sich mit vielen Funktionen im Bereich Objekterkennung aus. Die Funktionen können aber nur im eigenen PCD Format ausgeführt werden. Ein eigener Konvertiervorgang wird nicht angeboten. Auch dieses Tool basiert auf C++, aber es bietet keine weiteren Abstraktions levels wie PDAL über die JSON Pipelines. Zur Verwendung müssen eigene C++ Dateien geschrieben werden.

Mit Grammatiken und mathematischen Spezialisierungen kann man auch aus Punktwolken Vektordaten ableiten und so zum Beispiel Grundrisse, Fassaden, Dachformen und mehr erkennen.[2] Zusätzliche Informationen sind dem Anhang 1.3 zu entnehmen. Die Evaluation dazu wird in Kapitel I.3.6 beschrieben.

3 Evaluation

3.1 Kriterien

In diesem Abschnitt werden Entscheidungen der gewählten Tools dokumentiert. Die Entscheidungen können schnell und einfach aus den Y-Approaches rausgelesen werden. Bei tieferem Interesse wurden weitere Informationen zu den Entscheidungen in Fliesstext und Tabellenform notiert.

Das Folgende ist ein Y-Approach Beispiel:

Y-Approach

Beispiel: *Im Bereich Entscheidungsdokumentation haben wir angesichts der guten Verständlichkeit entschieden, Y-Approaches zu verwenden, um eine einheitliche Struktur zu erreichen. Wir vernachlässigen dabei die beschränkte Argumentationsfreiheit und akzeptieren die einheitliche Dokumentation.*

3.2 Evaluation der iPad Apps

Da es bereits mehrere Apps für den Gebrauch des LiDAR-Sensors des iPad Pro gibt, musste festgelegt werden, mit welchen dieser Apps das iPad in den Gerätevergleich starten sollte.

Kriterium	SiteScape	3d Scanner App	Scaniverse	Polycam	Pix4Dcatch
Kostenlos	Nein	Nein	16.00 CHF/Jahr	8.50 CHF/Monat	Abodienst
Funktioniert nur mit LiDAR-Sensor	Ja	Ja	Ja	Ja	Nein

Tabelle I.1: Evaluation Apps

Wir haben uns für alle Apps ausser Pix4Dcatch entschieden, weil dieses die selben Nachteile wie die fotogrammetrischen Verfahren aufweist. Laut anderen Testern lieferte es zwar gute Resultate, verarbeitet aber nur die Bilddaten auf einem externen Server zu einer Punktwolke. Die App kann also auch gut ohne LiDAR-Sensor verwendet werden, da die Punktwolke ausschliesslich über die Bilder berechnet wird. Der einzige Vorteil bei der Verwendung mit LiDAR-Sensor ist, dass die App Rückmeldungen zur Abdeckung geben kann. Leider konnten wir auch bei den anderen Apps nicht feststellen, wie viel über die Bilder ausgewertet wird und wie viel der LiDAR-Sensor in der Punktwolkenberechnung verwendet wird. Sicher ist aber, dass die Apps nur mit LiDAR-Sensor verwendet werden können und die Prozessierung sofort erfolgt.

Y-Approach

iPad Apps: *Im Bereich Scanner-Apps haben wir angesichts der Verfahrensoptimierung entschieden, SiteScape, 3d Scanner App, Scaniverse und Polycam zu testen, um zu erzwingen, dass der LiDAR-Sensor mitverwendet wird. Wir vernachlässigen dabei, dass Pix4Dcatch einen guten Ruf hat und akzeptieren, dass auch die gewählten Apps die Messungen mit fotogrammetrischer Auswertung ergänzen werden.*

3.3 Evaluation Punktwolke vs Meshes

Kriterium	Meshes	Punktwolke
Export aus Laserscanner	Nein	Ja
Aufbereitung	Ja für gute Genauigkeit	Nein
Aktuell	Ja	Ja
Einfaches Mergen	Ja	Ja
Einfaches Verdichten	Nein	Ja

Tabelle I.2: Punktwolken vs Meshes

Grundsätzlich könnte man die Scannerdaten als Meshmodell verwalten und speichern. Die andere Möglichkeit wäre die Verwaltung der rohen Punktdaten. Ein Vorteil der Meshes ist, dass sie eine noch bessere und einfachere Interpretation der Daten erlauben. Der Nachteil ist der Aufwand, der für die genaue Erarbeitung der Modelle benötigt wird, da die Dreiecksvermaschung ohne explizite Bruchkantendefinition weiche Kanten ergibt. Die rohen Punktwolken bieten die Möglichkeit, die Daten ohne Vorverarbeitung anzusehen. Die Daten müssen also weder aufgearbeitet noch prozessiert werden. Der Nachteil ist aber, dass zwischen den Punkten die Farbwerte verloren gehen und die Interpretation der Punkte mit grösseren Punktabständen schwieriger ist.

Wir haben uns trotz den Nachteilen für die Punktwolke entschieden, da die Daten so schnell dargestellt werden können. Zudem ist unser Ziel ein Prototyp einer Web-Applikation, die in Zukunft mehrere Datensätze vereinen und durch mehrere Messungen vom gleichen Objekt die Punkte verdichten sollte. Dies wird mit Punktwolken einfacher gehen, da die Vorverarbeitung zum Vereinen wegfällt und das Verdichten möglich macht.

Die Meshes kann der Benutzer aber selber aus den downloadbaren Punktwolken erarbeiten. Damit ist der Benutzer frei, mit der gewünschten Punktdichte die Meshes auf das aktuelle Problem passend zu berechnen, ohne dass dabei Rohdaten verloren gehen.

Y-Approach

Meshes vs Punktwolken: Im Bereich Meshes vs Puntwolken haben wir angesichts der Vision des Vereinen und Verdichtens der Daten entschieden, Punktwolken zu verwalten, um eine einfache Integration bei mehreren Messungen offen zu halten. Wir vernachlässigen dabei, dass die Interpretation schwerer ist und akzeptieren, dass die Daten je nach Anwendungszweck lokal in Meshes aufbereitet werden müssen.

3.4 Evaluation von Punktwolken Viewer

Es wurden einige webbasierte Opensource Punktwolken-Viewer genauer betrachtet. Die Evaluation zeigt, dass sich eine Eigenentwicklung nicht lohnen würde, da es bereits gute Viewer mit umfangreicher Funktionalität gibt. Viele dieser Viewer basieren auf den gleichen Grundlibraries wie ThreeJS, D3 und weiteren, die auch in einer Eigenentwicklung eingesetzt würden und sehr solide erscheinen. Folgende Tabelle zeigt einen schnellen Überblick über die relevantesten Viewer.

Kriterium	Potree	Cesium	WebODM	iTowns	Plasio	AHN point-cloud viewer
Ansicht einstellen	Ja	Ja	Ja	Ja	Ja	Ja
Punktwolke Download	Nein	Nein	Ja	Nein	Nein	Nein
Messen	Ja	Ja	Ja	Ja	Ja	Ja
Login	Nein	Nein	Ja	Nein	Nein	Nein
Breite Browser Unterstützung	Ja	Ja	Nein	Nein	Nein	Nein
Einfache Bedienbarkeit	Ja	Ja	Nein	Ja	Ja	Ja
Verbreitet	++	++	+	-	-	-
Unterhalten	Ja	Ja	Ja	Ja	Nein	Nein
Opensource	Ja	teils	Ja	Ja	Ja	Ja

Tabelle I.3: Vergleichstabelle der Punktwolken Viewer

Potree ist ein auf WebGL basierender Punktwolken Viewer, der schon viele Möglichkeiten zur Visualisierung mit sich bringt. Er nutzt bereits CesiumJS, ThreeJS, D3 und weitere Libraries. Zusammen mit CesiumJS bietet er am meisten Funktionalitäten, die für unsere Bedürfnisse passen. Deshalb wurden diese noch genauer untersucht.

Kriterium	Potree	Cesium	Gewichtung
Gut dokumentiert	4 - 0.6	8 - 1.2	15%
User Interface (für Visualisierung)	9 - 0.9	5 - 0.5	10%
User Interface (für Messungen)	8 - 0.8	9 - 0.9	10%
Optionen für den User	7 - 2.1	5 - 1.5	30%
Code Beispiele	8 - 1.6	7 - 1.4	20%
Navigation	9 - 0.9	6 - 0.6	10%
Einfache Einbindung	10 - 0.5	7 - 0.35	5%
Total:	55 - 7.4	47 - 6.45	100%

Tabelle I.4: Vergleichstabelle von Potree vs Cesium

In Abbildung I.3 befindet sich die Visualisierung der Messung der Eisenplastik im Potree-Viewer. Abbildung I.4 zeigt eine Punktwolke mit Cesium auf der Weltkugel.

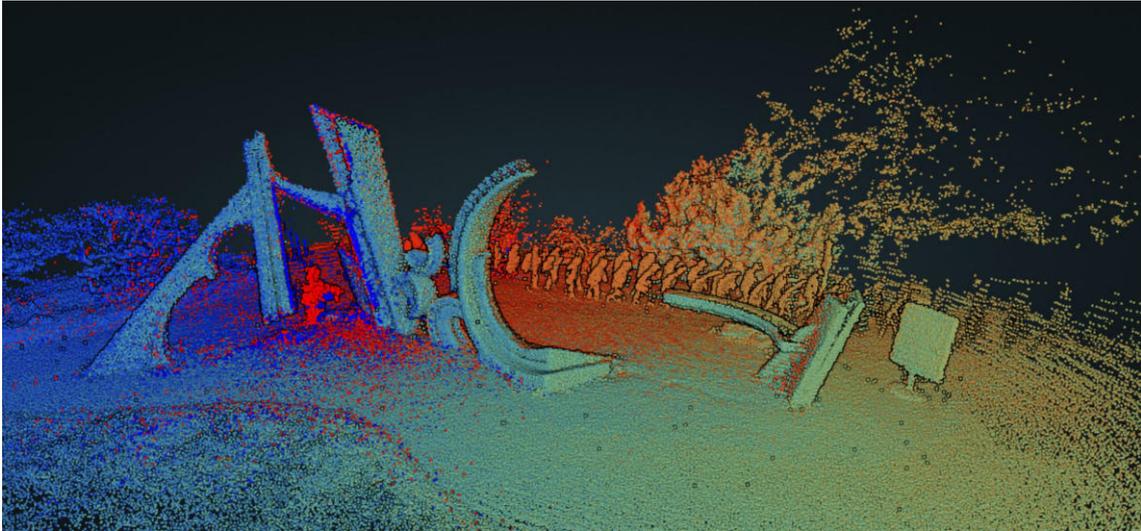


Abbildung I.3: Eisenplastik OST im Potree visualisiert



Abbildung I.4: Eisenplastik OST im Cesium visualisiert

Schlussendlich hat sich Potree durchsetzen können, da die Möglichkeiten, der Aufbau, das Design und die Benutzerfreundlichkeit überzeugen, obwohl alles ziemlich statisch ist. Pro Punktwolke besteht eine statische HTML Page. Bei einem Tool, in welchem der User Punktwolken hochladen sollte, müssten solche Files dynamisch generiert werden können. Auch können nur Punktwolken mit projizierten Koordinatensystemen angezeigt werden. Für die Entwicklung wird ausserdem WebODM relevant sein. WebODM bindet zur 3D-

Visualisierung Potree ein und bietet eine 2D-Ansicht, sowie einen Downloadknopf und einen Share-Button an. Der Grund, wieso wir nicht an WebODM anknüpfen, ist dass noch viel zu viel Fotogrammetrie-Overhead implementiert wurde und das Tool viel zu schwergewichtig für unsere Anforderungen ist.

Y-Approach

Punktwolken Viewer: *Im Bereich Viewer haben wir angesichts der bestehenden Funktionalitäten und der weiten Verbreitung entschieden, Potree zu verwenden, um einen soliden und einfach bedienbaren Viewer anzubieten. Wir vernachlässigen dabei, dass Potree nicht vollumfänglich dokumentiert ist und akzeptieren, dass nur projizierte Koordinatensysteme unterstützt werden.*

3.5 Evaluation von Punktwolken Verwaltung

Kriterium	Entwine (EPT)	Potree	LAZ	PLY	e57	PostgreSQL mit pg-Pointcloud
Pdal lesen, schreiben	Ja eingeschränkt	Nein	Ja	Ja	Ja	Ja
Octree/Patches	Ja	Ja	Nein	Nein	Nein	Ja
Verbreitet	+	+	++	+	+	-
Aktuell	Ja	Ja	Ja	Ja	Ja	Ja
Speicher-Grösse*1	27.4 MB	165 MB	23.8 MB	394.8 MB	87.8 MB	6.25 MB*2
Export iPad*3	Nein	Nein	Ja, App abhängig	Ja, App abhängig	Ja, App abhängig	Ja*4
Export ZEB Revo-RT*3	Nein	Nein	Nein	Nein	Ja	Ja*4
Export BLK2GO*3	Nein	Nein	Nein	Nein	Nein	Nein

Tabelle I.5: Vergleichstabelle der Datenverwaltungsmöglichkeiten

*¹File: 26935underline12355

*² Es wurde nicht extra eine Datenbank aufgesetzt und getestet, sondern berechnet:[15]
384.65 MB LAZ braucht in PostgreSQL mit Batch 101.77 MB Daraus abgeleitet: $101.77 \text{ MB} / 384.65 \text{ MB} * 101.77 \text{ MB} = 6.25 \text{ MB}$ *³ Export direkt vom Gerät. Mit entsprechender Software können diese Formate dann natürlich umkonvertiert werden.

*⁴ PDAL lesbare Formate importierbar (Vorteil wenn sie geordnet sind)

Für diese Arbeit wurden drei Speichervarianten verglichen. Neben der einfachen Fileablage (LAZ, PLY und e57) wurden auch das Octreeform (EPT und Potree) und die DB Varianten (Oracle und PostgreSQL) analysiert.

Bei den Datenbanken haben wir schnell festgestellt, dass sich nur eine optimierte Datenbank effizient einsetzen lässt. Dabei ist die Auswahl auf Oracle und PostgreSQL mit entsprechender Erweiterung gesunken. Die Oracle Datenbank wurde dabei nicht weiter berücksichtigt, da wir ein Opensource-Programm aufbauen wollen. Ein wichtiger Punkt bei der PostgreSQL-DB mit pgPointCloud ist, dass sich die Patches nicht überschneiden. Auch die Vorteile der PostgreSQL-DB mit pgPointCloud, wie die Integration mit andern geografischen Daten, haben nicht ausgereicht um den Overhead, der mit der Initialisierung und dem Unterhalt gekommen wäre, zu rechtfertigen. Da unsere Punktwolken nicht von vielen Benutzern zeitgleich verändert werden und die zu verwaltenden Daten sich überschneiden könnten, wurde diese Option für unser Problem ausgeschlossen.[17]

Die Datentypen der einfachen Fileablage unterscheiden sich in Speichergrosse und in ihrem Aufbau. Da wir mit den Handlaser Scanner und dem SLAM-Algorithmus zum Teil unstrukturierte Daten erfassen, bringt uns das e57 gegenüber dem LAZ-Dateiformat nicht immer einen Vorteil, braucht aber mehr Speicherplatz. Den Kandidaten PLY werden wir nicht verwenden, weil er eine enorme Speichergrosse im Vergleich zum LAZ aufweist. Somit bleibt unter den dateibasiereten Varianten das weitverbreitete und häufig eingesetzte, gut komprimierte Format LAZ.

Das LAZ-Dateiformat hat aber einen grossen Nachteil: Es eignet sich nicht für die Darstellung von Punktwolken. Als Lösung zum Darstellungsproblem wurden zwei Octree-Strukturen genauer angeschaut. Zum einen gibt es die für Potree entwickelte Struktur, die bestens mit dem Potree-Viewer kompatibel ist. Da aber Potree auch EPT-LAS und

EPT-LAZ unterstützt, ist EPT eine auch für unseren Einsatzzweck gute Alternative zum Potree-Format, zumal es weiter verbreitet ist. Durch seine Verbreitung ist es in Zukunft einfacher mit anderen Thirdparty-Tools verwendbar. Somit werden wir eine Kombination von EPT-LAZ für die weboptimierte Darstellung und LAZ für die sichere Verwahrung verwenden.

Y-Approach

Punktwolken Verwaltung: Im Bereich Datenverwaltung haben wir angesichts der geringen Speichergrösse und der Optimierung für die Darstellung entschieden, EPT-LAZ in Kombination mit LAZ zu verwenden, um eine effiziente Datenverwaltung zu erreichen. Wir vernachlässigen dabei, dass diese Formate nicht zwingend aus den Geräten gespielt werden können und akzeptieren, dass eine Vorprozessierung von LAZ in EPT-LAZ nötig ist, um die Daten darzustellen.

3.6 Evaluation von Punktwolken Verarbeitung

Das Ziel der Verarbeitung der Punktwolken ist ein Commandline Tool, welches mindestens ein 2D-Abbild von Punktwolken generieren kann. Dazu wurden verschiedene Funktionen und Algorithmen von PDAL und PCL getestet. Leider sind viele der Funktionen auf airbased Laserscanning ausgelegt und liefern für unsere Handlaserscanner noch keine brauchbaren Ergebnisse. Die Funktionen, die uns sinnvoll erschienen, waren die Bodenerkennung, das Minimieren des Rauschens oder auch das Erkennen von Strukturen und Objekten.

Kriterium	PDAL	PCL	LASstool
Daten lesen LAZ	Ja	Nein	Ja
2D Raster generierbar	Ja	Ja	Ja
Rauschen eliminieren	Ja	Ja	Ja
Strukturen erkennen	Ja	Ja	Nein
Verbreitet	++	+	++
Unterhalten	Aktuell	Aktuell	Aktuell
Opensource	Ja	Ja	teils

Tabelle I.6: Vergleichstabelle der Punktwolkenverarbeitung

Da die LAStools nur teilweise Opensource sind und nichts bieten, was PDAL oder PCL nicht auch bieten, wurde dies bereits ausgeschlossen und der Vergleich zwischen PDAL und PCD fortgesetzt.

Funktion	PDAL	PCL	Gewichtung
Plattform unabhängig und einfach installierbar	10 - 1	3 - 0.3	10%
Einfach anwendbar	9 - 1.35	4 - 0.6	15%
Bodenerkennung für 2D Ableitung	2 - 0.4	2 - 0.4	20%
Objekterkennung für 2D Ableitung	2 - 0.4	4 - 0.8	20%
DOM Export für 2D Ableitung	10 - 1.5	0 - 0	15%
In Python integrierbar	10 - 2	5 - 1	20%
Total:	43 - 6.65	18 - 3.1	100%

Tabelle I.7: Vergleichstabelle der PDAL PCL

PCL klingt sehr vielversprechend, was die Funktionen der Objekterkennung angeht, aber leider funktioniert dies nur mit ihrem eigenen PCD-Dateiformat. Auch die Implementation von PDAL ist eine Spur angenehmer, da es dazu ein Python Package gibt. Der Nutzen von PDAL überwiegt den von PCL deshalb. Es ist aber nicht auszuschliessen, dass bei einer Weiterentwicklung auch PCL in die Pipeline aufgenommen werden könnte, um über dessen Funktionen noch bessere Grundlagen für die Berechnung eines 2D-Plans zu schaffen.

Y-Approach

2D Konvertierung: *Im Bereich 2D-Umwandlung haben wir angesichts der Funktionen entschieden, PDAL zu verwenden, um eine 2D-Ansicht zu erzeugen. Wir vernachlässigen dabei, dass PCL viele Funktionen zur Objekterkennung bietet und akzeptieren, dass mit PDAL nur ein DOM GeoTIFF Export der Punktwolke möglich ist.*

Neben der 2D ableitbaren Ansicht haben wir entschieden, auch die Aufbereitung von LAZ in ein Octree-Format als Verarbeitungsschritt anzusehen. Dabei wurden drei Programme verglichen. Zum einen Entwine, der Potree Converter und auch untwine, welches bei QGIS im Einsatz ist. PDAL kann kein EPT Format erzeugen. Es kann es lediglich lesen, verarbeiten und wieder als EPT speichern oder die Daten in ein anderes Format umwandeln, aber nicht ein anderes Format lesen und EPT schreiben.

Kriterium	Entwine	Untwine	Potree Converter
Daten lesen LAZ	Ja	Ja	Ja
Datenumwandlung in Octree	Ja	Ja	Ja
Datenverbreitung	++	++	+
Toolverbreitung	++	+	+
Installierbarkeit	-	++	++
Unterhalten	Aktuell	Aktuell	Aktuell
Opensource	Ja	Ja	Ja

Tabelle I.8: Vergleichstabelle der Konvertierungstools

Da das EPT-Entwine Format durch seine Bekanntheit und vielseitige Einsetzbarkeit bereits im Speicherungsvergleich überzeugt, ist der Potree Converter weggefallen, da er eine leicht andere Struktur speziell für Potree erstellt. Dabei muss aus dem vorherigen Entscheid aufgegriffen werden, dass Potree bereits sehr gut mit EPT-LAZ verwendbar ist und die Pipeline auch unabhängig von unserer Weblösung eingesetzt werden kann. Zwischen entwine und untwine war die Installierbarkeit ausschlaggebend. Als Zusatz ist dies auch für die Fachleute, die QGIS verwenden im Einsatz und somit bestens kompatibel.

Dadurch ist die manuelle Bearbeitung, bei welcher Objekte entfernt werden können, nicht inbegriffen. Diese wurde absichtlich nicht mitaufgenommen, da das Tool CloudCompare dies bereits gut umsetzt und dieser Schritt besser mit einem umfangreicheren Tool gemacht wird. Die Benutzbarkeit mit einem Commandline-Interface wäre diesbezüglich eher schwierig.

Y-Approach

Ansichtsoptimierung: Im Bereich Aufbereitung der Punktwolken für eine optimierte Darstellung im Web haben wir angesichts der einfachen Installierbarkeit entschieden, *untwine* einzusetzen, um die Usability zu gewährleisten. Wir vernachlässigen dabei, dass der *Potree Converter* besser mit dem *Potree-Viewer* zusammenpasst und akzeptieren, dass wir nicht den offiziellen *EPT-Converter* *Entwine* einsetzen.

3.7 Evaluation Technologie Punktwolkenbrowser

Da das Programmieren in Python schon im Auftrag gewünscht wurde, haben wir Flask, Bottle und Django als bekannteste Python Frameworks gegeneinander antreten lassen.

Funktion	Django	Flask	Bottle
Packages	built-in	external	-
Frameworktyp	full-featured Model View Controller	microframework	microframework

Tabelle I.9: Vergleichstabelle der Backendframeworks

Kriterium	Django	Flask	Bottle	Gewichtung
Community	8 - 1.6	5 - 1	3 - 0.6	20%
ORM (object-relational mapping)	10 - 1.5	1 - 0.15	1 - 0.15	15%
Grosse Projekte möglich	10 - 1.5	4 - 0.6	2 - 0.3	15%
Funktionalität* ¹	9 - 2.25	5 - 1.25	1 - 0.25	25%
Erfahrung im Team	6 - 1.5	1 - 0.25	1 - 0.25	25%
Total:	43 - 8.35	16 - 3.25	8 - 1.55	100%

Tabelle I.10: Vergleichstabelle der Backendframeworks

*¹ Wichtig für uns, dass schon viel dabei ist.

Im Backend haben wir uns für das grössere Django-Framework entschieden, da wir damit bereits Erfahrungen haben und es schon viele built-in features besitzt. Wir werden zwar

den Prototypen noch nicht so umfangreich umsetzen können, sodass Flask zu klein wäre, aber da die Weiterentwicklung des Prototypen vorgesehen ist, scheint uns Django die passendere Variante zu sein.

Y-Approach

Backend Technologie: *Im Bereich Backend haben wir angesichts der Erfahrungen im Team entschieden, Django einzusetzen, um den engen Zeitplan einhalten zu können. Wir vernachlässigen dabei, dass Django von der Grösse des Prototypen noch nicht zwingend verwendet werden müsste und akzeptieren, dass wir auch Flask verwenden könnten.*

Da wir Benutzerdaten und einige Metadaten, sowie den Ablageort der einzelnen Dateien verwalten müssen, haben wir uns dazu entschieden, eine Datenbank einzubauen. Dabei wurde auf eine PostgreSQL-Datenbank gesetzt, weil diese im Gegensatz zu inMemory-Lösungen beim Einsatz der Applikation gut auf einen anderen Server verlagert werden kann. Es wäre auch möglich gewesen, eine noSQL-Datenbank zu verwenden. Aber da das Suchen und Filtern eine zentrale Rolle spielt, wurde auf PostgreSQL als Opensource-Datenbank gesetzt.

Was wir aber nicht vorgesehen haben, ist eine verteilte Datenbank. Da nur verhältnismässig wenig Daten darin gehalten werden, reicht eine zentrale Datenbank zurzeit aus.

Y-Approach

Beispiel: *Im Bereich Datenbank haben wir angesichts der guten Entkopplung entschieden, PostgreSQL zu verwenden, um eine einfache und effiziente Suche und Filtrierung zu ermöglichen. Wir vernachlässigen dabei, dass eine eher grosse Datenbank gewählt wurde und akzeptieren die gute Filtermöglichkeit.*

Im Frontend wurden die Opensource Frameworks React und Vue verglichen. Die Beiden sind sehr weit verbreitet, da sie über ihren virtuellen DOM nur die betroffenen Teile und nicht die komplette Seite neu laden. Sie bieten auch beide einen auf Komponenten basierenden UI-Aufbau, der das Trennen und Wiederverwenden von Code vereinfacht. Sie

unterscheiden sich aber in der Umsetzung des Routings und des Satusmanagement. In Vue ist dies bereits im offiziellen Umfang inbegriffen, während React dazu third party librerys verwendet.[18]

Kriterium	React	Vue	Gewichtung
Community	10 - 2	8 - 1.6	20%
Aktuell	9 - 2.7	5 - 1	30%
Erfahrung	6 - 1.8	1 - 0.3	30%
Lernkurve	4 - 0.4	8 - 0.8	10%
Weniger Paket basiert (third party Packages)	5 - 0.5	9 - 0.9	10%
Total:	34 - 7.4	31 - 4.6	100%

Tabelle I.11: Vergleichstabelle der Frontendframeworks

Eigentlich könnten beide Tools verwendet werden. Da wir aber schon mehr Erfahrung mit React haben und die Arbeit zeitlich begrenzt ist, werden wir auf React mit Hooks und ohne Redux Store setzen. Da wir für den Prototypen mit nicht übermässig vielen Daten rechnen, gehen wir davon aus, dass dies ausreicht. Den Redux Store könnte man aber zu einem späteren Zeitpunkt hinzufügen, sollte dieser durch Erweiterungen erforderlich werden.

Y-Approach

Frontend Technologie: *Im Bereich Frontend haben wir angesichts der Erfahrungen im Team entschieden, React einzusetzen, um den engen Zeitplan einhalten zu können. Wir vernachlässigen dabei, dass Vue den Anforderungen ebenfalls gerecht worden wäre und akzeptieren, dass wir Variablen über Hooks verwalten müssen.*

Zudem haben wir entschieden, React direkt im Django zu entwickeln, um uns durch hohe Kopplung den Umgang mit temporären Files zu erleichtern.

Y-Approach

Trennung: *Im Bereich Codebasis haben wir uns angesichts des einfacheren Handlings mit Files dazu entschieden, React direkt im Django zu entwickeln, um Programmieroverhead zu sparen. Wir vernachlässigen dabei die Trennung der Verantwortlichkeiten und akzeptieren dass Front- und Backend Entwickler eng miteinander zusammenarbeiten müssen.*

Wir haben uns entschieden, die Kommunikation über HTTP einzurichten:

Kriterium	REST	GraphQL	GRPC
Komplexe Abfragen	-	++	+
Server definierte Antworten	Ja	Nein	Ja
Über HTTP	Ja	Ja	Ja
Status Codes	Ja	Nein	Ja
JSON	Ja	Ja	Nein

Tabelle I.12: Vergleichstabelle der Kommunikation

Da wir keine komplizierten Datenstrukturen zu übermitteln haben und diese gerne in einem menschenlesbaren Format ans Frontend übermittelt wollen, haben wir uns für eine RESTfull-Schnittstelle entschieden.

Eine FTP Lösung für die Files wurde vermieden, weil die Erfahrungen fehlten.

Y-Approach

Beispiel: *Im Bereich Kommunikation haben wir angesichts der Erfahrung entschieden, REST zu verwenden, um die Daten einfach zu übermitteln. Wir vernachlässigen dabei die vielen anderen Möglichkeiten und akzeptieren die serverseitige Festlegung der Kommunikation.*

3.8 Evaluation Technologie Punktwolkenpipeline

Um die bestehenden Tools in einem Commandlinetool zu vereinen, war die Programmiersprache Python vorgegeben. Weiter haben wir entschieden, das Tool auf PiPy zu publizieren und sowohl als Package als auch als Commandlinetool anzubieten. Dadurch konnten wir das Management der Punktwolken auch in unserem Pointcloudbrowser auf dieses Tool auslagern.

Y-Approach

Punktwolkenpipeline: Im Bereich Pipeline haben wir angesichts der guten Installation entschieden, das Tool sowohl als Package als auch als Commandlinetool via PyPi zu veröffentlichen, um eine flexible Einsatzmöglichkeit zu bieten. Wir vernachlässigen dabei die Publikationsaufwände und akzeptieren, dass jeder dieses Tool verwenden kann.

4 Erfahrungsbericht Handlaserscanner

4.1 Messobjekte

Als Messobjekte standen eine Mauer der Ruine-Neuenburg in Weinfelden TG, die Villa Aurum in Rapperswil und eine Eisenplasik an der OST in Rapperswil zur Verfügung. Die drei Messobjekte repräsentieren unser Zielpublikum, bestehend aus Archäologen, Denkmalpfleger und OSM-Mapper, optimal.

Die Mauer der Ruine Neuenburg in Abbildung I.5 ermöglichte, in Zusammenarbeit mit dem Amt für Archäologie des Kanton Thurgau, den Vergleich verschiedener bereits einsetzerproben Messmethoden an einem realen Objekt. Die Messungen wurden dabei innerhalb von 24 h erfasst und sollten deshalb einen gut vergleichbaren Stand zeigen.



Abbildung I.5: Mauer der Ruine Neuenburg in Weinfeld

Die Eisenplastikskulptur in der Abbildung I.6 ist dank ihrer komplizierten Form, den Löchern, Einbuchtungen und Fortsätzen ein gutes Beispiel für den Einsatz von Handlaser-scannern in der Denkmalpflege und Archäologie und somit ideal für die Beurteilung der Scanner. Zudem bietet sie mit ihrer rostigen Oberfläche eine Alternative zu den beiden anderen Objekten.

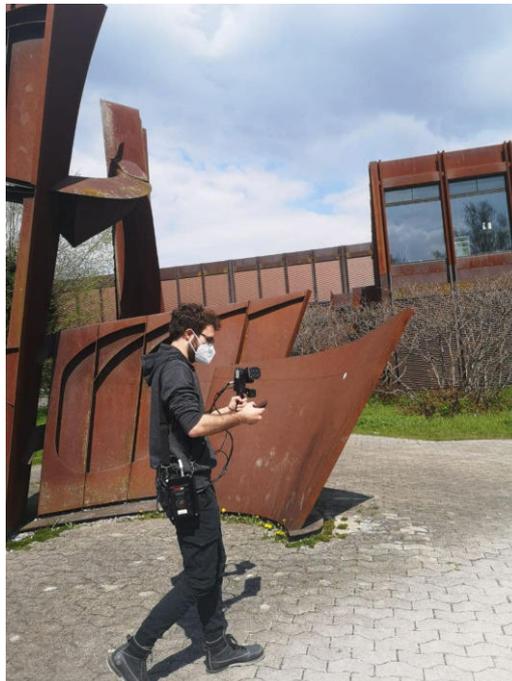


Abbildung I.6: Messung der Eisenplastik OST mit dem GeoSLAM ZEB-Revo RT

Die Villa Aurum in der Abbildung 1.7 kann sowohl für die Denkmalpflege, als auch den Hobby-Mapper interessant sein, der damit zum Beispiel die Open-Source Community unterstützen möchte. Bei dieser Messung haben wir die Resultate mit eigens gemessenen Referenzpunkten verglichen, um die Handlaser-scanner nicht nur intern zu vergleichen. In Abbildung 1.8 ist die dazu verwendete Leica iCON Robot 70 Totalstation in Aktion zu sehen. Die Referenzpunkte wurden farblich hervorgehoben und auch räumlich von der Oberfläche abgehoben, um sie in der Punktwolke zu erkennen. Messkugeln wurden allerdings keine verwendet, sondern selber entworfene Referenzpunkte an das Objekt geklebt oder gestellt.



Abbildung 1.7: Villa Aurum in Rapperswil mit den Referenzpunkten



Abbildung I.8: Referenzmessung

4.2 Leica BLK2GO

In der Abbildung I.9 ist das sehr kompakte Leica BLK2GO sichtbar. Es besteht lediglich aus einem Griff mit einer Kuppel in welcher sich der rotierende Laser befindet. Auch der Akku und die Speichereinheit befinden sich im Gerät. Für die Visualisierung des Messfortschritts kann zusätzlich eine App installiert werden. Mit der Android-App konnte nach mehreren Minuten keine Verbindung zum BLK2GO hergestellt werden, während es mit einem iOS-Gerät problemlos geklappt hat.



Abbildung I.9: BLK2GO[22]

Beim Start einer Messung wird das Gerät auf einer möglichst horizontalen Oberfläche auf dem dazugehörigen Standfuss platziert. Das BLK2GO kommt mit sehr wenig aus. Ein leuchtender LED-Ring zeigt beispielsweise bei grüner Farbe an, dass das Gerät einsatzbereit ist. Dann kann der Scan mit dem Drücken des einzigen Knopfes gestartet werden. Der Knopf hat je nach Geräte-Status, welcher am LED-Ring abgelesen werden kann, eine andere Funktion. So führt ein langes Drücken während der Messung zu einem zusätzlichen Bild der Umgebung. Geschmackssache ist, ob es vielleicht doch einen Knopf mehr vertragen würde. Nach kurzem Probieren kamen wir bei unseren Messungen allerdings damit klar.

Wo das Design aber klar über den Zweck gestellt wurde, ist beim dazugehörigen Trageelement. Der kleine Sockel, der zum Start der Messung benötigt wird, passt leider nicht hinein und muss separat mitgetragen werden. Dadurch kann dieser leicht verloren oder vergessen gehen.

Die Messung mit dem Gerät ist sehr schnell und angenehm, da das Gerät gut in der Hand liegt und auch vom Gewicht her für längere Messungen gut zu tragen ist. Der Scanbereich ist enorm gross und deckt die Objekte in einem Umkreis von etwa 15 m problemlos ab. In der zweiten Hand kann, sofern es das Gelände zulässt, das Mobilgerät mit der Liveansicht mitgetragen werden. Somit erkennt man sofort, welche Bereiche bereits gescannt wurden und welche noch ergänzt werden müssen.

Um die Punktwolken vom BLK2GO zu exportieren, muss der Leica BLK Data Manager

oder Cyclone verwendet werden. Via USB-C wird das BLK2GO mit einem Rechner verbunden. Dort können die Daten direkt in Cyclone importiert oder mit dem BLK Data Manager im .blk-Format auf das Dateisystem des Rechners exportiert werden. Für die Weiterverwendung wird aber Cyclone benötigt, da es aus dem .blk die Punktwolke in ein e57 exportieren kann. Dieser Schritt dauert extrem lange und benötigt einiges an freien Speicher, damit der Vorgang abgeschlossen werden kann. Schlägt der Export-Prozess fehl, zum Beispiel durch zu wenig Speicherplatz, wurde zum Teil keine Fehlermeldung ausgegeben. Das Resultat war lediglich eine korrupte e57-Datei. Auch der BLK Data Manager hat immer von einem erfolgreichen Download berichtet, obwohl er manchmal nur gerade die Hälfte der Datei exportiert hatte.

Sind die gesamten Scan- und Weiterverarbeitungsschritte abgeschlossen, liegen aber sensationelle Punktwolken vor. Die Punkte liegen dicht nebeneinander und auch die Outliers halten sich bei allen Messungen in Grenzen.

Mit seinem stolzen Anschaffungspreis von ca 50'000 CHF ist es ein teures Vermessungsgerät (persönliche Kommunikation Roger Bräm).

4.3 GeoSLAM ZEB-Revo RT

Der GeoSLAM ZEB Revo-RT ist in einem grossen Koffer verpackt (Abbildung I.10) und wesentlich schwerer als das BLK2GO. Auch sind es mehr Einzelteile, die zuerst zusammengesetzt werden müssen und bei der Messung etwas unhandlich getragen werden. Wie beim BLK2GO kann über ein Mobilgerät die Abdeckung der Messung live mitverfolgt werden. Anders als beim oben beschriebenen Leica-Gerät, braucht es für den GeoSLAM ZEB Revo-RT lediglich ein Gerät, welches sich über das vom Feldcomputer ausgesendete WLAN auf die lokale Seite verbindet.



Abbildung I.10: GeoSLAM ZEB-Revo RT

Nachdem der Feldcomputer, Scanner und die GoPro über die Kabel miteinander verknüpft und die Verbindung mit dem in unserem Fall bereitstehenden Tablet hergestellt wurde, wird der GeoSLAM ZEB Revo-RT auf den Startpunkt gelegt. Die einzelnen Komponenten werden in der richtigen Reihenfolge gestartet und die Messung kann erfolgen.

Da der Griff im Verhältnis zum Scanaufsatz relativ leicht ist, ist der Schwerpunkt zu weit oben, weshalb das Gerät relativ fest gehalten werden muss. Eine längere Messung ist somit ziemlich anstrengend. Der Feldcomputer kann dank einer Schlaufe über die Schulter gehängt werden. Wie auch beim BLK2GO kann in der freien Hand die Messung, welche einen überzeugenden Scanbereich von etwa 15 m hat, live beobachtet werden.

Nach der Messung wird ein PC an das Netz des Feldcomputers gekoppelt und die Daten können als .geoslam oder .e57 exportiert werden. Für die weitere Verarbeitung steht eine Gratisversion des GoeSLAM Hub zur Verfügung, welches den Export in weitere Formate ermöglicht. Soll die Punktwolke eingefärbt werden, ist allerdings eine kostenpflichtige Lizenz erforderlich.

Auch der GeoSLAM ZEB Revo-RT kann mit genauen Punktwolken überzeugen. Allerdings lassen sich die Punktwolken nur mit zusätzlichem Aufwand einfärben. Dabei ist auch der eingeschränkte Bereich der GoPro nicht optimal, da somit nicht alle Punkte eingefärbt werden können.

Auch das GeoSLAM hat einen Anschaffungspreis von ca 50'000 CHF (persönliche Kommunikation Jürg Pulver).

4.4 iPad Pro

Das iPad (Abbildung I.11) ist mit Abstand das günstigste Gerät in diesem Vergleich. Zur Vorbereitung vor der Messung muss lediglich eine passende App installiert werden. Die Bezahlmodelle reichen von kostenlos, Monats- oder Jahresabo bis zu Kosten pro Export. Je nach App ist die visuelle Darstellung des gemessenen Bereichs anders. So zeigt SiteScape schon einzelne Punkte der Punktwolke, während App Scaniverse den nicht gescannten Bereich rot/weiss einfärbt. Die 3d Scanner App und Polycam zeigen ein Polygon-Gitter, aus dem der Detailgrad des gemessenen Bereichs ermittelt werden kann.



Abbildung I.11: Messung mit iPad Pro

Das Problem beim iPad ist die eingeschränkte Reichweite und dass es keine 360-Grad Abdeckung hat, sondern lediglich in eine Richtung misst. Um grössere Objekte zu scannen, muss das iPad hoch und runter geschwenkt werden, um ein möglichst gutes Resultat zu erzielen. Durch den limitierten Bereich dauert auch die Messung deutlich länger als bei den professionellen Geräten.

Einige Apps sind in der Messgrösse beschränkt. So müssen schlussendlich mehrere Messungen des selben Objekts zusammengeführt werden, da die Limite bei einem grösseren

Objekt schnell erreicht wird. Bei anderen Apps stellen grosse Objekte wiederum kein Problem dar. Unterschiedlich sind auch die Einstellungsmöglichkeiten im Bereich Genauigkeit und Erfassungsdistanz. Ansonsten werden in allen Apps verschiedene Exportformate angeboten, welche sehr einfach vom Gerät auf den PC übertragen werden können.

Die Qualität der Punktwolken ist bei kleineren Objekten erstaunlich gut, kann aber nicht mit den Profigeräten mithalten. Mit grossen Messungen hat das iPad allerdings Mühe, was unter anderem an der SLAM Genauigkeit liegt.

Mit dem niedrigsten Anschaffungspreis von gerade einmal 2'200 CHF bleibt sicher auch noch etwas Budget für die eine oder andere kostenpflichtigen App wie Polycam (16.00 CHF/Jahr) oder Scaniverse (8.50 CHF/Monat) übrig.

4.5 Structure from Motion Referenzmessung

Basis der Modellberechnung sind insgesamt 429 Fotos, welche Herr Mayer mit seinem Team des Amtes für Archäologie TG erfasst haben. Die Aufnahmen stammen von einer Yuneec H 520 Drohne mit einer E90 Kamera (1 Inch CMOS-Sensor mit 20 Megapixel Auflösung). Diese Aufnahme dauerte rund 20 Minuten. Während der Befliegung wurde eine 4K-Videoaufnahme mit 60 Bildern pro Sekunde gemacht.

Jedes 48. Bild wurde für die Berechnung verwendet. Die Software AgiSoft Metashape (Version 1.5.3.8469) hat in den Bildern bereinigte 190'976 Verknüpfungspunkte erkannt und berechnete in hoher Qualität bei sanftem Filtermodus und einer Dauer von 2 h 27 min eine detaillierte Punktwolke.

Insgesamt wird der Aufwand für die Aufnahme, Berechnung, Bereinigung und Referenzierung von den erfahrenen Archäologen auf 8 h geschätzt (persönliche Kommunikation Daniel Mayer).

4.6 Gegenüberstellung der Handlaserscanner

Gerät	Farbige Punktwolke	Download Zeit* ¹	PC Software	Exportierbare Formate
Leica BLK2GO	Ja	30 min	BLK Download Manager, Cyclone	b2g (nur über Software: e57)
GeoSLAM ZEB Revo-RT	Schwierig* ²	10 min	Webseite des Feldcomputer, GeoSLAM Hub	geoslam, ply (über Software: LAZ, LAS, E57, PLY, TXT)
SiteScape (iPad Pro)	Ja	3 min	iTunes	FBX, GTF, USDZ, OBJ, STL, PLY oder Message Send/Sketchfab Upload
Scaniverse (iPad Pro)	Ja	5 min	iTunes	ply, e57
3d Scanner App (iPad Pro)	Ja	3 min	iTunes	XYZ Color, PLY, PTS, LAS, LAS Geo-Referenced, e57, PCD, XYZ no Color, XYZ color, space delimited oder Modell Export (siehe Anhang)
Polycam (iPad Pro)	Ja	3 min	iTunes	OBJ, GBL, DAE, USDZ, STL, DAE, DXF, PLY, XYZ, PTS oder Teilen(siehe Anhang)

Tabelle I.13: Gegenüberstellung der Messgeräte

*¹ Unter Download-Zeit verstehen wir die Zeit, bis wir eines dieser Formate: LAZ, LAS, E57, PLY, TXT aus der Messung der Villa Aurum auf unserem Computer haben.

*² Wegen des eingeschränkten Sichtbereichs der GoPro und der Lizenz für GeoSLAM Hub. Die grösste Datei konnte nicht reprozessiert werden.

Dieser Messungsvergleich wurde über alle 3 Messungen hinweg zusammen getragen.

Kriterium	Leica BLK2GO	Geo- SLAM ZEB Revo- RT	Site- Scape (iPad Pro)	3d Scanner App (iPad Pro)	Scani- verse (iPad Pro)	Polycam (iPad Pro)
Aufnahmezeit	sehr schnell	schnell	mittel	mittel	langsam	langsam
Messungen unterteilt	Nein	Nein	Ja (Villa 4 Teile)	Nein	Nein	Nein
Outliers, Ver- schiebungen, Löcher	wenige Outliers	wenige Outliers	einige Outliers	Viele Verschie- bungen und Löcher	Einige Verschie- bungen und Löcher	Viele Verschie- bungen und Löcher
Messbereich	15 m	1 5m	4 m	4 m	4 m	4 m
3d Completion* ⁵	98%	98%	92%	82%	83%	83%
2D Grundriss ermittelbar	sehr ge- nau	sehr ge- nau	genau	genau	genau	eher un- genau
Abweichung auf Villa Fassade* ¹	0.04 m	0.01 m	0.32 m	0.57 m	0.27 m	0.31 m
Abweichung zu Referenz- punkten* ²	0.06 m	0.08 m* ³	0.39 m	1.05 m* ⁴	0.46 m	0.6 m

Tabelle I.14: Vergleichstabelle der Messgeräte

*¹ Die Länge des Vergleiches bezieht sich auf die Länge der Westfassade des Gebäudes mit 10.68 m.

*² Die Punktwolken wurden über drei Punkte zu den Referenzmessungen ausgerichtet. Der Wert ist dabei die maximale Abweichung zwischen Punktwolke und den restlichen Referenzpunkten. *³ Durch schlechte Farben und Punktdichte sind die Punkte kaum zu erkennen und daher geschätzt *⁴ Viele Verschiebungen *⁵ 3d Completion Mittelwert über alle 3 Messungen.

4.7 Handlaserscanner vs Structure from Motion

Dieser Vergleich basiert auf unserer Messung vom 14.04.2021 und der Messung von Herrn Mayer am 15.04.2021.

Kriterium	Structure from Motion	Leica BLK2GO	GeoSLAM ZEB-Revo RT	iPad	
Aufnahmezeit	20 min	5 min	5 min	20 min	
Auswertungszeit* ¹	2h 27 min	30 min	10 min	5 min	
Georeferenzierungsaufwand* ⁴	30-45 min	20 min	1h* ³	Scaniverse Polycam 3d Scanner App SiteScape	20 min 20 min 40 min* ² 40 min* ²
Anzahl Punkte	32'939'936	23'346'161	2'634'453	Scaniverse Polycam 3d Scanner App SiteScape	1'038'461 815'460 2'234'417 18'471'492
Durchschnitt der Punktabstände	0.006	0.014	0.040	Scaniverse Polycam 3d Scanner App SiteScape	0.018 0.025 0.014 0.029
Ausdehnung	X37.60 Y13.87 Z8.15	X72.29 Y45.81 Z28.53	X62.58 Y41.58 Z23.37	Scaniverse Polycam 3d Scanner App SiteScape	X30.48 Y5.67 Z5.44 X31.81 Y12.94 Z5.45 X36.38 Y9.48 Z5.63 X30.98 Y9.44 Z8.12
3D Completion	100%	95%	95%	Scaniverse Polycam 3d Scanner App SiteScape	85% 85% 70% 90%

Tabelle I.15: Vergleichstabelle der Messgeräte

- *¹ Bis die Punktwolke auf dem PC ist.
- *² Da die Messungen unterteilt waren, entstand ein Mehraufwand.
- *³ Zuerst Farbwerte prozessieren.
- *⁴ Handlaserscanner mit durchschnittlich vier Punkten referenziert und Structure from Motion mit 10.

Bei der 3D-Vollständigkeit haben alle Handlaserscanner Nachholbedarf. Da die Mauer bis zu 3.5 m hoch war, haben alle Punktwolken an der Oberkante sehr spärlich bis gar keine Punkte erfasst. Dazu wäre eine Verlängerung des Griffes, um von unten zur Oberkante oder von oben über die Kante zu gelangen, keine schlechte Erfindung.

In den Abbildungen I.12 bis I.18 ist jeweils der gleiche Ausschnitt von den Punktwolken aller Geräten sichtbar.

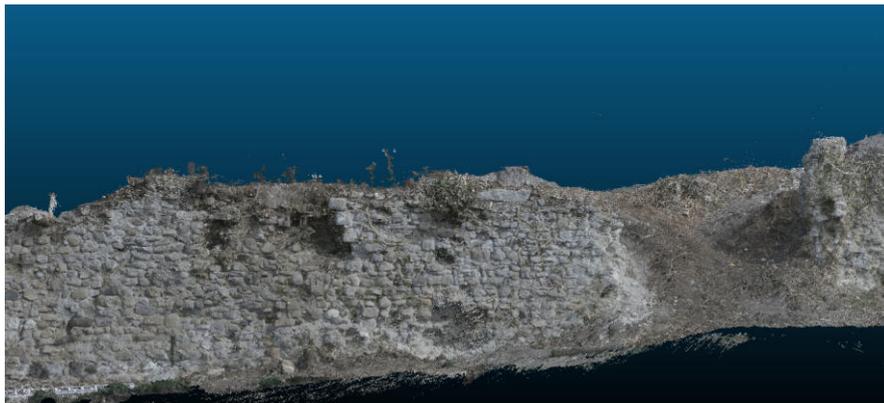


Abbildung I.12: Punktwolke aus Structure from Motion



Abbildung I.13: Punktwolke von Leica BLK2GO

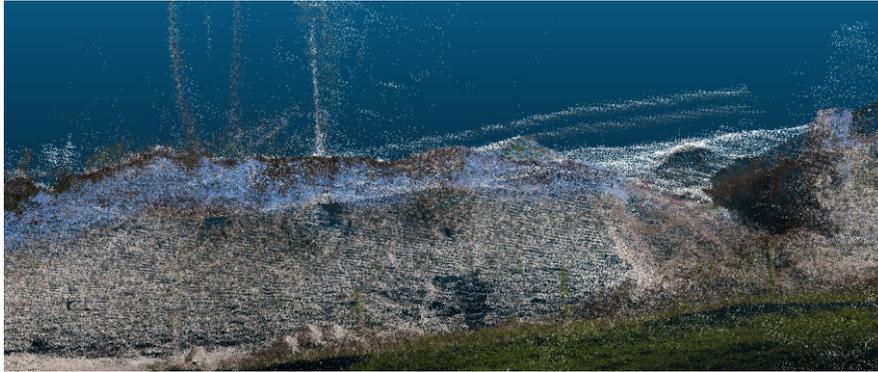


Abbildung I.14: Punktwolke von GeoSLAM ZEB-Revo RT



Abbildung I.15: Scaniverse

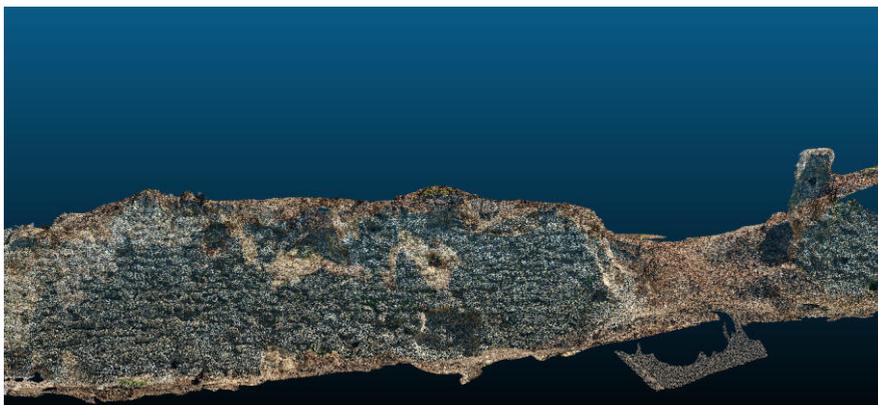


Abbildung I.16: Polycam

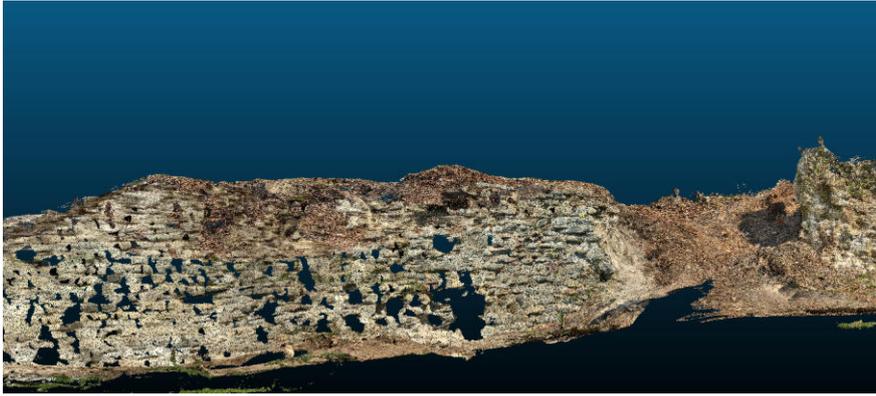


Abbildung I.17: 3d Scanner App



Abbildung I.18: SiteScape

4.8 Vorgehen beim Vergleich

Um die Punktwolken zu vergleichen und zu georeferenzieren, wurde CloudCompare v2.12 eingesetzt. Alle Resultate wurden mindestens einmal in CloudCompare geöffnet. Die Georeferenzierung musste anhand der Farbwerte vorgenommen werden. Das ist sicherlich nicht das genaueste Verfahren. Bei mehreren Messungen wurde jeweils die mit dem besten Resultat für die Auswertung verwendet.

Die Referenzpunkte der Mauer der Ruine Neuenburg wurden uns von Daniel Mayer vom Amt für Archäologie zur Verfügung gestellt. Die Referenzpunkte waren direkt an oder neben der Mauer gesetzt. Sie wurden mit einer Trimble S6 Totalstation in den Landeskoordinaten LV95 eingemessen.

Die Referenzpunkte der Villa Aurum wurden mit einem Leica iCON Robot 70 reflektorlos

und ohne Stationierung gemessen. Sie sind also nicht für eine Georeferenzierung geeignet, reichen aber aus um eine Aussage über die Genauigkeit zu treffen.

4.9 Schlussfolgerung

Der Test hat gezeigt, dass die Handlaser-scanner im Gegensatz zu den herkömmlichen Messverfahren auch Messungen bei schlechten Lichtverhältnissen zulassen. Sie geben eine sofortige Rückmeldung, dank der Bereiche mit einer unzureichenden Punktdichte umgehend erkannt und nachgemessen werden können. Bei den fotogrammetrischen Berechnungen nimmt man die Fotos auf und hat in den meisten Fällen erst am nächsten Tag die daraus resultierende Punktwolke. Wenn dann ein Bereich fehlt, muss man erneut zur Aufnahmestelle fahren und weitere Fotos schießen, bevor man es Neuberechnen lassen kann. Die Gefahr dabei ist, dass die Lichtverhältnisse beim zweiten Messen schon ganz anders sind. Aber auch die Aufnahmezeit kann durch den Einsatz des Leica BLK2GO oder dem GeoSLAM Revo RT verkürzt werden. Das iPad hinkt in diesem Punkt etwas hinterher, da der Messbereich mit den maximal 4 m ziemlich eingeschränkt ist und das Scannen so länger dauert. Die enorme Punktdichte des mit Structure from Motion berechneten Modells konnte keines unserer Geräte erreichen. Eine Möglichkeit die Punktdichte zu erhöhen wäre es, sich langsamer fortzubewegen. Zumindest die beiden professionellen Handlaser-scanner sollten dadurch ein besseres Resultat liefern. Bei den Apps könnte man noch etwas mit den Einstellungen der einzelnen Apps spielen, muss dann aber beachten, dass das unter Umständen mehrere Teilscans ergibt oder die Reichweite noch stärker eingeschränkt wird. Bei der Prozessierung wurde bereits der höchste Standard von jeder App verwendet.

Beim Vergleich der Geräte untereinander stellt man fest, dass die Genauigkeitsrange vom GeoSLAM und Leica BLK2GO mit Abweichungen im Bereich von 5 cm wesentlich besser sind als die der iPad-Apps. Genauso verhält es sich mit dem Scanbereich. In den Punktabständen eines Scans konnten einige Apps das GeoSLAM übertrumpfen. Dabei ist zu beachten, dass mit dem iPad viel länger gemessen wurde und dass das iPad bei der Mauer und bei der Skulptur in einigen Bereichen Löcher aufweist. Bei der Villa und der Skulptur kann man auch gut erkennen, dass das iPad seine eigene Position nicht so gut

bestimmen kann und die einzelnen Teile deshalb nicht so schön zusammenpassen. Die besseren Resultate der professionellen Geräte in diesem Bereich führen wir auch darauf zurück, dass diese mit dem Start und Ende am gleichen Punkt eine Korrekturrechnung inkludieren, um mögliche Verschiebungen beim Positionieren mit dem SLAM Algorithmus auszugleichen.

Im Gegensatz zu den Erfahrungen die Nadine Sennhauser aus ihrer Zeit als Geomatikerin mit stationären Laserscanner gemacht hat, ist die Aufnahme mit Handlaserscanner viel schneller durchgeführt, da der wiederholte Auf- und Abbau des Stativs wegfällt.

Weitere Informationen sind im Anhang 2 Scanner Vergleich zu finden.

5 Umsetzungskonzept

5.1 Aufbau

Der Aufbau des Point Cloud Browsers wird in der Abbildung I.19 dargestellt. Die Technologien die auf Basis der beschriebenen Evaluationen gewählt wurden, lassen sich folgendermassen zusammensetzen.

Das React Frontend ist im Django inkludiert und verwendet zur Ansicht den Punktwolken-Viewer Potree. Die angezeigten Punktwolken und 2D-Ansichten werden aus den entsprechenden Static Ordnern bezogen. Das Backend achtet darauf, dass nur Berechtigte die jeweiligen Daten abfragen können und sorgt zum Beispiel dafür, dass die Pointcloudpipeline zur richtigen Zeit mit den richtigen Parameter ausgeführt wird. Die Pointcloudpipeline nimmt die hochgeladenen Dateien entgegen, konvertiert diese ins LAZ-Format für die Langzeitspeicherung und bereitet die 2D Ansicht über PDAL vor. Untwine wird von der Pipeline zur Umwandlung der LAZ-Dateien ins weboptimierte EPT-LAZ-Format verwendet. Die grossen Anzeige- und Downloaddateien sind im Dateisystem genauer in einem abrufbaren Static-Ordner untergebracht. Die ergänzenden Metadaten zu den Punktwolken werden in einer PostgreSQL-Datenbank abgelegt.

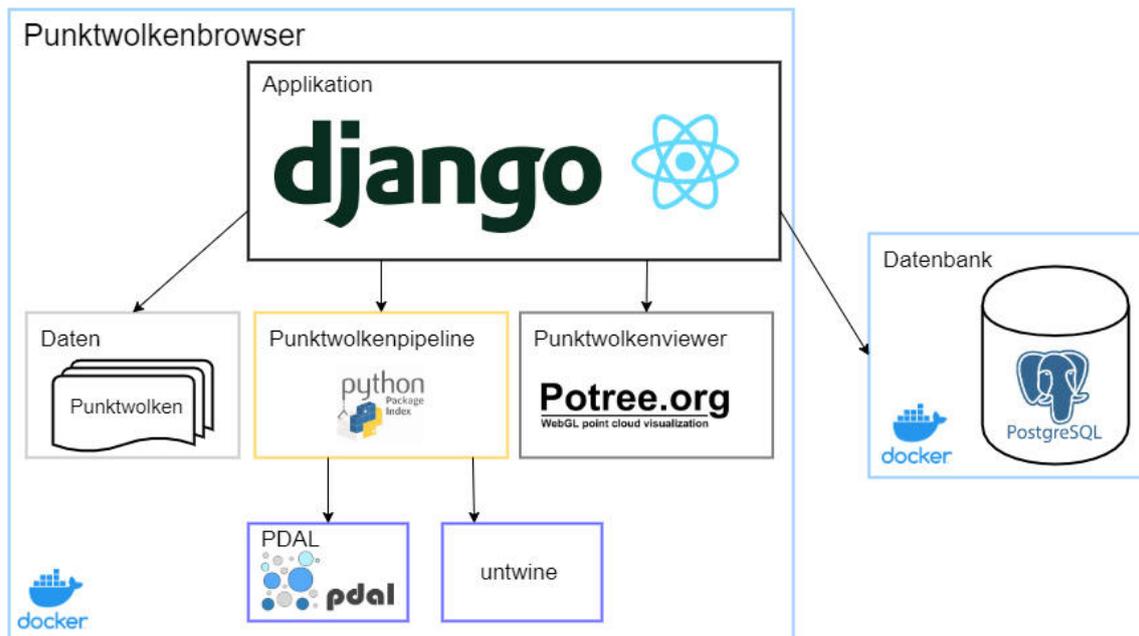


Abbildung I.19: Aufbau Überblick

5.2 Punktwolken Hoch- und Herunterladen

Der Point Cloud Browser akzeptiert das Hochladen von e57, LAS, LAZ und PLY Daten. Damit die teils riesigen Dateien nicht am Stück über HTTP gesendet werden müssen, werden sie in Chunks zerstückelt und in kleineren Paketen an den Server gesendet. Den Ablauf eines Uploads kann man im nachfolgenden Sequenzdiagramm (Abbildung I.20) nachvollziehen. Das erste PUT beinhaltet bereits den ersten Teil der Punktwolke. Damit das Backend bei den nächsten Teilen erkennt, zu welcher Datei die einzelnen Chunks gehören, wird eine spezifische URL und der Dateioffset als Response auf das erste PUT gesendet. Die nachfolgenden Chunks müssen dann alle an diese URL gesendet werden. Der Offset kann im Frontend verwendet werden, um z.B. den Uploadfortschritt anzuzeigen.

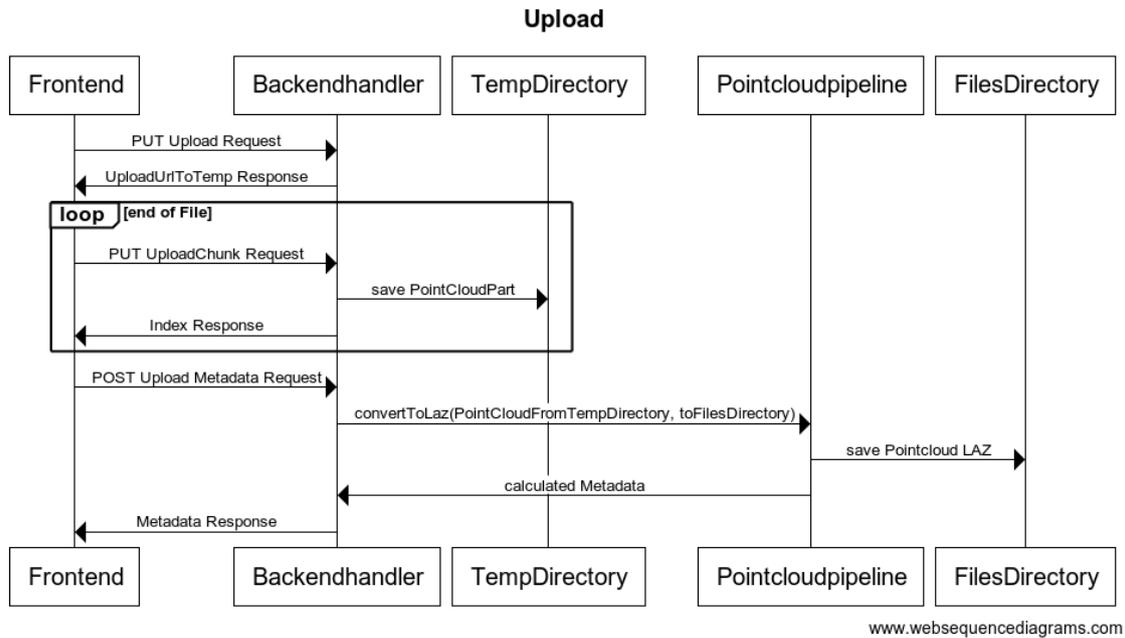


Abbildung I.20: Upload

Das Herunterladen (Abbildung I.21) wurde im Gegensatz zum Upload direkt über einen <a>-Tag gelöst.

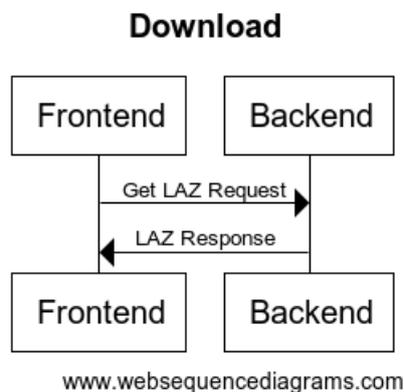


Abbildung I.21: Download

5.3 Punktwolken Ansicht

Da zum Archivieren geeignete LAZ-Dateien nicht für die Anzeige verwendet werden können, ist eine Aufbereitung ins EPT Format nötig. Das Frontend fragt über `api/-pointclouds/{id}/view3d` den Pfad zur entsprechenden Punktwolke ab. Im Backend wird geprüft, ob die Punktwolke schon im EPT Format vorliegt. Falls "nein" wird die Point-

cloudpipeline angestossen und die Umwandlung vorgenommen. Ist diese abgeschlossen wird dem Frontend ein Link zur aufbereiteten Datei zurückgegeben. Die URL zum EPT besteht aus einer zufälligen Zeichenfolge. So wird verhindert, dass man durch Manipulationen des GET Requests fremde Punktwolken ansehen kann. Der Ablauf kann im Sequenzdiagramm der Abbildung I.22 nachvollzogen werden.

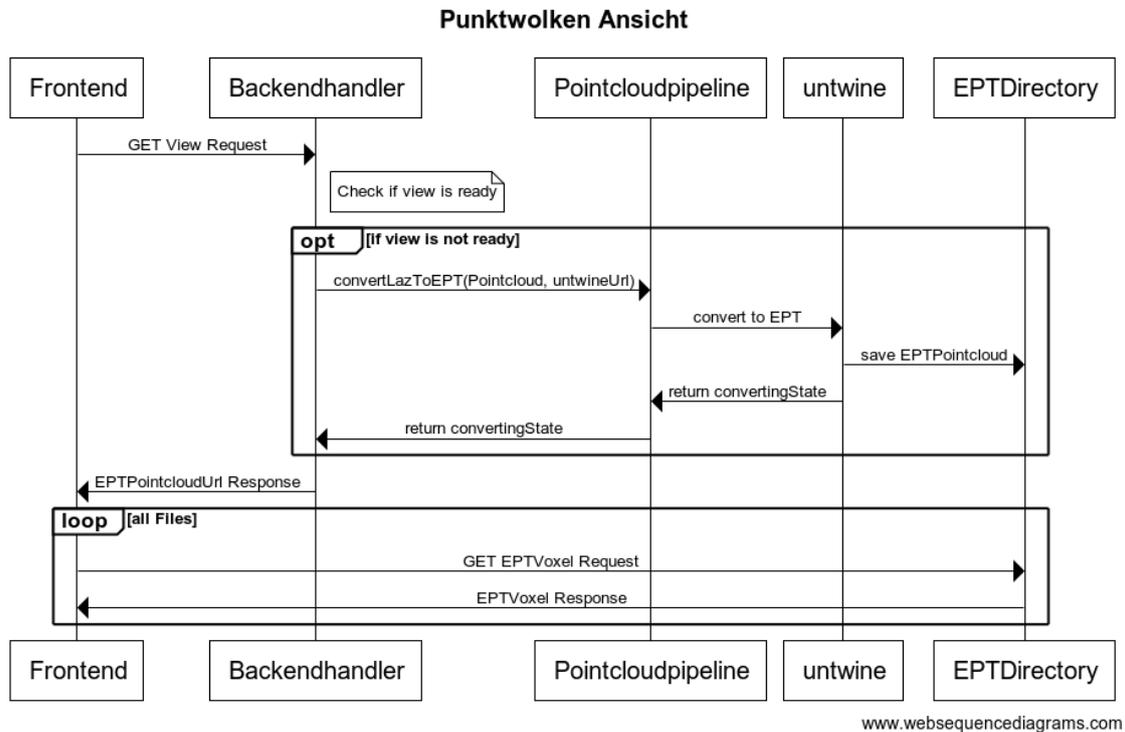


Abbildung I.22: Berechnung der 3D-Ansicht

Die aufbereiteten EPT Daten werden nach 24 Stunden Nichtgebrauch entfernt und müssen bei Bedarf neu berechnet werden. Somit muss die Berechnung nicht bei jeder Anfrage durchgeführt werden, was die Performance bei viel eingesehenen Punktwolken steigert. Die automatische Löschung der abgelaufenen Daten übernimmt ein Task im Django, der das Alter aller Dateien alle 10 min überprüft und alte EPTs entfernt.

5.4 2D-Ansicht und Export

Der 2D-Export des GeoTIFFs setzt eine vorgängige Berechnung der Ansicht voraus. Nach der Aufbereitung kann sie ebenfalls über einen `<a>`-Tag heruntergeladen werden.

Ähnlich wie bei der 3D-Ansicht sendet das Frontend einen Request an das Backend, um die URL zum GeoTIFF zu bekommen. Ist dieses schon wegen einer früheren Prozessierung vorhanden, wird direkt die URL zurückgesendet. Falls der letzte Zugriff schon über 24 Stunden her ist, wird das GeoTIFF gelöscht und muss neu berechnet werden. Die Berechnung der 2D-Ansicht ist auch wieder ein aufwändiger Vorgang, den die Pipeline mit einer PDAL Funktion löst. Bei der Berechnung wird pro Pixel der höchste Punkt in dem Bereich gespeichert. So wird ein Kacheln erzeugt und als GeoTIFF exportiert. Um die Ausgabe noch etwas zu optimieren, werden eindeutige Outliers vor der Umwandlung eliminiert. Das generierte GeoTIFF wird ebenfalls mit einem zufälligen Namen auf dem Server abgelegt und ist dort für 24 h verfügbar. Sobald das entsprechende GeoTIFF bereit steht, wird der Link dem Frontend übermittelt und die View kann das Bild beziehen. Ab diesem Zeitpunkt kann der Benutzer auch den Download starten. Der 2D-Generierungsablauf ist in Abbildung I.23 dargestellt.

Während der Dauer dieses Vorgangs wird der Benutzer dank eines rotierenden Ladesymbols und eines Textes darüber aufgeklärt, was gerade geschieht. Drücken zwei Personen nacheinander auf die gleiche Ansicht, wird es nicht zwei Mal berechnet. Beide Benutzer bekommen zeitgleich ihr GeoTIFF, wenn die Prozessierung abgeschlossen ist. (Dieser Abschnitt funktioniert auch genau so bei der 3D-Ansicht)

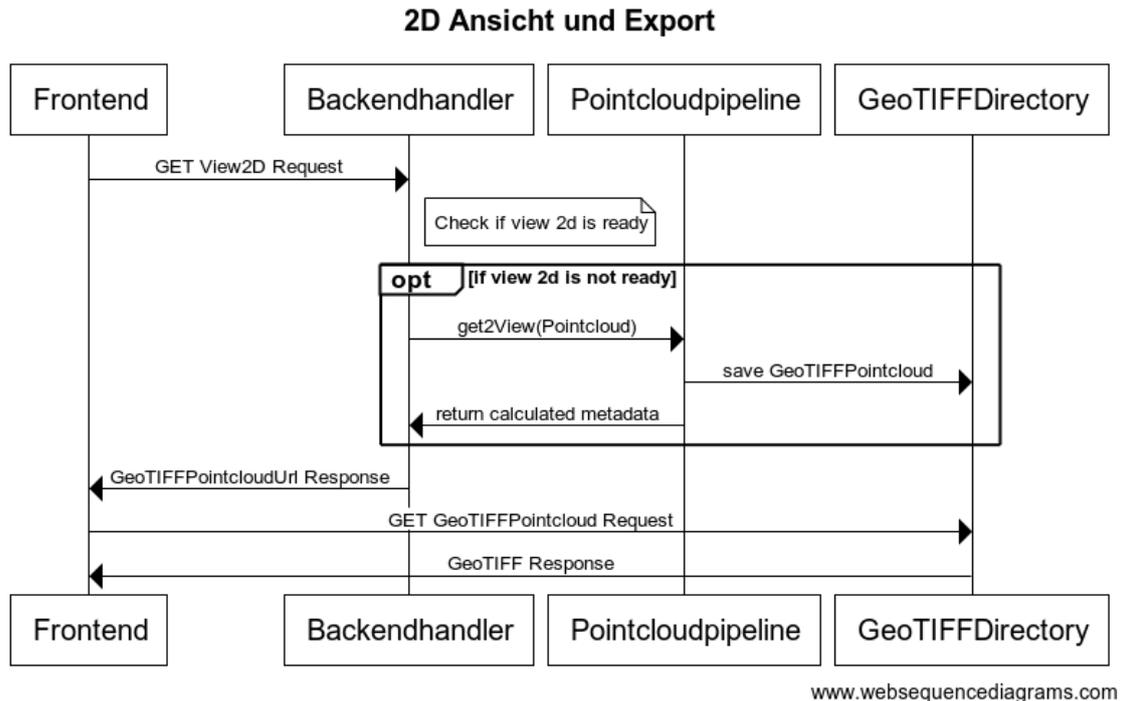


Abbildung I.23: Berechnung der 2D-Ansicht

6 Schlussfolgerung

6.1 Zielerreichung

Die Resultate der Analyse haben gezeigt, dass die getesteten Handlasescanner bereits untereinander einige Unterschiede aufweisen. Das Leica BLK2GO zum Beispiel überzeugt mit der Genauigkeit, das GeoSLAM ZEB-Revo RT mit der besten Ansicht während der Messung und das iPad durch den günstigen Preis. Gegenüber den weiterverbreiteten Erfassungsmethoden für Punktwolken wie Structure from Motion, bieten die Laserscanner den Vorteil, auch bei schlechteren Lichtverhältnissen zu funktionieren. Im Vergleich zu Stativ mit Stop-and-Go sind sie schneller in der Datenerfassung und ermöglichen das sofortige Einsehen der Messresultate. All diese Erkenntnisse sind zusammen mit den Recherchen im technischen Bericht dokumentiert. Dadurch wurde das erste Ziel erreicht.

Mit dem Point Cloud Browser können Punktwolken hochgeladen, verwaltet, visualisiert und geteilt werden. Damit sind die Hauptziele des Prototypen erreicht worden. Aus

zeitlichen Gründen wurde auf die Nebenanforderung - Teilen der Punktwolke inklusive der aktuellen Darstellung per URL - verzichtet. Deshalb muss zurzeit jeder Benutzer die Ansicht der für ihn freigegebenen Punktwolken selber einstellen. In einer Liste sieht der Benutzer alle für ihn verfügbaren Punktwolken. Diese kann er sich in 3D oder 2D anzeigen lassen und bei Bedarf herunterladen. Die selber hochgeladenen Punktwolken können für bestimmte Personen, für sich selber oder für die Öffentlichkeit zugänglich gemacht werden. Wenn wegen der Anzahl Punktwolken die Liste langsam zu unübersichtlich wird, kann man mit der Suchfunktion die Liste nach einem Begriff filtern. Beim Filtern werden dabei alle Felder der Metadaten nach dem Begriff durchsucht. Dadurch kann man sowohl nach Ort oder nach anderen Metadaten filtern. Bis auf das Teilen der Punktwolke konnten alle Anforderungen an den Prototypen erfüllt werden.

Die Pointcloudpipeline kann von jedem über PiPy installiert werden. Das Tool kann neben der Benutzung in der Commandline auch als Package verwendet werden. Damit das Tool eigenständig das Management der LAZ Dateien übernehmen kann, verfügt es über vier Funktionen. Den Auftrag hätten wir bereits mit der Funktion erreicht, welche die Punktwolke in ein GeoTIFF überführt. Wir wollten die Pointcloudpipeline aber so ausrichten, dass sie alles beinhaltet, was man für die Archivierung von Punktwolken braucht. Dazu gehört die Umwandlung der gängigsten Formate in ein archivgeeignetes Format und die Rückgewinnung der Formate, falls sie später noch einmal gebraucht werden. Zudem sollten die Punktwolken im Web angesehen werden können. Dazu ist die Umwandlung in ein EPT-LAZ Format nötig. Mit der Pointcloudpipeline wurde das Ziel dadurch übertroffen, obwohl auch die 2D-ableitbare Berechnung jederzeit weiter optimiert werden kann.

6.2 Weiterentwicklung

- **Ansicht teilen:**

Um Details hervorzuheben wäre es sinnvoll, man könnte die Ansicht mit all den Konfigurationen per Link teilen. Die Idee ist, dass die eingestellte Punktdichte, Ansichtswinkel, Messungen, usw geteilt werden kann. Das Teilen würde erfordern, dass die im Potree-Viewer gesetzten Variablen dynamisch über die URL-Parameter gelesen und gesetzt werden.

- **2D Ansicht:**

In der 2D Ansicht fehlen weitere Interaktionsmöglichkeiten. Das GeoTIFF kann weder verschoben, noch gezoomt oder rotiert werden. Dazu könnte die Farbskala der Ansicht durch den Benutzer auswählbar gemacht werden. Standardmässig in unserem Prototyp sind Graustufen.

- **2D Berechnung:**

Die 2D Berechnung liefert zurzeit nur eine Aufsicht, was zur Erkennung von Gebäudegrundrissen reicht, aber für die Dokumentation einer Fassade zum Beispiel keinen Mehrwert bringt. Für solche Berechnungen käme zum Beispiel PCL infrage. Einige erfolglose Versuche sind im Teil 2 Abschnitt 3.2 ersichtlich.

- **Login und Benutzer:**

Das Login kann noch nicht verändert oder zurückgesetzt werden. In einer produktiven Version müsste zumindest die E-Mail Adresse und das Passwort geändert werden können. Das erfordert allerdings einen Mail-Server. Auch könnte eine erweiterte Benutzerverwaltung mit Benutzergruppen und Rechtesystem implementiert werden.

- **Suche:**

Die Suche nach einer bestimmten Punktwolke funktioniert zwar, ist aber bei einer riesigen Menge an Punktwolken zu wenig. Ein Sortier- bzw. Filtersystem wäre im Django-Backend relativ einfach zu implementieren. Über die Queryset Filter kann diese Anforderung erfüllt werden. Aufwändiger wird es im Frontend, ein Key-Value Filtersystem benutzerfreundlich und ansehnlich zu gestalten.

- **Gruppen:**

Angelehnt an die Weiterentwicklung der Benutzer mit Rechtesystem, könnte in Zukunft ein Gruppensystem implementiert werden. In einer Extraliste könnten zusammengehörige Punktwolken eingegliedert werden. Die Benutzer könnten so ihre Punktwolken besser sortieren und katalogisieren.

6.3 Wir sagen Danke

Stefan Keller danken wir für die Betreuung und die wertvolle Unterstützung.

Roger Bräm danken wir für die Ausleihe des Leica BLK2GO für die Testmessungen.

Jürg Pulver danken wir für die Ausleihe des GeoSLAM ZEB-Revo RT für die Testmessungen.

Daniel Mayer und der Archäologie des Kantons TG danken wir für die Referenzdaten und die Möglichkeit, die Geräte auf der Grabung Neuenburg zu testen.

Pia Langenbach danken wir für die Möglichkeit, die Geräte an der Villa Aurum zu testen.

Dejan Šeatović danken wir für die Auffrischung in tachymetrische Aufnahmetechnik mit den Tachymetern der Fachhochschule OST.

Nicola Jordan danken wir für die Unterstützung mit Django und dem CI/CD im GitLab.

Lars Barmettler danken wir für die Unterstützung beim Einbinden des Potree Viewer

Marc Vögele danken wir für seine Zeit bei den Messungen.

Joël Schwab danken wir für die Unterstützung bei Fragen.

Jonas Weber danken wir für die Überprüfung der Rechtschreibung.

Kapitel II

Projektdokumentation

1 Vision

Die Vision wird in Kapitel 1.2 beschrieben.

2 Anforderungsspezifikation

2.1 Requirements

Die folgenden Requirements wurden aus dem Auftrag zu den drei Lieferobjekten einzeln erarbeitet und auch separat dokumentiert. Dazu haben wir uns mit den Vor- und Nachteilen von bestehenden Tools auseinandergesetzt und den Kontakt zu Archäologen und Archäologinnen gesucht. Da das Interesse an verbesserten Methoden gross ist, sind wir auf entsprechende Hilfsbereitschaft gestossen.

2.1.1 Requirements Lieferobjekt 1 (Dokumentation)

Nr	Anforderung	Gewichtung
1	In der Dokumentation werden drei verschiedene Scanner* ¹ verglichen.	Muss
2	Die drei verschiedenen Scanner* ¹ werden bei je drei Messungen* ² eingesetzt.	Muss
3	Die Erkenntnisse und die Unterschiede aus dem Scannertest werden dokumentiert.	Muss
4	Mindestens eine Test-Punktwolke wird in Potree und Cesium visualisiert.	Soll
5	Die Erkenntnisse aus den Visualisierungen der Testpunktwolken werden dokumentiert.	Soll
6	Zwei verschiedene 3D* ³ Verwaltungsmöglichkeiten werden analysiert und dokumentiert	Soll
7	Mindestens zwei verschiedene* ⁴ Datenspeichermöglichkeiten von Punktwolken werden getestet.	Kann
8	Die Erkenntnisse aus den mind. zwei Datenspeichermöglichkeiten von Punktwolken werden dokumentiert.	Kann

Tabelle II.1: Requirements Delivery 1

*¹ Leica BLK2GO, GeoSLAM ZEB Revo-RT und iPad Pro

*² Eisenplastik an der OST in Rapperswil, Mauer der Neuenburg TG, Villa Aurum

*³ Meshes vs. Punktwolken

*⁴ Mindestens LAZ vs EPT

2.1.2 Requirements Lieferobjekt 2 (Webapplikation)

Nr	Anforderung	Gewichtung
1	Ein registrierter Benutzer kann LAZ Files hochladen.	Muss
2	Ein registrierter Benutzer kann Punktwolken in verschiedenen Datenformaten* ¹ hochladen.	Vision
3	Punktwolken können in 3D visualisiert* ² werden.* ³	Muss
4	Ein registrierter Benutzer kann seine Punktwolken anderen registrierten Benutzern freigeben (Keine Bearbeitungsrechte).	Soll
5	Ein registrierter Benutzer kann seine Punktwolke für alle öffentlich zugänglich machen.	Soll
6	Ein registrierter Benutzer kann die für ihn freigegebenen, öffentlich zugänglichen und die eigenen Punktwolken visualisieren.	Muss
7	Ein registrierter Benutzer kann seine hochgeladenen Punktwolken bereinigen, damit er störende Punkte entfernen kann.	Soll
8	Ein registrierter Benutzer kann nur seine hochgeladenen Punktwolken löschen und bearbeiten.	Soll
9	Punktwolken können in 2D visualisiert und heruntergeladen werden.* ³	Kann
10	Die Datenverwaltung wird vom Tool übernommen.	Muss
11	Punktwolken können heruntergeladen werden.* ³	Kann
12	Ein Gast kann nur öffentlich verfügbare Punktwolken visualisieren, in 2D konvertieren und herunterladen.	Soll
13	In den Punktwolken-Visualisierungen kann gezoomt, rotiert und verschoben werden.	Muss
14	In den Punktwolken-Visualisierungen können Strecken, Distanzen und Abstände gemessen werden.	Soll
15	Die Visualisierungen der Punktwolken können per Link geteilt werden und bei ausreichender Berechtigung* ³ angesehen werden.	Soll
16	Ein Benutzer* ⁴ sieht alle für ihn verfügbaren* ³ Punktwolken in einer Liste.	Soll
17	Ein Benutzer* ⁴ kann die Liste der Punktwolken nach Ort filtern.	Kann
18	Ein registrierter Benutzer kann seine Punktwolken kategorisieren, um sie schneller zu finden.	Kann
19	Ein Benutzer* ⁴ kann die Punktwolken-Liste nach Kategorie filtern.	Kann
20	Ein registrierter Benutzer kann seine Punktwolken mit anderen Punktwolken vereinen.	Vision

Tabelle II.2: Requirements Delivery 2

*¹ LAS, LAZ, e57, PLY

*² Zum Beispiel einfärben, Punktdichte verändern, etc.

*³ Mit der nötigen Freigabe (Möglichkeiten: eigene, einem freigegebene oder öffentliche Punktwolken).

*⁴ Gast und registrierter Benutzer

2.1.3 Requirements Lieferobjekt 3 (Geoprozessierungs-Pipeline)

Nr	Anforderung	Gewichtung
1	Als Benutzer kann ich aus einer 3D Punktwolke 2D Ansichten generieren, um davon schnell und einfach 2D-Vektordaten abzuleiten.	Muss
2	Integration in Prototyp.	Kann

Tabelle II.3: Requirements Delivery 3

2.2 Benutzer-Analyse

Die Analyse lässt zwei Personas erahnen. Zum einen gibt es die Endvision eines Tools, welches jedem erlaubt, die LiDAR Technologie einfach zu verwenden. Dabei sollten LiDAR-Sensoren von Smartphones für jeden verfügbar sein. Dies werden wir aber mit dem Prototypen noch nicht erreichen. Damit wir die Augen dennoch offen halten, wurde auch eine Visions Persona in Abbildung II.2 erarbeitet. Die Bedürfnisse der Archäologen und Denkmalpfleger werden im Abbild II.1 abgebildet und sind auch die relevanteren in diesem Projekt. Einige ihrer Frustrationspunkte sollten mit dem Prototyp und je nach Resultat der Handlaserscanner durch diese abgeschwächt werden.

2.2.1 Persona 1

<p>Biografie</p> <ul style="list-style-type: none"> • Henry Jones Junior • 39 Jahre alt • Er ist seit 5 Jahren bei der Archäologie. • Er arbeitet viel draussen auf den verschiedenen Grabungsstellen. • Er kam ursprünglich aus dem Denkmalpflegebereich. 		<p>Aufgaben und Verhalten</p> <ul style="list-style-type: none"> • Er ist neben dem kräfteaubenden Freilegen der Objekte auch mit Ausmessen und Dokumentieren beschäftigt. • Er hat in den letzten Jahren viele Techniken gelernt, um relative schnell Zeichnungen von den Grabungsstellen anzufertigen. • Er ist aber auch mit den moderneren Vermessungstechniken in Berührung gekommen und kennt die Fotogrammetrie, Laserscanning und kann tachymetrische Messung vornehmen. • Er möchte die Situationen immer so genau wie möglich dokumentieren, aber dies ist aus Zeitdruck und Kosteneffizienz nicht immer möglich.
<p>Situation und Kontext</p> <ul style="list-style-type: none"> • Die Notgrabungen sollten schön und schnell dokumentiert werden. • Das Budget seines Arbeitgebers ist limitiert, das beschränkt den Einsatz von teuren Geräten. • Das Interesse an Ausgrabungen ist gross. Es sollte deshalb für die Nachwelt sinnvoll und verständlich dokumentiert und zumindest für das Fachpublikum zugänglich gemacht werden. • Von der Entwicklung neuer Technologien ist er begeistert und würde sich damit gerne seinen Arbeitsalltag verbessern. • Zurzeit wendet er vorwiegend fotogrammetrische Verfahren an. 	<p>Bedürfnisse</p> <ul style="list-style-type: none"> • Die Grabungen sollen so schnell und genau wie möglich erfasst werden. • Die Daten sollten einfach und zeitnah in 2D und 3D präsentiert und speichereffizient archiviert werden können. <p>Frustpunkte</p> <ul style="list-style-type: none"> • Die neuen und genauen Geräte (wie Tachymeter und terrestrischer Laserscanner) sind sehr aufwendig in der Verwendung, Auswertung und teuer in der Anschaffung. • Die Fotogrammetrie braucht gute Lichtverhältnisse und hat lange bei der Auswertung. 	

Abbildung II.1: Hauptpersona Archäologe

2.2.2 Persona 2

<p>Biografie</p> <ul style="list-style-type: none"> • Tanja Martinez • 20 Jahre alt • Sie ist Psychologie-Studentin. • Sie ist leidenschaftliche OSM-Mapperin. 		<p>Aufgaben und Verhalten</p> <ul style="list-style-type: none"> • Sie ist schon seit Langem hobbymäßig OSM-Mapperin. • Sie ist stets bemüht, alle relevanten Neuerungen zu erfassen. • Sie sieht häufig Kleinigkeiten, die sie verbessern kann, hat aber zu wenig Möglichkeiten, diese effizient zu messen, um sie anschliessend in der Karte zu erfassen. • Sie besitzt neu ein iPhone 12 pro Max mit integriertem LiDAR-Sensor.
<p>Situation und Kontext</p> <ul style="list-style-type: none"> • Ihr ist sich nicht wirklich bewusst, wozu sie den LiDAR-Sensor braucht, geschweige dann, wie die Technologie dahinter funktioniert. • Die Zeit, die sie neben dem Studium für OSM investieren kann, ist leider knapp. • Sie verwendet in Ihrem Alltag ihr Smartphone für fast alles (Bezahlen an der Kasse, Fotos, Kalender, ...). 	<p>Bedürfnisse</p> <ul style="list-style-type: none"> • Sie würde gerne mehr Daten in OSM erfassen, ohne mehr Zeit zu investieren. • Sie wünscht sich eine einfache und sinnvolle Einsatzmöglichkeit des LiDAR-Sensors. <p>Frustpunkte</p> <ul style="list-style-type: none"> • Sie verfügt über keine Erfahrungen zum effizienten Einmessen neuer Gegebenheiten. • Es gibt zu wenig und kaum aktuelle Grundlagedaten für eine schnelle Erfassung in OSM. 	

Abbildung II.2: Visionspersona Opensource Mapper

2.2.3 Ist-Customer Journey Map

Messung vorbereiten:

1. Es muss entschieden werden, wie genau die Dokumentation zu erfolgen hat. Dazu stellt sich die Frage, was für einen Detaillierungsgrad benötigt wird und welche Geräte zurzeit zur Verfügung stehen. Wenn bei der Ausgrabung Knochen gefunden werden, ist eine Absprache mit den Anthropologen bestimmt eine sinnvolle Idee. Denn der Anspruch an die Genauigkeit erhöht sich dadurch, weil sie Analysen aufgrund der Messungen anstellen können und nicht nur bestimmt wird, was für Beigaben mitbestattet wurden. Nach allen Abklärungen wählt Henry ein fotogrammetrisches Verfahren.
2. Kameras stehen immer zur Verfügung und auch die Lizenz für die Auswertungssoftware steht bereit.
3. Leider sind die Lichtverhältnisse an der Grabstelle sehr schlecht. Es muss ein Beleuchtungskonzept her, damit geeignete Fotos vom Fund gemacht werden können.
4. Um das Resultat richtig zu skalieren und gerade zu georeferenzieren, werden Referenzpunkte markiert und mit GNSS gemessen.
5. Damit bei den Aufnahmen kein Chaos entsteht und nichts vergessen geht, legt Henry fest, auf welchem Pfad die Fotos aufgenommen werden.

Messen:

1. Das Fotografieren eines Fundes von sechs Quadratmetern dauert ca. 70 min.

Auswerten:

1. Das Auswerten wird im Büro gemacht, da die Modelle vor Ort nicht gebraucht werden und es ein leistungsstarkes Notebook zur Auswertung braucht.
2. Den Markern werden automatisch die Koordinaten zugewiesen.
3. Doppelte Fotos und solche mit schlechter Qualität werden entfernt.
4. Bereiche mit ungewünschten Objekten werden maskiert.
5. Das Berechnen der relativen Bildorientierung und der Sparse Cloud dauert heute ca. 10 min.

6. Das Weiterverarbeiten zum Mesh und das Bereinigen der Kanten, die zu flachen Übergängen wurden, dauert noch einmal 50 min.

Präsentieren:

1. Im Plan der Archäologie wurde der Fund mithilfe der exportierbaren und georeferenzierten Orthofotos eingezeichnet.
2. Mithilfe eines 3D-PDFs werden die Daten für Drittanwender wie die Anthropologie weitergereicht.
3. In einigen Fällen werden die virtuellen 3D-Modelle auf eine HoloLens-Brille übertragen, damit Anwohner die Grablegungen an Ort und Stelle betrachten können.

Archivieren:

1. Grundlagenbilder werden vorübergehend archiviert.
2. 3D Mesh Modelle würden sie gerne archivieren, da sie häufiger verwendet werden als Punktwolken, zurzeit besteht aber kein geeignetes Verfahren für diese Datenmengen.
3. Die 3D-PDFs sind nur vorübergehend abrufbar.

2.2.4 Soll-Customer Journey Map

Messung vorbereiten:

1. Es sind genügend Handlaser Scanner verfügbar und die Verarbeitung ist mit OpenSource-Software sichergestellt.
2. Der Messpfad wird definiert und entsprechend genügend viele geladene Batterien für den Einsatz eingepackt.

Messen:

1. In etwas langsamerem Schrittempo wird das Objekt umkreist und die Messungen am Startpunkt nach 5 min wieder beendet. Durch die langsame Aufnahme kann eine höhere Punktdichte und dadurch ein detailreicheres Resultat erwartet werden.

Auswerten:

1. Die Daten werden auf den zentralen Rechner geladen.
2. Die Punktdaten werden auf dem Feld auf ihre Vollständigkeit geprüft und gegebenenfalls nachgemessen.
3. Im Büro werden die Daten bereinigt und unerwünschte Punkte entfernt.

Präsentieren:

1. Die Punktwolken können freigegeben werden.
2. Die freigegebenen Punktdaten können heruntergeladen werden.
3. 2D-Ableitungen können exportiert werden.
4. Die Punktwolken können nach erfassten Attributen gefiltert werden.

Archivieren:

1. Alle hochgeladenen Daten werden dauerhaft gespeichert.

2.3 Aktoren

Aus der Benutzer-Analyse zeichneten sich die folgende Aktoren ab. An sich haben sie verschiedene Hintergründe und Ziele, aber wenn sich die Technik, wie zu erwarten ist, weiter in Richtung Verbreitung von LiDAR-Sensoren entwickelt, könnten beide von einer Plattform zum Verarbeiten, Speichern und Freigeben von Daten profitieren.

Aktor	Beschreibung
Archäologe oder Denkmalpfleger draussen	Ein Benutzer, welcher eine Aufnahme eines Objekts macht und anhand der raschen Darstellung der Punktwolke im Browser über das weitere Vorgehen bei der Erfassung der 3D-Daten entscheiden will.
Archäologe oder Denkmalpfleger im Büro	Ein Benutzer, welcher die erfassten Daten aufwändig prozessiert und bereinigt hat und diese für eine lange Zeit speichern will. Er will die Daten schön aufbereitet an weitere Beteiligte und der Öffentlichkeit freigeben.
Opendata Mapper	Ein Benutzer, welcher mit seinen erfassten Daten die Opendata Community unterstützen will.

Tabelle II.4: Aktoren-Beschreibung

Pitch-Beschreibung:

- **Archäologe:**

Als Archäologe gehen Sie mit einem Handlaserscanner auf die Grabungsstelle und 3D-erfassen ihre Objekte in einem Bruchteil der Zeit, die sie sonst mit fotogrammetrischen Verfahren gebraucht hätten. Zu Hause oder an einem Laptop direkt auf der Grabungsstelle laden Sie die aus dem Scan entstandene Punktwolke über den Browser in unser Tool, in welchem Sie diese per Link mit ihren Berufskollegen teilen können. Die Kollegen können die Punktwolke dann nicht nur anschauen, sondern darin auch Messungen vornehmen oder sie herunterladen. Zudem kann Ihnen der 2D-Export helfen, Bruchkanten zu erkennen, um ein schöneres Modell aus der Punktwolke abzuleiten.

- **Denkmalpfleger:**

Als Denkmalpfleger scannen Sie mit einem Handlaserscanner das Denkmal und 3D-erfassen das Objekt schneller als mit fotogrammetrischen oder terrestrischen Scanmethoden. Die aus dem Scan entstandene Punktwolke laden Sie noch vor Ort am Laptop oder zu Hause am Computer in unser Tool. In diesem können Sie die Punktwolke per Link mit ihren Berufskollegen oder sogar mit der Öffentlichkeit teilen.

- **Mapper:**

Als Mapper scannen Sie mit einem neueren Smartphone oder Tablet die Fassade eines Gebäudes oder einer sonstigen für OSM relevante Struktur. Direkt über einen Laptop oder zu Hause am Computer können Sie die Punktwolke des gescannten Objekts in unser Tool laden. Um in OSM-Details des Objekts einzupflegen, können Sie in unserem Tool Messungen an der Punktwolke vornehmen oder direkt eine 2D Ansicht exportieren, um diese zu vektorisieren.

2.4 Use Cases

Die Use Cases wie sie in Abbildung II.3 und im nachfolgenden Abschnitt dokumentiert sind, wurden zu Beginn erarbeitet.

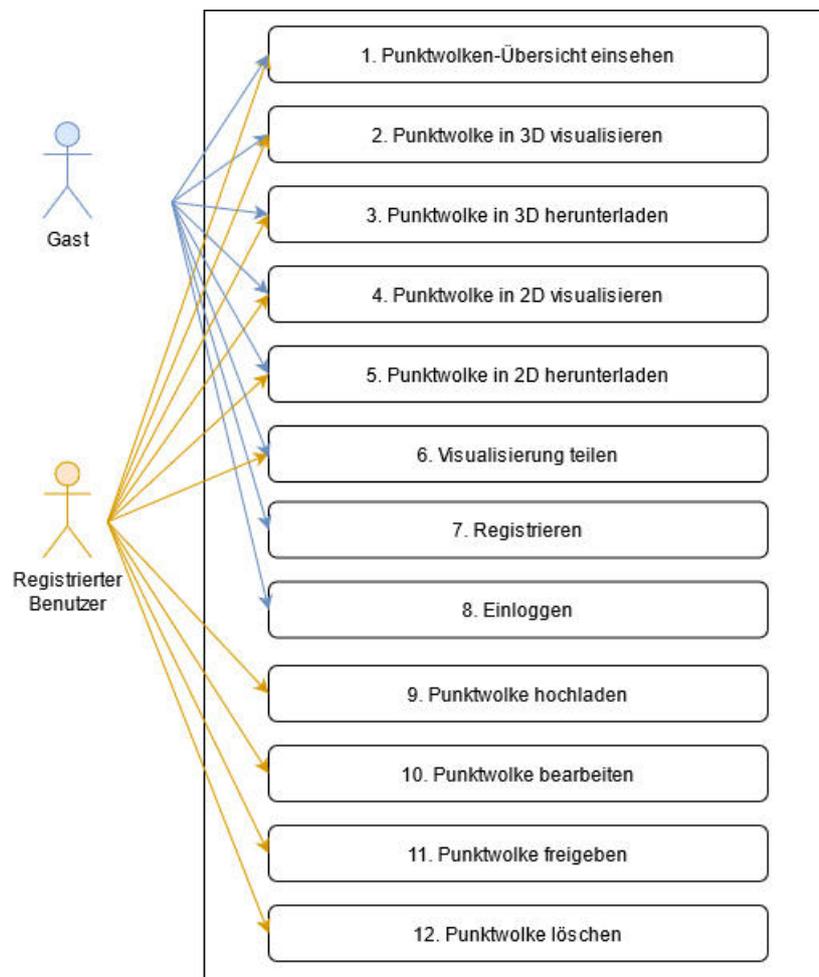


Abbildung II.3: UseCase-Diagramm

1. **Punktwolken-Übersicht einsehen** - Der Benutzer kann in einer Übersicht die

für ihn verfügbaren Punktwolken ansehen.

2. **Punktwolke in 3D visualisieren** - Der Benutzer kann die eigenen oder die ihm freigegebenen Punktwolken visualisieren. Die Punkte der gewünschten Punktwolke werden dargestellt und der Benutzer kann die Darstellung anpassen. Die Visualisierung lässt sich rotieren, verschieben und drehen. Ausserdem kann man an der Punktwolke Messungen vornehmen.
3. **Punktwolke in 3D herunterladen** - Der Benutzer kann die für ihn verfügbaren Punktwolken herunterladen.
4. **Punktwolke in 2D visualisieren** - Der Benutzer kann die für ihn verfügbaren Punktwolken in 2D visualisieren.
5. **Punktwolke in 2D herunterladen** - Der Benutzer kann die für ihn verfügbaren 2D Ansichten der Punktwolken herunterladen.
6. **Visualisierung teilen** - Der Benutzer kann einen Link zu seiner Visualisierung mit der gewünschten Darstellung generieren und diesen mit anderen Benutzern teilen. Dabei muss natürlich die Berechtigung für die entsprechende Punktwolke vorhanden sein.
7. **Registrieren** - Der Gast registriert sich und wird somit zu einem registrierten Benutzer.
8. **Einloggen** - Der Gast loggt sich ein.
9. **Punktwolke hochladen** - Der registrierte und eingeloggte Benutzer kann Punktwolken in verschiedenen Dateiformaten hochladen.
10. **Punktwolke bearbeiten** - Der registrierte und eingeloggte Benutzer kann seine eigenen Punktwolken bearbeiten, indem er die Metadaten verändert oder die Punktwolke durch eine lokal bearbeitete Version ersetzt.
11. **Punktwolke freigeben** - Der registrierte und eingeloggte Benutzer kann Punktwolken anderen registrierten Benutzern freigeben oder die Punktwolke öffentlich und somit auch für Gäste zugänglich machen.

12. **Punktwolke löschen** - Der registrierte und eingeloggte Benutzer kann seine eigenen Punktwolken löschen.

2.5 User Stories

- Als Gast will ich öffentlich verfügbare Punktwolken visualisieren, damit ich interessante Bereiche der Punktwolke besser sehe.
- Als Gast oder eingeloggter Benutzer will ich meine Visualisierungen per Link teilen können, damit andere Interessierte dieselbe Ansicht betrachten können.
- Als Gast will ich öffentlich verfügbare Punktwolken in 2D visualisieren und anschliessend exportieren, damit ich 2D-Vektordaten davon ableiten kann.
- Als Gast will ich öffentlich verfügbare Punktwolken herunterladen, damit ich diese in anderen Tools verwenden kann.
- Als Gast will ich öffentlich verfügbare Punktwolken in einer Liste sehen und diese nach Keywords filtern, damit ich die Interessantesten schnell finde.
- Als eingeloggter Benutzer will ich Punktwolken in verschiedenen Datenformaten hochladen, damit ich diese anschliessend visualisieren kann.
- Als eingeloggter Benutzer will ich die Metadaten meiner Punktwolken bearbeiten oder die Punktwolke durch eine von mir lokal bearbeitete Version ersetzen, damit die Daten so aktuell wie möglich sind.
- Als eingeloggter Benutzer will ich meine Punktwolken anderen eingeloggten Benutzern oder für die Öffentlichkeit freigeben, damit andere Personen auch von meiner Punktwolke profitieren können.
- Als eingeloggter Benutzer will ich meine Punktwolken auch löschen können, damit ich Ordnung in meinen Daten schaffen kann.

Die User Stories des Gastes sind natürlich auch für den eingeloggten Benutzer relevant. Dabei ändert sich, dass es sich beim eingeloggten Benutzer nicht mehr nur um die öffentlich verfügbaren Punktwolken, sondern um die ihm Verfügbaren handelt.

2.6 Nichtfunktionale Anforderungen

Die nichtfunktionalen Anforderungen sind je nach Lieferobjekt unterschiedlich.

Lieferobjekt 1 (Dokumentation): Beim Vergleich der Handlaserscanner entstand keine Software, sondern ein Bericht. Deshalb wurden keine nichtfunktionalen Anforderungen erarbeitet.

Lieferobjekt 2 (Webapplikation):

Security:

- Nur ein registrierter Benutzer kann Daten hochladen.
- Nur der, der die Daten hochgeladen hat, kann sie löschen und bearbeiten.
- Freigegebene Daten mit Release Type "Public" können ohne Login angesehen werden.
- Freigegebene Daten an bestimmte Nutzer können nur durch diese eingesehen oder heruntergeladen werden.
- Freigegebene Visualisierungen können nur bei Zugriffsberechtigung auf die zugrunde liegende Punktwolke angesehen werden.

Availability:

- Im erzeugten Prototypen werden bei ausreichender Netzwerkverbindung 95% aller Anfragen an das Backend fehlerfrei beantwortet.
- Keine Angaben zu Betriebszeiten, da der Betrieb des Prototyps auch nicht geprüft werden kann.

Reliability:

- Hardware Fehler [800'000 FIT] + Software Fehler [800'000 FIT] (FIT = Fehler in 1×10^9 Stunden)

Recoverability:

- Im Falle eines vollständigen Systemausfall kann der Prototyp in einem Personentag wiederhergestellt werden.

Usability:

- Das Hochladen einer 200 MB grossen Punktwolke gelingt dem Benutzer in 30 min mithilfe des Handbuches.
- In Eingabefeldern können nur Daten im passenden Format eingegeben werden. Alle Eingaben werden validiert, um falsche Daten zu verhindern.
- Es werden verständliche Fehlermeldungen ausgegeben.
- Die hochgeladenen Dateien sind nach maximal einem Tag für alle sichtbar.

Performance:

Definition Hardware: Unser Backend läuft auf einem Rechner mit vier 2.2 Ghz Prozessorkernen, 8GB RAM und 60GB Festplattenspeicher und einer Gigabit-Ethernet Netzwerkkarte.

Definition 1: Die Webseite ist in Betrieb. Es sind 20 User registriert (Es arbeiten 20 Personen im Amt für Archäologie des Kanton Thurgau). Fünf weitere User nutzen die Webseite zurzeit aktiv und sind eingeloggt. Die Netzwerkverbindung beträgt mindestens 50 Mbps.

Definition 2: Die Webseite ist in Betrieb. Es ist ein weiterer registrierter User am Upload einer Punktwolke. Es sind fünf weitere User am Visualisieren von Punktwolken. Die Netzwerkverbindung beträgt mindestens 50 Mbps.

- Das Login in die Applikation dauert unter Annahme der Definition 1 nicht länger als 10 Sekunden.
- Nach dem Login werden unter der Annahme der Definition 2 innerhalb von 20 Sekunden die verfügbaren Punktwolken angezeigt.

- Das Hochladen einer Punktwolke mit (3'000'000 Punkten) dem Klick auf Upload dauert unter Annahme der Definition 1 nicht länger als zehn Minuten.

Resource Behavior:

- Der Prototyp wird mindestens 40 registrierte User unterstützen. (Mitarbeiterzahl von Archäologie TG + einige Hobby Scanner)
- Der Prototyp wird mindestens 50 gleichzeitige User unterstützen. (Interessenten Publikum an den Daten von Archäologie TG)

Analysability:

- Bei einem Benutzererror wird dem User eine Meldung ausgegeben. Diese fordert ihn auf, die Aktion zu wiederholen.
- Logfiles werden auf dem Server abgelegt.

Testability:

- Mit den System-Tests wird sichergestellt, dass die funktionalen Anforderungen erfüllt werden.
- Mit automatisierten Performancetests werden die Geschwindigkeiten ausgewertet.
- Alle automatischen Unit-Testfälle müssen erfolgreich sein, bevor Änderungen des Quellcodes in die stabile Software übernommen werden.
- Mit Sonarqube wird die Qualität des Codes laufend überwacht.

Installability:

- Die Verwendung der Webseite sieht keine Installation vor.
- Das Aufsetzen der Web-Applikation sollte mittels Docker erfolgen und genau beschrieben sein.

- Mit der Dokumentation kann jeder Softwareentwickler die Webapplikation in einem Tag aufsetzen und zum Laufen bringen.

Modifiability:

- Der Ausbau des Prototyps ist für einen Developer, der die Repo-Beschreibung gelesen und Erfahrung mit Django und Javascript hat, möglich.

Requirements Lieferobjekt 3 (Webapplikation):

Security / Usability / Fault Tolerance:

- Die Eingangsdaten werden vor der Umwandlung geprüft.
- Daten in unbekanntem Format werden nicht transformiert.

Availability:

- 90% aller Anfragen an die Pipeline werden fehlerfrei beantwortet.
- Keine Angaben zu Betriebszeiten, da der Betrieb des Prototyps auch nicht geprüft werden kann.

Reliability:

- Software Fehler [800'000 FIT] (FIT = Fehler in 1×10^9 Stunden)

Recoverability:

- Die Installation ist genau beschrieben.
- Mit der Dokumentation kann jeder diesen in 30 min installieren und zum Laufen bringen.

Performance:

Definition Hardware: Die Pipeline läuft auf einem Rechner mit vier 2.2 Ghz Prozessorkernen, 3.8 GiB RAM.

Definition Testpunktvolke: Die umzuwandelnde Punktvolke hat 1843780 Punkte und ist im LAZ format 6.9 MiB (7,190,847 bytes) gross.

- Der Rechenaufwand der Umwandlung in eine 2D-ableitbare Ansicht der Testpunktvolke dauert nicht länger als 10 Minuten.

Analysability:

- Bei Problemen wird ein Kommentar auf der Kommandozeile ausgegeben.

Testability:

- Mit den System-Tests des Point Cloud Browsers wird sichergestellt, dass alle funktionalen Requirements erfüllt werden.
- Alle automatischen Unit-Testfälle müssen erfolgreich sein, bevor Änderungen des Quellcodes in die stabile Software übernommen werden.
- Mit Sonarqube wird die Qualität des Codes laufend überwacht.

Modifiability:

- Die Pipeline sollte in die Software eingebaut werden können.

3 Design

3.1 Architektur

Die Architekturbeschreibung wird anhand des C4-Architekturmodells veranschaulicht. C4 bietet drei Levels, mit denen die Architektur auf unterschiedlichen Zoomstufen nachvollzogen werden kann.[6]

3.1.1 C4 Level 1

Die Abbildung II.4 zeigt eine Gesamtübersicht über die Architektur auf dem Level 1 im C4-Architekturmodell und bildet ab, in welchem Kontext sich der Point Cloud Browser befindet.

Level 1

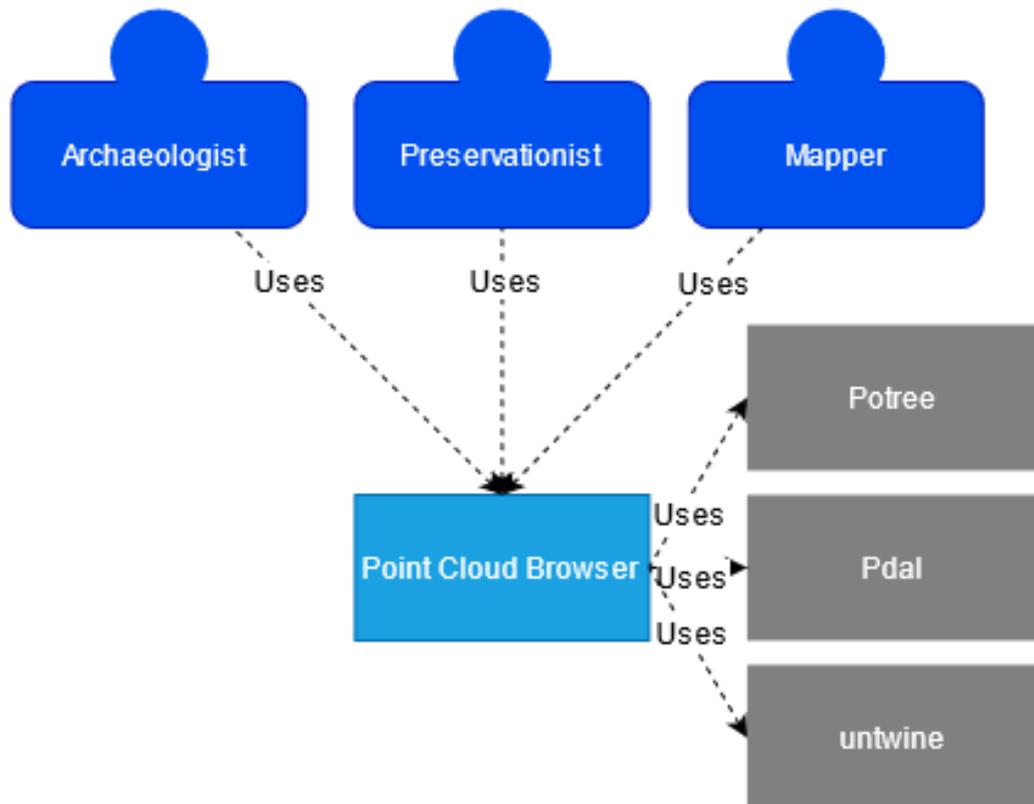


Abbildung II.4: C4 Modell Level 1

3.1.2 C4 Level 2

Im Abbild II.5 ist der Level 2 dargestellt, worin die einzelnen Teile des Systems und deren Zusammenspiel erkannt werden.

Level 2

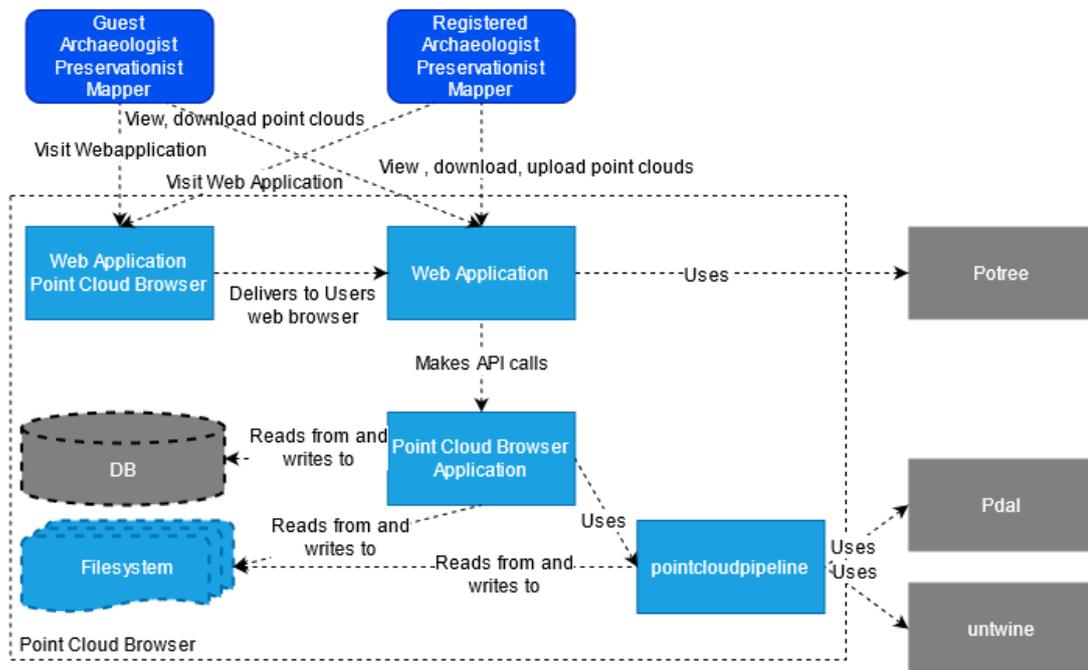


Abbildung II.5: C4 Modell Level 2

3.1.3 C4 Level 3

Der Level 3 ist der letzte Schritt bevor es zur Code-Umsetzung übergeht und dadurch der Level 4 entsteht. In der Abbildung II.6 ist der innere Aufbau des React Frontends abgebildet, in der Abbildung II.7 der des Backends und zu guter Letzt in der Abbildung II.8 der der Pointcloudpipeline.

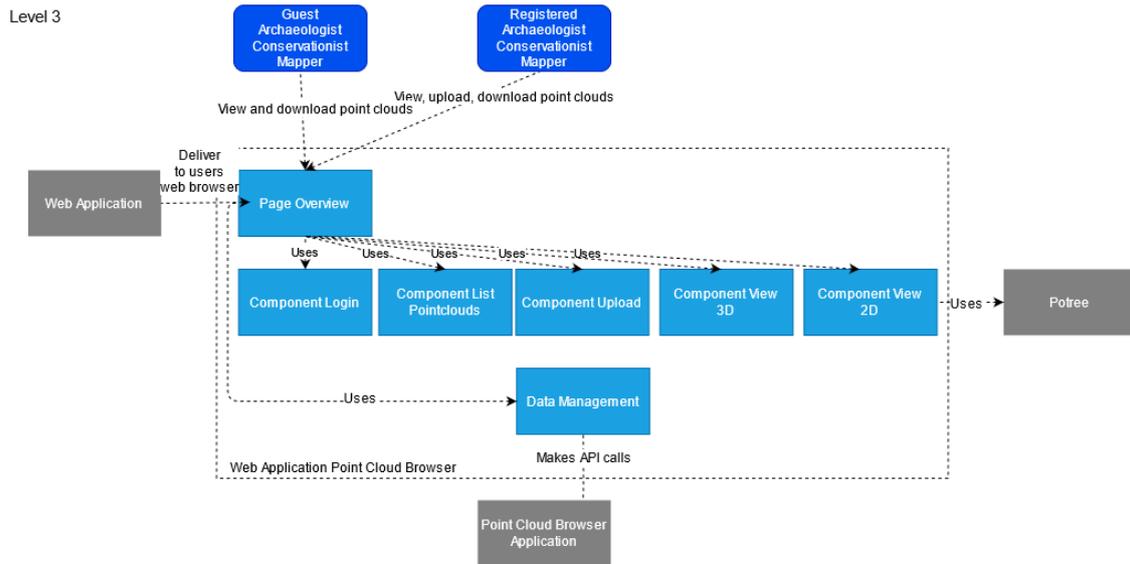


Abbildung II.6: C4 Modell Level 3 Frontend

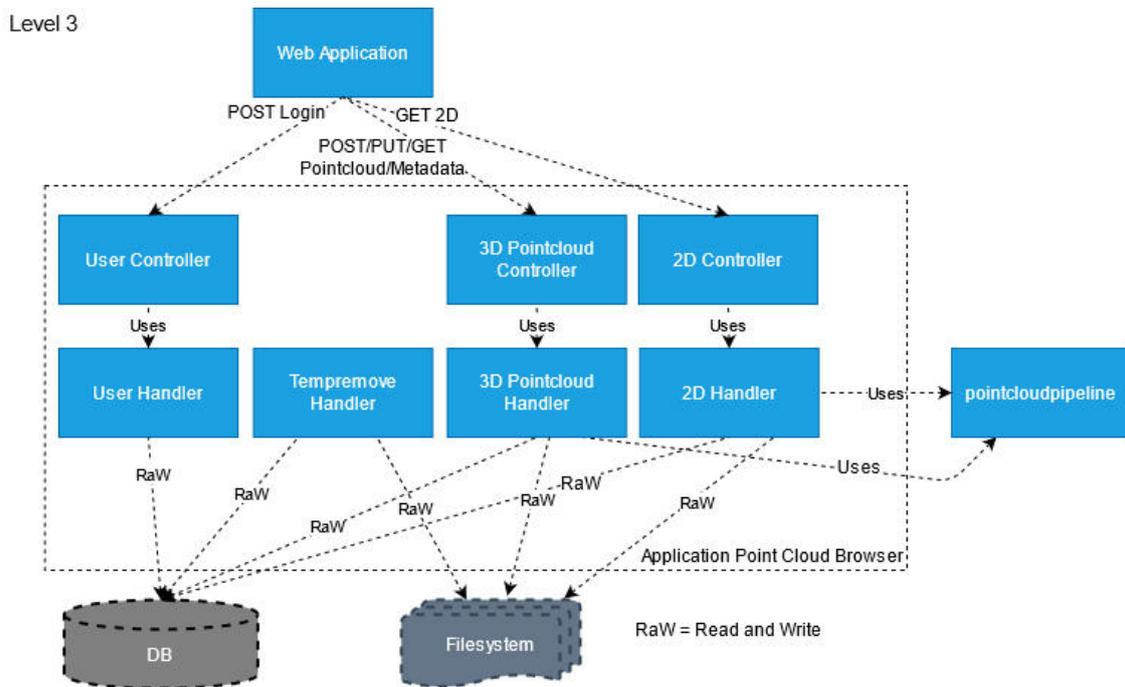


Abbildung II.7: C4 Modell Level 3 Backend

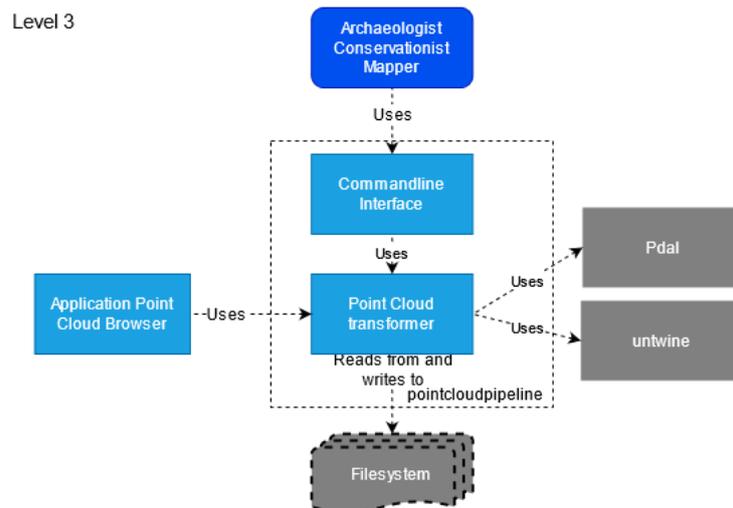


Abbildung II.8: C4 Modell Level 3 Pointcloudpipeline

3.2 Wichtige Abläufe

Punktwolken Hoch- und Herunterladen:

In diesem Projekt wurden wir zum ersten Mal mit dem Problem konfrontiert, dass grössere Dateien zwischen Server und Benutzer übermittelt werden müssen. Zu Beginn waren wir nicht sicher, wie gross die Dateien bei Up- und Download über HTTP sein dürfen. Bei Recherchen sind wir immer wieder darauf gestossen, dass das Protokoll HTTP keine Beschränkungen hat, aber viele Server standardmässig eine Uploadgrössenlimitation von ca. 2-4 GB eingebaut haben. Da die Daten vielleicht sogar direkt bei einer Messung "im Feld" bei schlechter Verbindung übertragen und visualisiert werden sollen, haben wir uns schlussendlich für eine Lösung mit einzelnen Chunks entschieden. So werden die grossen Dateien in kleine Teilstücke geteilt und in mehreren Requests auf den Server hochgeladen. Das hat auch den Vorteil, dass im Frontend ein genauer Uploadfortschritt angezeigt werden kann. Diese Option bietet auch die Möglichkeit, den Upload zu pausieren und zu einem späteren Zeitpunkt fortzufahren. Der Download der Punktwolke und auch die des GeoTIFFs wurde über einen einfachen `` HTML-Tag gelöst. Genauere Informationen zum Vorgang inklusive Diagrammen befinden sich im technischen Bericht im Kapitel 5.2.

2D GeoTIFF Anzeige:

Die GeoTIFF-Anzeige war nicht von Beginnweg so geplant. Bereits bei der Berechnung haben wir gehofft mit automatischen Klassierungen, wie Boden und nicht Boden, noch bessere Resultate zu erreichen. Die entsprechenden PDAL-Filter lieferten aber keine sinnvollen Resultate. Die Abbildung II.9 zeigt den Versuch mit dem `Cloth Simulation Filter (CSF)`. Deshalb wurden, nachdem die Installationsschwierigkeiten überwunden waren, auch viele PCL-Funktionen getestet. Das Einlesen und Ausgeben der Punktwolke ist uns relativ schnell gelungen. Die Feinjustierung der Parameter in den C++-Programmen waren für uns aber extrem schwer und auch der Berechnungsaufwand und die Rechenzeit waren nicht vertretbar. Zu der langen Rechenzeit wäre auch noch die Zeit dazugekommen, um die Daten in das von PCL unterstützte PCD-Format umzuwandeln. Damit war klar, dass dies so nicht in den Prototypen implementiert werden konnte.

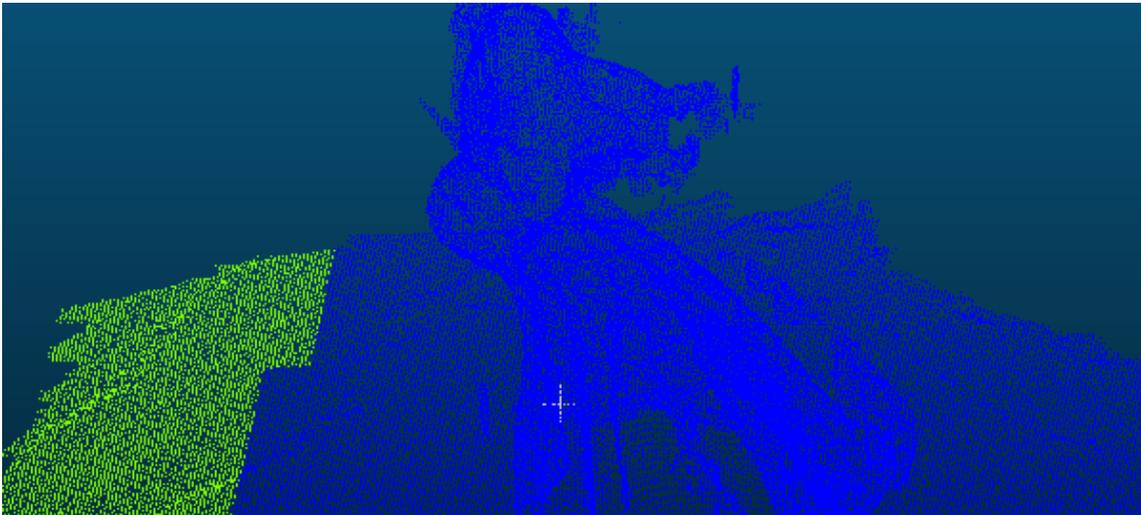


Abbildung II.9: PDAL Fehlversuch der Funktion Cloth Simulation Filter (CSF)

In der Abbildung II.10 ist der Versuch abgebildet, über ein Range Image eine 2D-Abbildung, zum Beispiel für die Denkmalpfleger, zu erzeugen. Dieser Versuch sollte nicht eine Aufsicht zeigen, sondern die Fassade eines Hauses. Da wir keine Stativaufnahme als Grundlage haben, konnten wir den Viewpunkt und den Blickwinkel nicht sinnvoll definieren. Dementsprechend ist nichts von der aufgenommenen Fassade und der Veranda erkennbar.

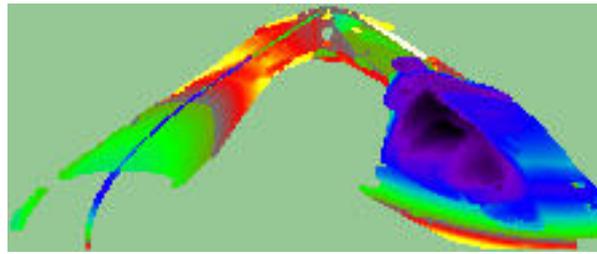


Abbildung II.10: PCL Fehlversuch der Funktion range image

Bei weiteren Funktionen wie region growing segmentation oder difference of normals based segmentation, die Strukturen anhand der Normalen erkennbar machen sollten, haben wir keine sinnvollen Resultate erhalten. Entweder ist die Berechnung abgebrochen oder hat Resultate ohne Mehrwert erzeugt. Das nachfolgende Bild II.11 bildet den Vorgang der difference of normals based segmentation Funktion ab. Mit einem GeoTIFF Export der Aufsicht hätten aus dem letzten Schritt die Strukturen in eine 2D-Ableitbare überführt werden können. Diese und weitere Versuche sind im Anhang 3 dokumentiert.

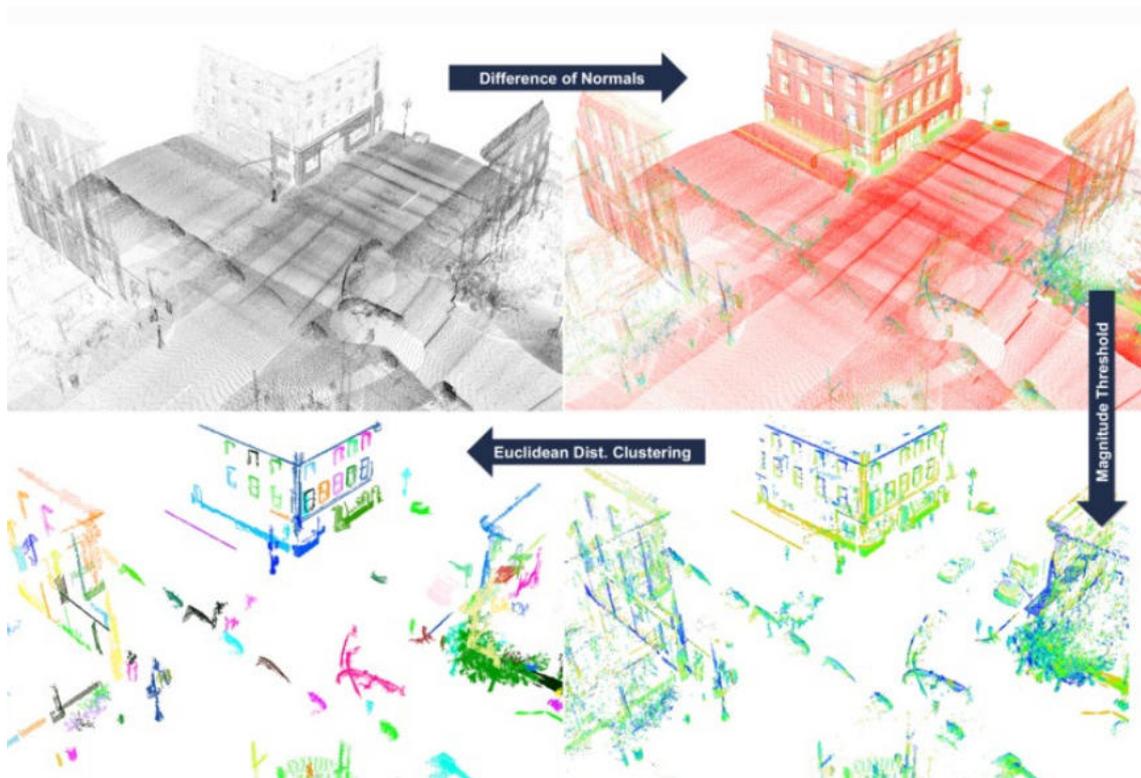


Abbildung II.11: difference of normals based segmentation Vorgang[23]

Wegen dem Zeitdruck, der schlechten Performance und den vielen Abstürzen wurde

versucht, eine möglichst hochauflösende DOM Berechnung über PDAL zu erreichen. Hierbei wird eine Punktwolke im LAZ Format gelesen. Mit einem k-Means-Algorithmus werden Outliers der Klasse 7 klassiert und im Anschluss eliminiert. Um das GeoTIFF zu schreiben, wird ein GDAL writer verwendet. Mit den Parametern resolution und radius kann die gewünschte Auflösung definiert werden. Das Code Listing II.1 erzeugt eine zu kleine Auflösung wie im Bild II.12 zu erkennen ist. Das Code Listing II.2 ergibt dann eine zu hohe Auflösung wie im Bild II.13 zu sehen ist. In unserer Pointcloudpipeline wird jede GeoTIFF Berechnung über die Parameter des Code Listing II.3 und der Abbildung II.14 gemacht, um Fehlinterpretationen zu vermeiden.

Listing II.1: Test mit zu kleiner Auflösung

```
"pipeline": [  
  "input_file.laz",  
  {  
    "type": "filters.outlier",  
    "method": "statistical",  
    "mean_k": "12",  
    "multiplier": "1.0"  
  },  
  {  
    "type": "filters.range",  
    "limits": "Classification ![7:7]"  
  },  
  {  
    "filename": "output_file.tif",  
    "resolution": "1.0",  
    "output-type": "max",  
    "radius": "3.0",  
    "type": "writers.gdal"  
  }  
]
```

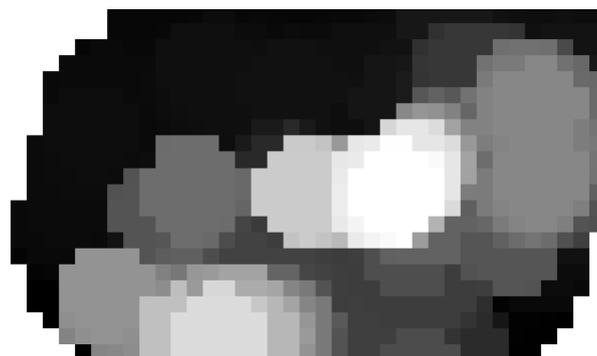


Abbildung II.12: Test mit zu kleiner Auflösung

Listing II.2: Test mit zu grosser Auflösung

```
"pipeline": [  
  laz_input_file ,  
  {  
    "type": "filters.outlier",
```

```

    "method": "statistical",
    "mean_k": "12",
    "multiplier": "1.0"
  },
  {
    "type": "filters.range",
    "limits": "Classification![7:7]"
  },
  {
    "filename": "output_file.tif",
    "resolution": "0.05",
    "output_type": "max",
    "radius": "0.05",
    "type": "writers.gdal"
  }
]

```



Abbildung II.13: Test mit zu grosser Auflösung und zu vielen Strukturen

Listing II.3: Umwandlung GeoTIFF zu JPEG

```

"pipeline": [
  laz_input_file ,
  {
    "type": "filters.outlier",
    "method": "statistical",
    "mean_k": "12",
    "multiplier": "1.0"
  },
  {
    "type": "filters.range",
    "limits": "Classification![7:7]"
  },
  {
    "filename": "output_file.tif",
    "resolution": "0.05",
    "output_type": "max",
    "radius": "0.1",
    "type": "writers.gdal"
  }
]

```



Abbildung II.14: Verwendete Auflösung

Auch bei der Darstellung im Browser hätten wir einen Backup-Plan bereit gehabt, wenn die TIFF-Darstellung wie im Kapitel I.5.4 beschrieben, nicht in letzter Sekunde noch geklappt hätte. Diese sah vor, den react-images-viewer mit Zoom, Rotier- und Verschiebungsmöglichkeit im Frontend zu implementieren und die GeoTIFF Dateien über einen Pythonskript Code im Backend (Listing II.4) in ein JPEG umzuwandeln. Der Vorteil wäre gewesen, dass die Datei wesentlich kleiner und die Ansicht dynamisch gewesen wäre. Die Berechnung hätte aber noch einmal einige Zeit gedauert und hätte auch wieder Speicherplatz beansprucht. Ebenso hätte das Skript zum automatischen Abräumen lange nicht gebrauchter Dateien eine Erweiterung gebraucht, um auch die JPEG Dateien zu löschen.

Listing II.4: Caption hinzufügen

```
import imageio
import os
import numpy

def getJpeg( file ):
    image = imageio.imread( file )
    minValue = numpy.amin(numpy.array(image)[image != numpy.amin(img)])
    maxValue = numpy.max(image)
    resetNoDataValue = numpy.where(image < minValue, minValue, image)

    with numpy.nditer( resetNoDataValue, op_flags=['readwrite'] ) as it:
        for x in it:
            x[...] = int( ((255/(maxValue-minValue)x)-255/(maxValue-minValue)minValue)

    imageio.imwrite( os.path.splitext( os.path.basename( file ) )[0] + ".jpg", resetNoDataValue.astype(numpy.uint8) )

    return [ numpy.amin( resetNoDataValue ), numpy.max( resetNoDataValue ) ]
```

3.3 UI Design

Beim Design unserer Singlepage Applikation war uns wichtig, dass Leute mit GIS-Erfahrung ihren gewohnten Aufbau vorfinden. Das bedeutet unter anderem, dass die Punktwolke grösstmöglichst dargestellt werden muss.

Damit sich GIS-Experten schnell zurechtfinden, wurde die Auswahl der Punktwolken links in einem wegklickbaren Fenster untergebracht, wie es in den meisten Online-Geoportalen üblich ist (zum Beispiel maps.zh.ch oder map.geo.admin.ch). Da mit dem Potree Viewer recht viele Konfigurationsmöglichkeiten mitgeliefert wurden, wurden diese in ein einklappbares Fenster rechts im Bild eingebunden. Dabei haben wir uns mehr an den GIS-Browser des Kanton Zürichs angelehnt, der auch eine grosse Auswahl in der Konfigurationsleiste bietet. Die Metadaten wurden ebenfalls auf der rechten Seite über den Konfigurationsmöglichkeiten platziert. Um in die 2D-Ansicht zu wechseln, wurde ein entsprechender Button unterhalb der 3D-Konfiguration ergänzt. Das ist Usability-technisch besser als die Unterbringung eines Togglebuttons innerhalb des wegklickbaren Bereichs, weil der Benutzer so auch im zugeklappten Zustand sofort erkennt, was ausgewählt und möglich ist. Die Punktwolken sollte man auch vom Server herunterladen, bearbeiten und löschen können. Das haben wir bewusst in den rechten Kontext gesetzt, da dies Änderungen an genau der dargestellten Punktwolke bewirkt.

Upload, Login und Logout sollten von Beginn weg als Modals umgesetzt werden. Nach dem ersten Usabilitytest hat sich vor allem gezeigt, dass die Uploadfunktion bereits für den nicht registrierten Benutzer sichtbar sein muss. Deshalb wurde er in die Ansicht integriert und verweist den nicht eingeloggten Benutzer auf das Login.

In Abbildung II.15 befindet sich das Wireframe, mit dem ein erster Usability-Test gemacht wurde. In Abbildung II.16 befindet sich das finale Design des Prototypen.

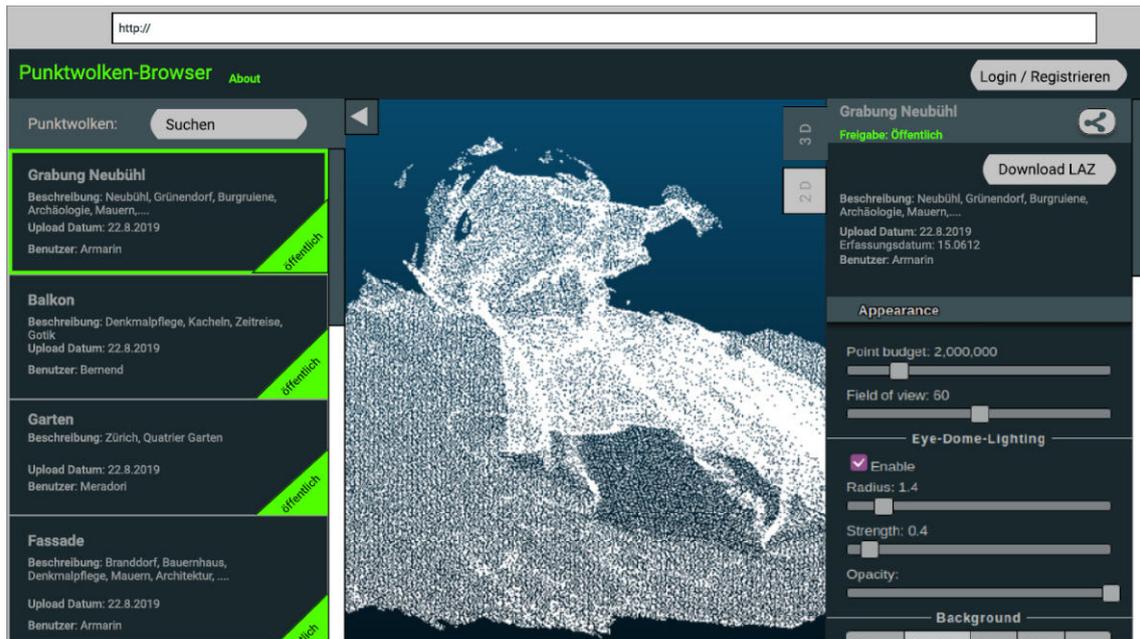


Abbildung II.15: Wireframe Design



Abbildung II.16: Design am Ende

4 Testing

4.1 Automatische Testverfahren

Der Point Cloud Browser und die Pointcloudpipeline verfügen über Unittests. Die Abdeckung in den einzelnen Komponenten ist wie folgt:

Point Cloud Browser Frontend: Stmts - 36.19%, Branch - 59.81%, Funcs - 21.21%

Point Cloud Browser Backend: Mit coverage getestet - 38%

Pointcloudpipeline: 95%

Die Resultate der Performancetests befinden sich in Anhang 4.4.

Zudem haben wir bei unseren Mergerequests in der GitLab Pipeline eine Sonarqubeanalyse aktiviert. Mit Hilfe von Sonarqube sind uns vor allem Codesmells und Verstöße gegen die coding convention bewusst geworden. So wurde unser Code schöner und stabiler.

4.2 Manuelle Testverfahren

Die manuellen Testverfahren bestehen aus zwei Usabilitytests und vier Systemtests.

Die Usabilitytests wurden zwei Mal mit den gleichen Personen durchgeführt. Der erste Test wurde im Figma und der Zweite an der Endversion durchgeführt. Die zweite Durchführung war bereits bei der Alpha Version geplant, da wir aber einige Hindernisse zu bewältigen hatten, haben wir diesen bewusst nach hinten zur Endversion geschoben. Nichtsdestotrotz sind beide Tests erfreulich verlaufen. Die Testpersonen waren am Ende, bis auf kleine Mängel, begeistert von unserem Resultat. Wir haben sehr viel durch die Usabilitytests erfahren und konnten von den hilfreichen Verbesserungsvorschlägen und Tipps profitieren. Eine komplette Auswertung inklusive den Protokollen befinden sich im Anhang 4.3.

Bei den Systemtests wurden von uns alle Hauptaktivitäten durchgeführt, sofern sie bereits verfügbar waren. Die entsprechenden Protokolle befinden sich im Anhang 4.2. Für die

Pointcloudpipeline wurde kein separater System-Test durchgeführt, sondern anhand der Funktionen (Upload, 3D-View und 2D-View) im Point Cloud Browser geprüft, da diese auf den Funktionen der Pointcloudpipeline basieren.

5 Resultate und Weiterentwicklung

5.1 Resultat

Delivery 1

Der Vergleich der Geräte konnte erfolgreich durchgeführt werden. Alle drei Messungen und die Referenzmessungen konnten erfolgen. Die Auswertung hat auch klare Vor- und Nachteile sichtbar gemacht. Da die Tests alle bei besten Bedingungen durchgeführt werden konnten (kein Regen oder Schnee), konnten sie auch gut untereinander verglichen werden. Wir sind mit dem Resultat des Vergleiches zufrieden und denken, dass die ermittelten Resultate, die im Kapitel I.4.9 dokumentiert sind, für einige Entscheidungen im Bereich Messtechnik als Grundlage verwendet werden.

Alle Punktwolken wurden in CloudCompare visualisiert und die der Burgruine sogar gereferenziert. Viele der Punktwolken haben wir auch im Potree-Viewer betrachtet. Im technischen Bericht wurden die Unterschiede von Meshes und Punktwolken beleuchtet, sowie verschiedene Speicherformate einander gegenübergestellt. Dabei wurden einige Dateien konvertiert und die Merkmale der Formate dokumentiert.

Delivery2

Der Point Cloud Browser war als Prototyp gedacht, könnte aber bereits eingesetzt werden. Er erlaubt das Hochladen, Verwalten, Visualisieren und Herunterladen von Punktwolken. Gäste oder registrierte Benutzer können den Prototypen verwenden, wobei der registrierte Benutzer mehr Rechte besitzt. Hochladen, Bearbeiten und Löschen von Punktwolken steht nur dem registrierten Benutzer bei seinen eigenen Punktwolken zur Verfügung. Beim Bearbeiten kann er allerdings lediglich die Metadaten im System verändern oder die Punktwolke austauschen. Für die Bearbeitung der Punktwolke steht eine Down-

und erneute Uploadfunktion zur Verfügung. Dadurch kann man die Bearbeitung in den bekannten Programmen wie CloudCompare durchführen und die Punktwolke im Tool ersetzen. Anders als der Gast hat der registrierte Benutzer nicht nur Zugriff auf die öffentlichen Punktwolken, sondern auch auf die ihm Freigegebenen oder Privaten. Ein Gast ist somit nicht wirklich eingeschränkt, da auch er Punktwolken visualisieren und herunterladen kann. Die Visualisierung kann intuitiv rotiert, gezoomt und verschoben werden und erlaubt das Messen von Distanzen und Winkeln. Zudem konnten viele weitere Funktionen von Potree übernommen werden. Eine 2D-Ansicht wird ebenfalls angeboten und kann als GeoTIFF exportiert werden. Was nicht erreicht wurde, war das Teilen einer Visualisierung via URL. Angedacht war, dass man die Visualisierung mit der eingestellten Punktdichte, den Markierungen, der Zoomstufe, dem Blickwinkel und weiteren Einstellungen mit anderen teilen kann.

Auch bei den nichtfunktionalen Anforderungen konnten wir alle wesentlichen Punkte abdecken. Der Prototyp macht aber noch kein Backup der LAZ-Daten. Dies ist aber bei einem produktiven Einsatz einfach zu ergänzen, da die Daten persistent in Ordnern liegen und beim Stoppen der Dockercontainer nicht abgeräumt werden. So kann auch jeder das Backup an seine Umgebung und spezifischen Anforderungen anpassen. Die Performance ist im Zusammenspiel mit den Benutzerrückmeldungen trotz den riesigen Daten und aufwändigen Berechnungen angemessen.

Delivery 3

Die Pointcloudpipeline ist ein Commandline-Tool zur 3D-Punktwolkenkonvertierung in eine 2D-Ansicht. Zusätzlich kann es Python Package via PyPi installiert und importiert werden. Das Tool kann somit nicht nur über die Commandline verwendet werden, sondern auch in weitere Projekte importiert werden. Auch unser Point Cloud Browser verwendet für das Management und Konvertieren der Daten die Pointcloudpipeline.

Neben dem erweiterten Einsatzbereich wurde auch die Funktionalität erweitert. Neben der Konvertierung von der 3D-Punktwolke ins 2D-GeoTIFF wurden folgende Funktionen eingebaut:

- Daten können aus e57, PLY und LAS in LAZ konvertiert werden

- Daten können aus LAZ in e57, PLY und LAS konvertiert werden
- LAZ Daten können in EPT-LAZ Daten konvertiert werden

Dadurch hat sich die Pipeline zu einem Tool entwickelt, das das komplette Archivierungsmanagement von LAZ-Dateien erlaubt. Das Tool kann somit verwendet werden, um die Langzeitspeicherung von Punktwolken im weitverbreiteten und komprimierten LAZ-Format zu managen, aber diese bei Bedarf auch wieder zurück oder in ein browseroptimiertes Darstellungsformat zu konvertieren.

Die erarbeiteten nichtfunktionalen Requirements konnten alle eingehalten werden.

5.2 Weiteres Vorgehen

Die Resultate des Projekts, werden sicher in der Zusammenarbeit mit dem Amt der Archäologie des Kanton Thurgau weitergeführt. Der Auftrag steht lose im Zusammenhang mit einer Studie, die das IFS im Auftrag dieser tätigt.

Die Resultate des Gerätevergleiches, aber auch die Publikationsmöglichkeit über den Point Cloud Browser werden wir gerne zusätzlich nach der Arbeit dem entsprechenden Fachpublikum präsentieren. In welcher Form wird noch abgeklärt.

Der Prototyp hat noch viel Erweiterungspotential. Diese Möglichkeiten wurden im Teil I.6.2 beschrieben. Die Verbreitung des Point Cloud Browser und die Umsetzung der dokumentierten oder eigenen Erweiterungen wird unter der BSD-2 Lizenz erfolgen müssen.

6 Projektmanagement

6.1 Stakeholder

Denis Nauli und Nadine Sennhauser haben als Projektteam den grössten Einfluss in dieser Arbeit. Weitere Personen und Personengruppen können zudem Einfluss auf den Verlauf des Projekts nehmen.

Das können unter anderem Personen sein, welche uns die Geräte leihen, oder aber auch Messobjektverantwortliche, die jederzeit abspringen könnten. Zudem sollten die Archäologen davon einen Mehrwert erfahren und zählen deshalb zu den einflussreichsten und interessiertesten Benutzergruppen. Auch die Visions Nutzergruppen wie GIS-Spezialisten und Hobby Mapper können Einfluss auf gewisse Entscheide haben. Speziellen Einfluss hat auch Daniel Mayer, der uns seine Daten zum Vergleich zur Verfügung stellen möchte. Nicht zu vergessen ist der Einfluss und die Unterstützung durch unseren Betreuer Stefan Keller. Weiteres ist der Abbildung II.17 zu entnehmen.



Abbildung II.17: Stakeholder Map

6.2 Zeitplan

Der Gesamtaufwand beläuft sich auf rund 720 Stunden. Das ergibt pro Mitglied und Woche ca. 21.2 Stunden. Dabei wird mit 17 Wochen gerechnet, die in die folgenden Iterationen unterteilt wurden:

Nr	Name	Dauer	Inhalt	Assessment
0	Initialize	22.02.21 - 26.02.21	Dokumentationsvorlage, Dokumentationsrepositary	Allgemeine Projekt- und Dokumentations- struktur festgelegt
1	Inception	26.02.21 - 12.03.21	Requirements erarbeiten	Auftrag, Requirements, UseCases, Userprofiles
2	Elaboration A	12.03.21 - 26.03.21	Research	Analyse bestehender Tools und Technologien, Wireframes
3	Elaboration B	26.03.21 - 09.04.21	Architektur erarbeiten	Design, Architekturrentscheide getroffen, Detail-Konstruktionsplan
4	Prototyp	09.04.21 - 23.04.21	Front- und Backend Kommunikation	Prototyp
5	Alpha	23.04.21 - 07.05.21	Alpha konstruieren	Alpha Version
6	Beta	07.05.21 - 21.05.21	Beta konstruieren	Beta Version
7	End Version	21.05.21 - 04.06.21	Final konstruieren	Finale Version
8	Documentation	04.06.21 - 18.06.21	Dokumentieren, Abstract und Poster erarbeiten	Hochladen der Dokumente auf archiv- i.hsr.ch, Abgabe an Dozenten

Tabelle II.5: Projektplan Iterationen

Der Konstruktionsplan des Point Cloud Browsers befindet sich in Anhang 6.2. Ebenso können die Protokolle der wöchentlichen Besprechungen im Anhang 6.1 eingesehen werden.

6.3 Risiken

Die Risiken wurden während des gesamten Projektes überwacht und mit passenden Aktionen minimiert. Wie immer kann man nicht alle Risiken mitigieren und ausschliessen. Deshalb sind auch bei uns ein paar der Ereignisse eingetroffen. Durch klar definierte

Massnahmen konnten wir mit den kleinen Rückschlägen gut umgehen. Die Risiken der Inception Phase, sowie die am Ende sind hier abgebildet. Die komplette Risikoanalyse befindet sich im Anhang 5. Zu beachten ist dabei, dass die Gewichtung und der Schaden immer vor der Ergreifung der vorbeugenden Massnahmen beurteilt wurde.

Stand der Risiken Ende von Inception (Abbildung II.18)

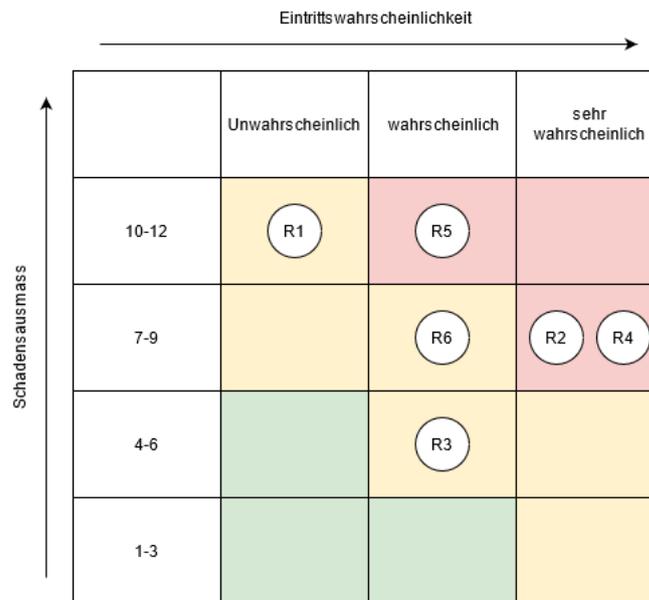


Abbildung II.18: Risiken Inception Phase

Stand der Risiken Ende von Final (Abbildung II.19)

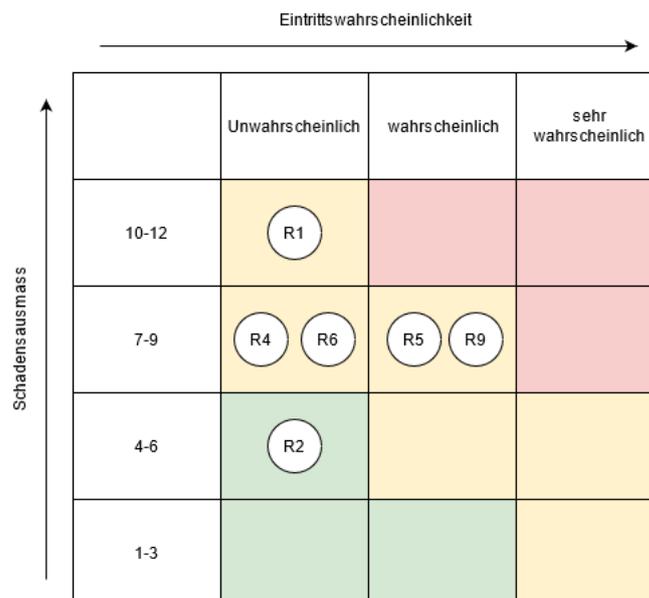


Abbildung II.19: Risiken Final Phase

An diesen Beispielen erkennt man, dass bis zum Schluss an den Risiken gearbeitet werden muss, da sich diese während des Projekts stetig verändern können.

7 Projektmonitoring

7.1 Ereignisse

Akku:

Die Messungen verliefen fast problemlos. Nur bei der Ausleihe der Totalstation war der Akku beim Bedien-Tablet leer. Zum Glück haben wir die Abholung eine Stunde vor Messbeginn geplant. Somit konnten wir das Tablet noch soweit aufladen, dass der Akku gerade für die geplanten Referenzmessungen an der Villa Aurum ausgereicht hat.

GitLab:

Leider sind die langen Wartezeiten und die vielen 500 Errors von GitLab für ein effizientes Arbeiten nicht gerade förderlich. Auch das CI/CD konnte nicht wie gewünscht konfiguriert werden, da keine Verbindung zum Deployment-Server hergestellt werden konnte. Nach langem Ausprobieren und mit Hilfe von Nicola Jordan, haben wir das Problem auf "nicht in unserem Einflussbereich" eingestuft. So musste das Deployment auf den Server immer von Hand durchgeführt werden, was aber während der kurzen Zeit dieser Bachelorarbeit nicht besonders störend war. Der Zeitverlust durch den Konfigurationsaufwand war da schon deutlich störender.

Potree-Viewer:

Bis der Viewer zu 100% in unserem Frontend integriert war, haben wir leider bereits etliche Stunden mehr aufwenden müssen, als geplant war. Bei der Evaluation wurde zu wenig genau darauf geachtet, dass die Einbindung via iFrame in unserem Aufbau eine denkbar schlechte Lösung war, obwohl es auf den ersten Blick verlockend und einfach wirkte. Mit Unterstützung und einem bestehenden GitHub-Issue mit der Beschreibung unseres Problems, konnten auch wir den statischen Viewer verwenden, um die Punkt-

wolken unseres Point Cloud Browsers dynamisch anzuzeigen.[20]

Chunkedupload:

Die Frage, wie wir mit den grossen Dateien und der entsprechend langen Verarbeitungszeit umgehen, hat uns auch einige Zeit gekostet. Bisher hatten wir keine Erfahrungen mit dem Übermitteln von potentiell dutzenden von GB von Daten. In dem Bereich war auch das Testing nicht sehr einfach und die Dokumentationen von Django nicht sehr aufschlussreich. Aber auch das Testing hat am Ende geklappt.[21]

TIFF Viewer:

Darauf wurde bereits in Kapitel II.3.2 und I.5.4 eingegangen. Die Implementation von Plotty und dem GeoTIFF reader geotiff.js hat uns wegen unserer Webpackage 5 Version einige Sorgen bereitet und wir hatten deswegen sogar bereits einen funktionierenden Backup-Plan. Aber nachdem wir den Grund für das Problem gefunden hatten, war nur noch das bereits aus dem Backup-Plan bekannte Problem der passenden Einfärbung, statt mit Python nun im Javascript, zu lösen.

Vergleichsdaten:

Für den Vergleich mit den bestehenden Methoden haben wir von Herrn Mayer vom Amt für Amt für Archäologie des Kanton Thurgaus Messdaten aus dem Aufnahmeverfahren Structure from Motion erhalten. Leider sind diese erst in der letzten Woche eingetroffen. Dadurch mussten wir noch einmal einige zusätzliche Zeit in die Dokumentation investieren, da diese Auswertung doch eine zentrale Rolle bei der Beurteilung der Handlaser-scanner spielte.

Zeitplan:

Durch die oben beschriebenen Ereignisse wurde natürlich auch unser enger Zeitplan zum Problem. Aber mit dem Streichen eines nicht sehr relevanten Requirement und einer leichten Zeitüberschreitung konnte dennoch ein umfangreicher und funktionierender Prototyp abgegeben werden.

7.2 Soll-Ist-Zeit-Vergleich

Die aufgewendete Zeit übersteigt das Budget von 720h um insgesamt 24 h 25 min. Dies kam zustande, weil wir einigen Risiken mit mehr Aufwand entgegenwirken mussten. Die Abbildung II.20 ermöglicht es, den Aufwand pro Phase zu erkennen und zeigt deutlich, dass wir in der Betaphase zum Beispiel wegen der Umsetzung und Testing der Chunked Uploads mehr als geplant leisten mussten. Bei der Dokumentation musste durch die späte Lieferung der Vergleichsdaten in der letzten Woche auch noch mehr Zeit investiert werden, als geplant war. Die Abbildung II.21 zeigt, dass uns das Aufteilen der Arbeit untereinander sehr gut gelungen ist. Abbild II.22 zeigt, wie viel Zeit wir in welche Tätigkeit investiert haben.

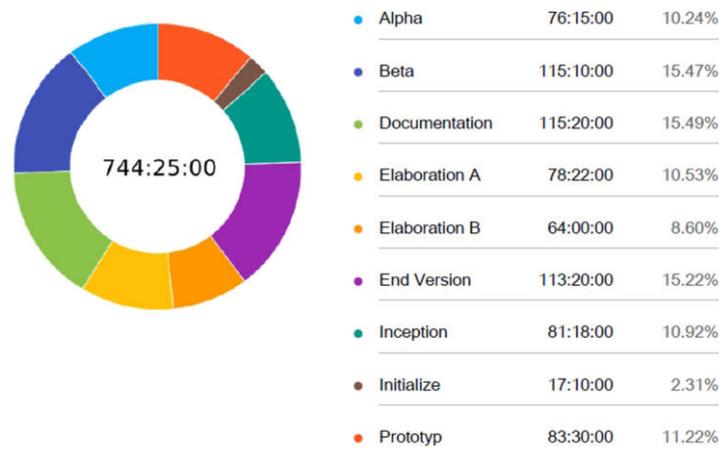


Abbildung II.20: Zeitaufwendung pro Phase

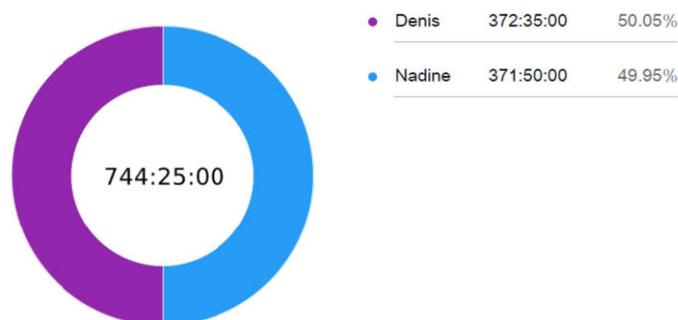


Abbildung II.21: Zeitaufwendung pro Teammitglied

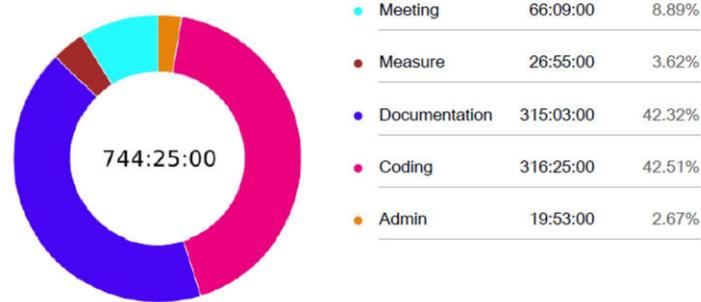


Abbildung II.22: Zeitaufwendung pro Aufwandstyp

7.3 Codestatistik

Die Zahlen des Point Cloud Browsers ohne Tests:

Codezeilen insgesamt: 3733

Die Zahlen der Pointcloudpipeline ohne Tests:

Codezeilen insgesamt: 300

In Abbildung II.23 ist die SonarQube-Übersicht des Point Cloud Browsers ersichtlich.

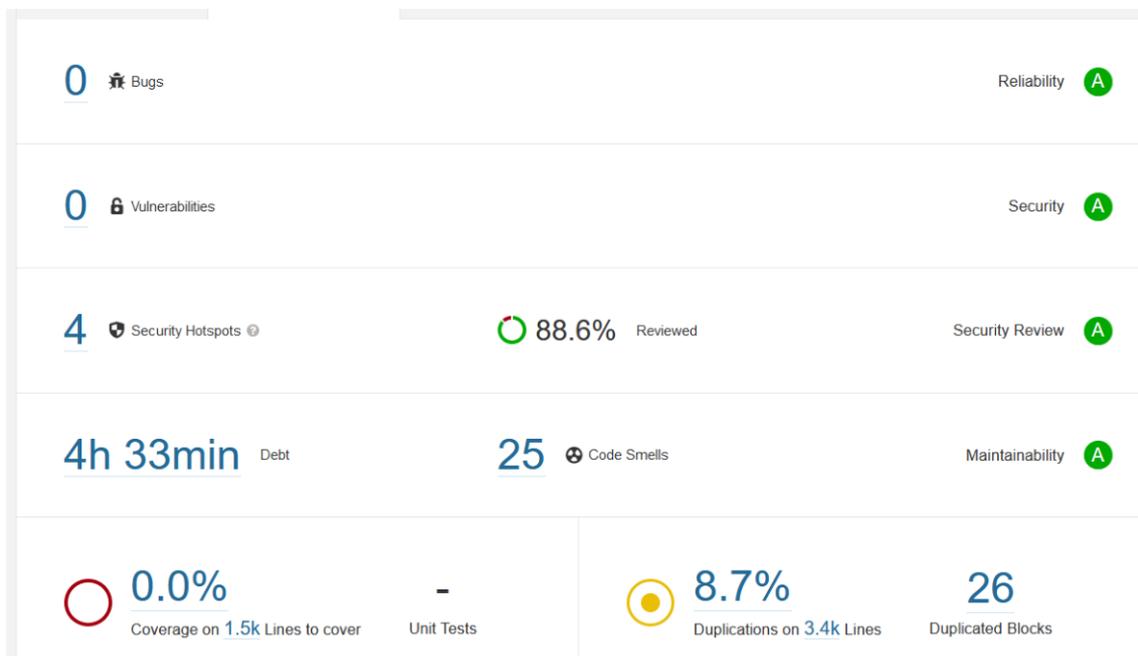


Abbildung II.23: SonarQube-Übersicht

7.4 Fazit

Die Welt der Geodaten und speziell die der 3D-Daten ist sehr faszinierend. Im Gegensatz zu anderen Daten können sie visuell dargestellt werden und ermöglichen zum Beispiel die Erkundung von weit entfernten Gegenden, vorher nie gesehene Perspektiven und interessanten Objekten von längst vergangenen Zeiten. Wir sind der Meinung, dass gerade die Daten der Archäologie einen grossen Reiz für viele in der Bevölkerung haben. Deshalb hoffen wir, dass aufwändig erfasste 3D-Punktwolken nicht nur auf unzugänglichen Servern gelagert werden, sondern durch den Point Cloud Browser weiter verbreitet und geteilt werden.

Der Vergleich der Messgeräte hat uns gezeigt, dass ein SLAM basierter Handlaserscanner nicht nur ein Spielzeug ist, sondern dem Wunsch nach schnell erfassten und genauen Daten nachkommt. Auch wir sind von der Technologie in den Bann gezogen worden und freuen uns darauf, wenn wir ein mobiles Gerät mit integriertem Laserscanner unser Eigen nennen dürfen. Die Genauigkeit des iPads ist zurzeit zwar sehr beschränkt, aber für den persönlichen Gebrauch völlig ausreichend. In einem Test in den eigenen vier Wänden haben wir zudem festgestellt, dass diese auch für eine Wohnungsbesichtigung zu Lockdownzeiten eingesetzt werden könnte. Da unser Fokus aber auf dem Aussenbereich lag, wurde dies nicht weiter verfolgt.

Wir sind stolz auf unsere Arbeit. Obwohl wir einige ungeplante Umwege einschlagen und den einen oder anderen Rückstoss abfangen mussten, haben wir ein gutes Endresultat vorzuweisen, was auch unseren angewendeten Projektmanagementfähigkeiten, unserer Flexibilität und der angenehmen Betreuung und Unterstützung seitens IFS zu verdanken ist. Wir konnten alle Kernanforderungen abdecken und hatten die Möglichkeit eine grossartige Technologie kennenzulernen und uns weiterzuentwickeln. Neben dem Einblick in die Archäologie und Messtechnik, konnten wir durch den Prototypen auch viele neue und bedeutende Erfahrungen in der Softwareentwicklung sammeln. Danke.

8 Softwaredokumentation

8.1 Installationsanleitung und Benutzerhandbuch

Die Installationsanleitungen des Point Cloud Browsers und der Pointcloudpipeline sind im jeweiligen README des entsprechenden GitLab Repository abgelegt. Dadurch können diese auch bei Erweiterungen aktualisiert werden und bleiben aktuell. Ein Auszug des aktuellen Standes der Installationsanleitungen und Benutzerhandbücher findet man zusätzlich in den Anhängen 9 für den Point Cloud Browser und 10 für die Pointcloudpipeline.

8.2 Schnittstellendokumentation

Die Schnittstellendokumentation wurde mit Swagger automatisch generiert und ist im Point Cloud Browser unter /swagger abrufbar. Ein Screenshot zum aktuellen Stand findet sich in Abbildung II.24.

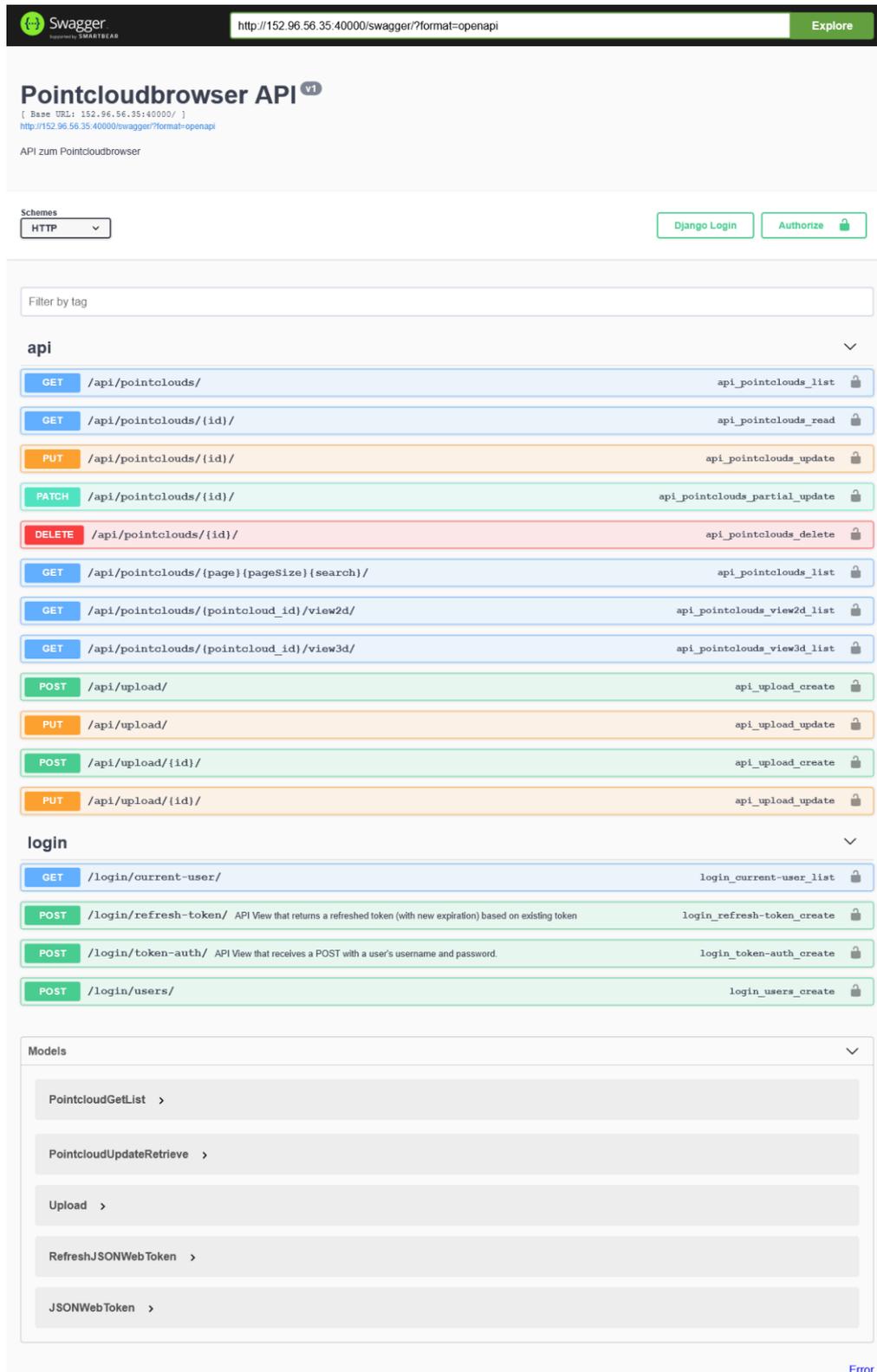


Abbildung II.24: Schnittstellendokumentation mit Swagger

Anhang

1 Evaluation

1.1 Recherche Viewer

Research View

Hier werden webbasierte Open-Source Pointcloud-Viewer und ein paar bekannte Packages, die im Zusammenhang mit 3D Webvisualisierungen von Pointclouds stehen, angeschaut.

Da wir opensource Software zum Einbinden in unser Projekt suchen, werden bekannte webbasierte Closed-Source Pointcloud-Viewer Kurz erwähnt, aber nicht genauer analysiert.

Open-Source Pointcloud Viewer

Potree

- Open Source [Lizenz](#)
- WebGL basiert (OpenGL + Browser = WebGL)
- Unbedingt lesen: <https://www.cg.tuwien.ac.at/research/publications/2016/SCHUETZ-2016-POT/SCHUETZ-2016-POT-thesis.pdf>
- LAS, LAZ und custom binary format (File format of Scanopy) -> EPT LAZ/ LAS geht auch
- Each octree node contains a point cloud model that is stored in its own file on disk.

Schon von Potree erfüllt, werden folgende von uns definierte Requirements:

Nr	Anforderung
3	Punktwolken können in 3D visualisiert werden.
13	In den Punktwolken-Visualisierungen kann gezoomt, rotiert und verschoben werden.
14	In den Punktwolken-Visualisierungen können Strecken, Distanzen und Abstände gemessen werden.

Ein Login und Benutzer mit Benutzerverwaltung gibt es in Potree nicht. Somit ist es auch nicht möglich als Benutzer Punktwolken hochzuladen. Die Punktwolken werden vom Admin der Seite in einem Ordner abgelegt und zu jeder Punktwolke eine passende (statische) HTML-Seite erstellt.

Aus dem Paper: "Potree's structure is based on a slightly adapted modifiable nested octree (MNO)structure, which was introduced, and built into the Scanopy point cloud renderer, by Scheiblauer"

Funktionen zur Visualisierung:

The image displays a software interface for 3D point cloud processing, organized into several panels:

- Appearance Panel:** Contains settings for 'Point budget' (2,000,000), 'Field of view' (60), 'Eye-Dome-Lighting' (Enabled, Radius: 1.4, Strength: 0.4, Opacity), 'Background' (Skybox, Gradient, Black, White, None), 'Other' (Splat Quality: Standard/High Quality, Min node size: 30, Box, Lock view).
- Tools Panel:** Includes 'Measurement' (Show/Hide labels), 'Clipping' (Clip Task: None/Highlight/Inside/Outside; Clip Method: Inside Any/Inside All), 'Navigation' (Camera Projection: Perspective/Orthographic; Speed: 417.8).
- Scene Panel:** Shows 'Export' options (JSON, DXF, Potree) and a tree view of 'Objects' (Point Clouds, lion, Measurements, Annotations, Other, Camera, Vectors, Images) and 'Properties'.
- Filters Panel:** Features a 'Classification' list with checkboxes and color swatches for categories like 'never classified', 'unclassified', 'ground', 'low vegetation', 'medium vegetation', 'high vegetation', 'building', 'low point(noise)', 'key-point', 'water', 'overlap', and 'default'. It also includes an 'invert' button, 'Returns' (Return Number: 0 to 7, Number of Returns: 0 to 7), 'Point Source ID' (four sliders, each with a range of 0.000 to 65,535.000), and 'GPS Time'.

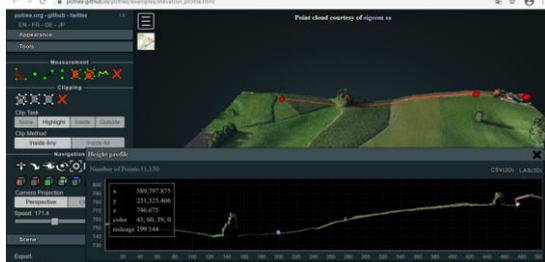


Potree ist ein solider Viewer, mit dem man nicht nur seine Visualisierung anpassen, sondern auch Messungen



verschiedenster Art durchführen kann.

Höhenprofil anzeigen und in 2D (CSV) oder 3D (LAS) runterladen:

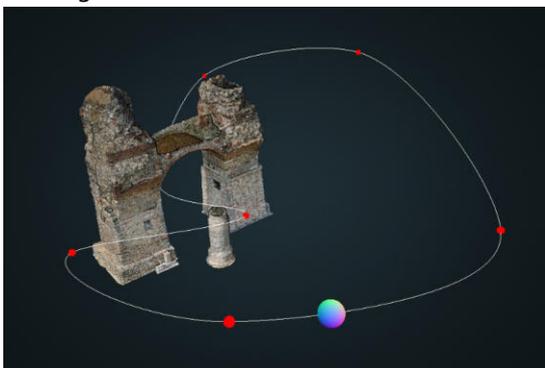


Potree verwendet folgende Libraries, welche später hier noch angeschaut werden:

- D3
- ThreeJs
- Cesium (wird für die Weltkugeldarstellung verwendet, siehe weitere Funktionen)

Funktionen, die interessant/ praktisch sind, aber nicht in unseren Requirements sind:

- Annotations: Feststellung: Auch die Annotations sind statisch. Für z.B. <https://potree.github.io/potree/examples/annotations.html> gibt es die Files `gradient_colors.html` und `annotations.html` (vielleicht noch weitere?). Das müsste alles dynamisch gemacht werden, falls der Benutzer selber Annotations oder Ähnliche Funktionen benutzen können soll.
- Weltkugel einbinden (Mit CesiumJS) https://potree.github.io/potree/examples/cesium_sorvilier.html
- Man kann auch Sachen animieren, aber die Beispiele von Potree funktionieren bei mir nicht - beim selber gehosteten Potree auf Localhost in einer Linux-VM ging es.



Die Kugel bewegt sich entlang der Linie. Was das bringt bin ich mir nicht ganz sicher. Hier ist es schon klarer mit einer schönen Kamera-Animation: http://potree.org/potree/examples/camera_animation.html

- <http://potree.org/potree/examples/classifications.html>

So könnte man weitere Funktionen implementieren:

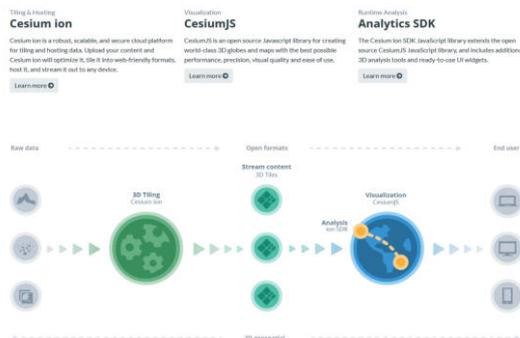
http://potree.org/potree/examples/custom_sidebar_section.html - Also eine eigene Sidebar in Potree einbauen.

Cesium

Cesium ist eine teilweise Open-Source Plattform für Pointclouds. Das Hosten und Tiling der Daten ist closed Source, aber die Visualisierung ist Open-Source. Die Punktwolken werden auf einem 3D Globus abgebildet.

"At the heart of our platform is 3D tiling, which transforms massive and diverse 3D geospatial data into streamable 3D content ready to be used in your application and in any environment."

- Chrome > Firefox (Firefox geht nicht)



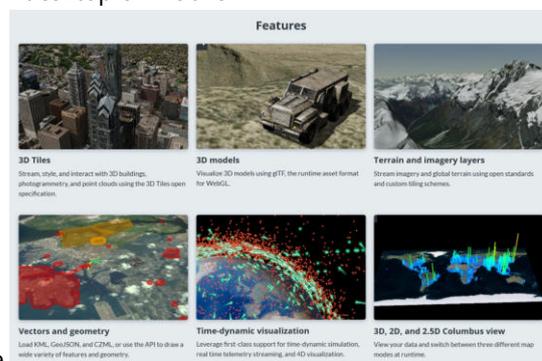
Cesium Ion

Ist nur die Tiling und Hosting Plattform und deshalb für uns nicht weiter relevant.

CesiumJS

Open Source

- 3D Globe
- Stream in 3D Tiles and other standard formats from Cesium ion or another source
- Visualize and analyze on a high-precision WGS84 globe
- Share with users on desktop or mobile



- Apache 2.0 license

Cesium Ion SDK

Bring analysis to your 3D visualization. The Cesium ion SDK JavaScript library extends the open source CesiumJS JavaScript library and includes additional GPU-accelerated 3D analysis tools and ready-to-use UI widgets.

Measure distance and areas, perform line of sight, viewshed, and visibility analysis, and include ready-to-use UI widgets into your custom apps.

Ist aber wieder closed Source und deshalb nicht relevant.

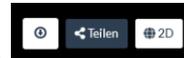
WebODM

WebODM ist die Webversion von Open Drone Map und ist Open Source (<https://github.com/OpenDroneMap/WebODM>).

WebODM verwendet zur Anzeige der 3D-Daten den Potree-Viewer.

- Falls wir den Potree-Viewer ebenfalls verwenden, könnten wir eventuell die zusätzlichen Knöpfe

(Herunterladen, teilen, 2D-Ansicht) von WebODM übernehmen.



WebODM macht und kann viel. Da unsere Applikation nicht für die Photogrammetrie ausgelegt sein muss, ist die ganze Prozessierung bei ODM "zu viel für uns". Dennoch können wir vielleicht aus dem WebODM-Code einiges lernen und uns abschauen.

iTowns

- <http://www.itowns-project.org/>
- iTowns is a web framework written in Javascript/WebGL for the visualisation of 3D geographic data and precise 3D measurements. iTowns is highly extensible and based on three.js and thus support all three.js features (post-processing, controls, 3D models and more).
- Viele Funktionen. Interessant hierbei ist die Möglichkeit, per Drag and Drop Laz-Files anzeigen zu lassen. Vielleicht wäre das für uns auch eine Möglichkeit:
https://github.com/iTowns/itowns/blob/master/examples/laz_dragndrop.html
- Entwine Point Tile (ept) formats

Plasio

<https://plas.io/>

- Kann LAS und LAZ anzeigen
- Chrome only
- Letzter Commit vor 3 Jahren

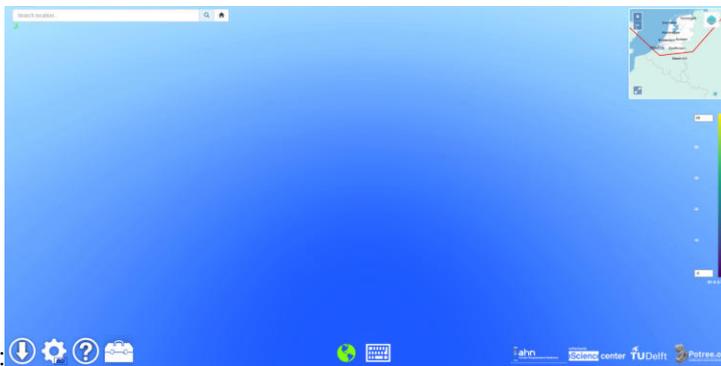
Plasion ist ein kleiner Viewer mit wenig Funktionen. Keine Speicherung von Punktwolken nötig. Vermutung: Probleme mit riesigen Files.

AHN pointcloud viewer

Für Punktwolken in der Grösse eines ganzen Landes.

- It uses a PostGIS database to approximate number of points in selected area.

- <https://github.com/NLeSC/ahn-pointcloud-viewer> -> Webversion: <https://github.com/NLeSC/ahn-pointcloud-viewer-ws>
- letztes Issue 2019 und letzter pullrequest 2017
- Letzte Commits vor mehreren Jahren. Bei der nicht-web Version schlägt das CI fehl und wurde seit 2018 nicht gefixt.
- <http://www.pointclouds.nl/docs/taming-beast-free.pdf>
- Braucht irgendwie das Potree-Format (Massive-PotreeConverter - ist eine Erweiterung/ Verschnellerung des PotreeConverters) auch
- Messungen können vorgenommen werden.
- <http://ahn2.pointclouds.nl/> lädt sehr lange bis unendlich lange (Unklar ob das am Tool oder dem Server liegt)
- Wenig Visualisierungsfreiheit
- Google Chrome oder Microsoft Edge



Lädt nicht:    alt="" width="200"/>

Open-Source Libraries

D3.js

- Viewer Variante.
- Standardmässig über SVG und mit 2D-Daten ca 1,000 datapoints, Canvas ca.10,000 datapoints.
- Erweiterung D3FC mit WebGL GPU Verwendung ~1M datapoints und Loading Large Datasets with Streams
- Ermöglicht Zoom, Rotating, Colouring, Annotationen etc.
- d3-3d.js: D3 mit 3D Punktwolke; d3-3d is meant for 3d visualizations. d3-3d allows the projection of 3d data onto the screen in the webbrowser. It is specially designed to work with d3.js.
- Leute die sich gegen den generellen Einsatz von D3 stellen, erklären, dass seit dessen Einführung viel gegangen ist und man viele Diagramme auch mit CSS und HTML einfach hinkriegt. Aber auch sie sehen, dass die Wartung mit D3.js einfacher ist, sofern alle im Team bereits Erfahrungen damit haben. Wenn aber nicht, ist es zu hinterfragen, ob der Einsatz und das Laden der knapp 70MB und die Einarbeitung des Teams wirklich nötig ist.
- <https://d3js.org/>
- <https://blog.scottlogic.com/2020/05/01/rendering-one-million-points-with-d3.html>
- <https://bl.ocks.org/Nieked/d8007a5f71f45ab80a2977a8eb7ab3c9>
- <https://github.com/nieked/d3-3d/>
- <https://bl.ocks.org/Nieked/raw/d8007a5f71f45ab80a2977a8eb7ab3c9/>
- <https://dzone.com/articles/render-geographic-information>
- <https://www.toptal.com/javascript/a-map-to-perfection-using-d3-js-to-make-beautiful-web-maps>

- https://www.d3-graph-gallery.com/intro_d3js.html

Threejs

- JavaScript 3D Library
- <https://threejs.org/editor/> man kann Formen im Editor importieren, Lichter platzieren und dies wieder exportieren. (Keine Punktwolken Importmöglichkeit.)
- <https://github.com/mrdoob/three.js/>
- Because glTF is focused on runtime asset delivery, it is compact to transmit and fast to load. Features include meshes, materials, textures, skins, skeletons, morph targets, animations, lights, and cameras. When glTF is not an option, popular formats such as FBX, OBJ, or COLLADA are also available and regularly maintained.
- Only a few loaders (e.g. ObjectLoader) are included by default with three.js — others should be added to your app individually. `import { GLTFLoader } from 'three/examples/jsm/loaders/GLTFLoader.js';`
- [PCDLoader](#): Pointcloudloader [Code](#)
- Punkte Beispiel: <https://threejs.org/docs/#api/en/materials/PointsMaterial>
- <https://dev.to/maniflames/pointcloud-effect-in-three-js-3eic>
- Implementation von Messen noch nicht gegeben.
- Laden von Octree Struktur selber organisieren.

Closed-Source Pointcloud Viewer

PointSharePlus

<https://pointshareplus.com/>

- Nicht Open-Source

PointSharePlus ist ein kostenpflichtiger Pointcloud Viewer, der unsere Requirements nahezu vollständig erfüllen würde. Der kostenpflichtige, closed-Source Teil ist aber nicht in unserem Sinne.

Wichtige Funktionen:

- Pointclouds hochladen
- Pointclouds bearbeiten (Ungewollte Punkte verstecken)
- Pointclouds mit allen oder nur ausgewählten Personen teilen
- Messungen vornehmen Und mehr...

Was allerdings nicht geht ist, die Punktwolke richtig eingefärbt und mit Einstellungen zu sharen. Man teilt die Punktwolke "roh", ohne die eigenen rein visuellen Veränderungen.

Useful Features



Viewing and Navigation

Variety of display and navigation tools to help better understanding the context of the pointcloud and assist novice users to find their way around the 3D data. Supported formats include Las, Laz, Pts, Ptx, Ply and Obj



Import 3D Model

Design Visualisation and Clash Detection. Import massive 3D design model into your point cloud for the ultimate visualisation tool. Ideal for Design Clash detection. Supported formats are OBJ, PLY, Revit (*.rvt), Navisworks (*.nwd), IFC and FBX



Measurement

Ability to measure Distances, Heights, Slopes, Coordinates, Angles and Areas. The ability to measure critical items leads to a reduction in site visits. Supporting Metric and Imperial



Sharing

Securely share the datasets with other people in and outside of your organisation. All relevant experts working on the same project will benefit by working in a single geo-spatial platform, with access to your team members' comments.



Profile View

Draw a path and view the profile of the path with measurement and tagging capabilities in profile view. Profile points can be exported to LAS/CSV files



Clipping and Layering

Isolate your view to specific objects or hide unwanted points. This function is handy for cleaning your point cloud and removing unnecessary information.



Drawing Tools

Draw Floor Plans and Section with various drawing tools such as Polyline, Arch, Circle and Rectangle. Finally export to AutoCAD DXF format.
 ✓ Floor Plan
 ✓ Sections



Annotation Tags

Highlight points of interest (POIs) and annotate them. Add notes and attach photos. Export and Share the POIs with other users.



Explore More...

Sign up for your **14 days trial** to explore more features and experience navigating through the 3D pointcloud world.

))

udStream

<https://udstream.euclidean.com/>

- Kostenpflichtig
- Funktionen wie Messen, Visualisierung ändern, Annotations setzen, Punktwolke per Link teilen, etc.
- Kann riesige Punktwolken anzeigen, welche beim Bewegen scharf bleiben (Keine Octree struktur sondern optimierter Speicherzugriff (Festplatte oder sogar USB Stick möglich))
- Chrome oder Edge (Chrome hat bei mir trotz neuester Version nicht funktioniert)
- Laptop während Benutzung voll ausgelastet (und auch wenn Browser nicht im Vordergrund ist, ist der Laptop langsam)

Resultate

Es werden nur die Open Soure Viewer verglichen:

Kriterium	Potree	Cesium	WebODM	iTowns	Plasio	AHN pointcloud viewer
Opensource	Ja	teils	Ja	Ja	Ja	Ja
Pointcloud ansehen	Ja	Ja	Ja*	Ja	Ja (max grösse vermutlich schnell erreicht)	Ja(Potreebased?)

Kriterium	Potree	Cesium	WebODM	iTowns	Plasio	AHN pointcloud viewer
Pointcloud Ansicht einstellen	Ja	Ja	Ja	Ja	Ja	Ja
Pointcloud download	Nein	Nein	Ja	Nein	Nein	Nein
Messen in Ansicht	Ja	Ja	Ja	Ja	Ja	Ja
Login	Nein	Nein	Ja	Nein	Nein	Nein
Breite Browser unterstützung	Ja	Ja	NEIN	Nein	Nein	Nein
Einfache Bedienbarkeit**	Ja	Ja	Nein	Ja	Ja	Ja
Aktuell unterhalten	Ja	Ja	Ja	Ja	Nein	Nein
Viel verwendet	++	++	+	-	--	--

*Potree basiert **Analysiert auf die Funktionalitäten unserer Requirements

Cesium (genauer CesiumJS) und Potree müssen nochmal genauer verglichen werden, da beide viel können und gut erhalten sind. Dazu werden einige Kriterien in folgender Tabelle bewertet: (Bewertung von 1-10, jeweils ungewichtet - gewichtet)

Kriterium	Potree	Cesium	Gewichtung
Gut dokumentiert	4 - 0.6	8 - 1.2	15%
User Interface Look (für Visualisierung)	9 - 0.9	5 - 0.5	10%
User Interface Look (für Messungen)	8 - 0.8	9 - 0.9	10%
Optionen für den User (out of the Box)	7 - 2.1	5 - 1.5	30%
Code Beispiele	8 - 1.6	7 - 1.4	20%
Steuerung (Ansicht verändern)	9 - 0.9	6 - 0.6	10%
Einfache Einbindung	10 - 0.5	7 - 0.35	5%
Total:	55 - 7.4	47 - 6.45	100%

Potree hat nicht nur ungewichtet, sondern auch mit Gewichtung besser abgeschnitten. Cesium hat zwar viele coole Features (<https://cesium.com/blog/categories/user-stories/>), jedoch für unsere Zwecke zu viel. Ausserdem sollte der User keinen Code schreiben müssen. Die Beispiele in Cesium sind zu spezifisch. Man könnte sie zwar vereinfachen und nur wenige Features dem User bereitstellen, dafür ist das UI bei Potree aber schöner. Man beachte zusätzlich, dass die Visualisierungen in Cesium auf einem Globus abgebildet werden.

Das ist bei uns nicht zwingend notwendig. Wird dies in Zukunft aber gewünscht, wäre dies in Potree auch möglich, da es CesiumJS verwenden kann.

Die beiden Vergleichstabellen zeigen, dass eine Implementation ähnlich zu WebODM Sinn machen würde. Da dies aber viele photogrammetrischen Funktionen mitbringt, ist WebODM für unseren Anwendungsfall nicht sehr geeignet. Deshalb wird ein Frontend entwickelt, welches wie WebODM den Pointcloud Viewer Potree integriert. Eine Alternative zu Potree wäre nach dem obigen Vergleich nur eine Selbstimplementation. Da Potree allerdings schon alle darstellungsrelevanten Funktionen abdeckt und auch gut funktioniert, erachten wir dies als nicht sinnvoll. Potree nutzt schon Cesiumjs, Threejs, D3.js und weitere Libraries, die auch in einer Selbstimplementation sehr wahrscheinlich zum Einsatz kämen. Ausserdem kann Potree einfach in eine Website eingebunden werden (siehe https://potree.github.io/potree/examples/embedded_iframe.html). Zusätzlich dazu wäre eine Selbstimplementation im Rahmen dieser Arbeit gar nicht möglich. Wir werden also Potree als Viewer in unsere Website einbinden.

1.2 Recherche Speicher

Research Storage

Varianten (Kandidaten)

PLY

- Unstrukturiert
- In der Industrie weniger häufig verwendet wird als LAS. (https://www.kaart.com/scanning_tips/point-cloud-formats/)
- Von der Stanford University erfunden für die Verwendung mit 3D-Scandaten. Wird auch als Stanford Triangle-Format bezeichnet.
- Ein 3D-Objekt wird als Sammlung von Polygonnetzen beschrieben.
- Es enthält einen Header in ASCII und eine Liste mit den Punkten, die die Polygone beschreiben.

```
ply
format ascii 1.0           { ascii/binary, format version number }
comment made by Greg Turk   { comments keyword specified, like all
lines }
comment this file is a cube
element vertex 8           { define "vertex" element, 8 of them in
file }
property float x           { vertex contains float "x" coordinate }
property float y           { y coordinate is also a vertex property }
property float z           { z coordinate, too }
element face 6             { there are 6 "face" elements in the file
}
property list uchar int vertex_indices { "vertex_indices" is a list
of ints }
end_header                 { delimits the end of the header }
0 0 0                      { start of vertex list }
0 0 1
0 1 1
0 1 0
1 0 0
1 0 1
1 1 1
1 1 0
4 0 1 2 3                  { start of face list }
4 7 6 5 4
4 0 4 5 1
4 1 5 6 2
4 2 6 7 3
4 3 7 4 0
```

-> Im Binär-Format werden die numerischen Werte einfach nur kompakter zusammengefasst. Durch den im Header spezifizierten Endian und gegebene Datentypen der property-Attribute, können die Werte so weiterhin einzeln interpretiert werden. Die property-list-Notation für Polygone sieht für beide Versionen vor,

dass die erste Zahl die Vertex-Anzahl des Polygons angibt und die darauf folgende Liste dann die Vertex-Indexe aufreihet.(Zitat)

(Quelle: https://de.wikipedia.org/wiki/Polygon_File_Format)

LAZ (LAS)

- Bei der Archivierung sollten die Daten möglichst gut komprimiert werden und man möchte einen Fallback-Plan haben, falls das Format nicht mehr unterstützt werden sollte. Bei LAS/LAZ/XYZ/e57 gibt es eine offene Dokumentation, wie man auf die Daten zugreifen kann. Daher kann man hier, falls das niemand mehr benutzen sollte, sich selber was „schreiben“ (programmieren).
- LAS/LAZ/XYZ kann nur unstrukturierte Punktwolken (unstructured point clouds) speichern. E57 und RCP können strukturierte und unstrukturierte Punktwolken (structured & unstructured) speichern. Die beste Komprimierung (wenigster Speicherplatz) für unstrukturierte Punktwolken wäre LAZ (zipped LAS). Für strukturierte gibt es eigentlich nur E57, welches ein "offenes" Format ist. (Quelle: <https://www.laserscanning-europe.com/de/laserscanning-forum/deutsches-laserscanning-forum/laserscanning-software/unterschied-der-einzelnen-punktwolkenformate>)
- Eine LAZ-Datei ist eine verlustfreie Komprimierung einer LAS-Datei. LAZ-komprimierte Dateien sind 5 bis 13-mal kleiner als die Originaldatei.
- LAS ist ein offenes Dateiformat, welches sich zum Austausch von 3-dimensionalen Punktwolken zwischen verschiedenen Benutzern eignet. Es kann zu jedem Punkt neben der Koordinate auch ALS-spezifische Attribute abspeichern. Es hat sich zum Standard-Austauschformat für ALS-Daten entwickelt. Das LAS-Dateiformat wird vom American Society for Photogrammetry and Remote Sensing (ASPRS) verwaltet und weiterentwickelt. Das Dateiformat ist binär, darum benötigt die Speicherung der Daten bedeutend weniger Platz im Vergleich zu ASCII-Daten. Zudem können solche binären Dateien von Programmen deutlich schneller gelesen und geschrieben werden. ASCII-Daten müssen beim Lesen und Schreiben immer in binär umgewandelt werden. (Zitat also für Doku noch selber schreiben: <http://unigis.sbg.ac.at/files/Masterthesen/Full/103226.pdf>)
- Das LAS(LASer)-Format ist ein Dateiformat, das für den Austausch und die Archivierung von Lidar-Punktwolkendaten entwickelt wurde.
- Ist ein offenes Binärformat, von der American Society for Photogrammetry and Remote Sensing (ASPRS)
- Das Format ist weit verbreitet und gilt als Industriestandard für Lidar-Daten. (https://www.kaarta.com/scanning_tips/point-cloud-formats/)

e57

- Ein strukturiertes Format, das das schnelle Laden von Subwolken ermöglicht und die Aufzeichnung der Position des Lidar-Sensors für jede Cloud-Aufzeichnung vorsieht.
- Dies ermöglicht die Ausrichtung der geschätzten Normalen für jede Subwolke, die für viele Cloud-Analyseschritte erforderlich sind.
- Die Struktur mit Laser-SLAM-basierten Moving Scannern zu verwenden ist kompliziert, da Daten kontinuierlich in einer zeitlichen Abfolge erfasst und nicht räumlich organisiert werden (Punkte können sehr weit voneinander entfernt sein, da es einen Aufnahme-shift gibt, was eine spezielle Anwendung für die Struktur braucht.) (Zitat: https://www.kaarta.com/scanning_tips/point-cloud-formats/)

Entwine (EPT)

- Entwine ist eine Datenorganisationslibrary, damit kleine bis zu terrabyte-grosse Punktdatensätze verlustfrei indiziert werden können (man kann die Punktwolke komplett wieder rekonstruieren, sogar mit Attributen, order, etc). Efficiently enables Querying, Analyzing, Visualizing, Enriching (Quelle: <https://media.ccc.de/v/bucharest-129-continental-scale-point-cloud-data-management-and-exploitation-with-entwine#t=254>), Slider mit Punktdichte gerade über Kacheln umsetzbar.
- Lizenz LGPL
- EPT ist ein Octree-basiertes Speicherformat. Es enthält JSON Metadaten sowie die Binärpunktdaten. Das zentrale ept.json enthält dabei die wichtigsten Informationen (bounds: kubische Grenzen der Octree-Indexierungsstruktur; boundsConforming: engste Grenzen, die dem maximalen Umfang der Daten entsprechen; dataType: laz, binary, schema; hierarchyType: Die Hierarchie selbst wird immer als JSON dargestellt, kann aber Komprimiert werden, -> json, gzip; points: total Punkte; shema: Repräsentation der Indexdimensionen; span: Grösse der Octree Rastergrösse in 2er Potenz; srs: Bezugssystem wenn vorhanden automatisch oder manuell abgefüllt; version: Version). Die Punktwolkendaten selber sind in 3D-Kacheln, wie im span definiert, abgelegt (Name bei nur 2D-Kachelung und Flug-LiDAR häufig: Zoom-x-y oder 3D-Kacheln Depth-x-y-z, Die Rootkachel ist immer 0-0-0-0 und repräsentiert das Volumen des kubischen Grenzwertes aus dem ept.json "Eine Grenze mit einem Volumen von 256 Kubikmetern und einem Spannweitenwert von 256 entspricht also einem Wurzelknoten mit einer Auflösung von bis zu 1 Punkt pro Kubikmeter."). Eine Kachel der Tiefe d kann bis zu 8 Unterknoten der Tiefe d + 1 besitzen. In Web-Mercator beispielsweise, wo X nach Osten und Y nach Norden zunimmt, repräsentiert eine Durchquerung von 0-0-0-0 zu 1-1-0-1 eine Durchquerung des Teilvolumens Ost / Süd / Abwärts. Es gibt keine maximale Auflösung, sondern e wird so häufig geteilt, bis es keine weiteren Punkte mehr gibt. Im Gegensatz zum Raster-Kachelungsverfahren werden die Kacheln hier nicht ausgetauscht, sondern mit den Punkten der nächsten Kachel ergänzt. Beim ept-hierarchy ist die namensgebung gleich der Kacheln. Es zeigt auf, was für Knoten existieren und wie viele Punkte darin enthalten sind. Es muss immer einem positiven Wert entsprechen, ausser es gibt ein eigenes File mit dem entsprechenden Subtree, so wird dieses mit -1 markiert. Source Informationen aus der Eingabe-Datei werden hier in einem JSON-Array abgelegt. Sind keine Daten verfügbar, ist die Datei leer.

Struktur

```

├── ept.json
├── ept-data
│   └── 0-0-0-0.laz
├── ept-hierarchy
│   └── 0-0-0-0.json
└── ept-sources
    ├── list.json
    └── 0.json

```

ept.json

```

{
  "bounds": [634962.0, 848881.0, -1818.0, 639620.0, 853539.0, 2840.0],
  "boundsConforming": [635577.0, 848882.0, 406.0, 639004.0, 853538.0, 616.0],
  "dataType": "laszip",
  "hierarchyType": "json",

```

```

    "points": 10653336,
    "schema": [
      { "name": "X", "type": "signed", "size": 4, "scale": 0.01, "offset":
637291.0 },
      { "name": "Y", "type": "signed", "size": 4, "scale": 0.01, "offset":
851210.0 },
      { "name": "Z", "type": "signed", "size": 4, "scale": 0.01, "offset": 511.0
}],
      { "name": "Intensity", "type": "unsigned", "size": 2 },
      { "name": "ReturnNumber", "type": "unsigned", "size": 1 },
      { "name": "NumberOfReturns", "type": "unsigned", "size": 1 },
      { "name": "ScanDirectionFlag", "type": "unsigned", "size": 1 },
      { "name": "EdgeOfFlightLine", "type": "unsigned", "size": 1 },
      { "name": "Classification", "type": "unsigned", "size": 1 },
      { "name": "ScanAngleRank", "type": "float", "size": 4 },
      { "name": "UserData", "type": "unsigned", "size": 1 },
      { "name": "PointSourceId", "type": "unsigned", "size": 2 },
      { "name": "GpsTime", "type": "float", "size": 8 },
      { "name": "Red", "type": "unsigned", "size": 2 },
      { "name": "Green", "type": "unsigned", "size": 2 },
      { "name": "Blue", "type": "unsigned", "size": 2 },
      { "name": "OriginId", "type": "unsigned", "size": 4 }
    ],
    "span" : 256,
    "srs": {
      "authority": "EPSG",
      "horizontal": "3857",
      "vertical": "5703",
      "wkt": "PROJCS[\"WGS 84 ... AUTHORITY[\"EPSG\", \"3857\"]]"
    },
    "version" : "1.0.0"
  }

```

ept-hierarchy Beachten Sie, dass dieses Beispiel aus Gründen der Übersichtlichkeit hierarchisch angeordnet ist, was in der Praxis aber nicht der Fall ist. In diesem Beispiel enthält der Stammknoten 0-0-0-0 65341 Punkte. Knoten mit Nullpunkten werden niemals in die Hierarchie aufgenommen, sodass das Fehlen eines untergeordneten Schlüssels auf einen Blattknoten hinweist (z.B. 4-4-6-12 oben).

```

{
  "0-0-0-0": 65341,
  "1-0-0-0": 438,
    "2-0-1-0": 322,
  "1-0-0-1": 56209,
    "2-0-1-2": 4332,
    "2-1-1-2": 20300,
    "2-1-1-3": 64020,
      "3-2-3-6": 32004,
        "4-4-6-12": 1500,
        "4-5-6-13": 2400,
      "3-3-3-7": 542,
  "1-0-1-0": 30390,

```

```

    "2-1-2-0": 2300,
    "1-1-1-1": 2303
  }

```

ept-sources

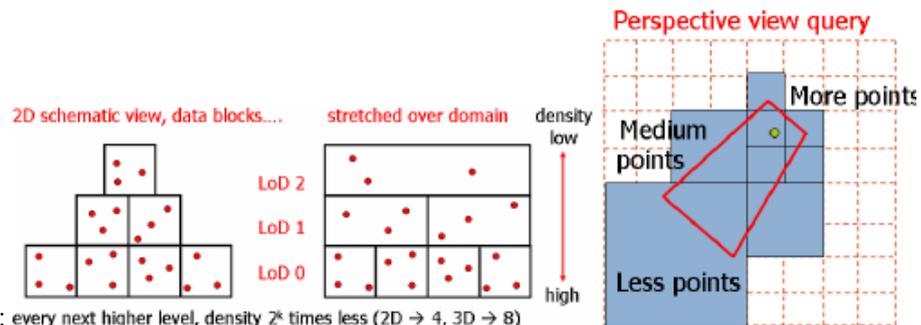
```

[
  {
    "id": "not-a-point-cloud.txt"
  },
  {
    "id": "autzen-low.laz",
    "bounds": [635577.0, 848882.0, 406.0, 639004.0, 853538.0, 511.0],
    "url": "autzen.json"
  },
  {
    "id": "autzen-high.laz",
    "bounds": [635577.0, 848882.0, 511.0, 639004.0, 853538.0, 616.0],
    "url": "autzen.json"
  }
]

```

Potree file format

- Octree-basiert
- Spacing ist die Mindestdistanz zwischen den Punkten im Root. In den children ist sie automatisch halb so gross.



- Gutes Bild: every next higher level, density 2^k times less (2D \rightarrow 4, 3D \rightarrow 8)

https://github.com/PropellerAero/potree-propeller-private/blob/master/docs/file_format.md

- Node Files can be BINARY, LAS oder LAZ
- Cloud.js enthält die Metadaten: version, octreeDir (Ordner oder Url wo die Daten sind), boundingBox, tightBoundingBox, pointAttributes, POSITION_CARTESIAN, COLOR_PACKED, spacing

```

{
  "version": "1.6",
  "octreeDir": "data",
  "boundingBox": {
    "lx": -4.9854,

```

```

        "ly": 1.0366,
        "lz": -3.4494,
        "ux": 0.7023000000000001,
        "uy": 6.7243,
        "uz": 2.2383
    },
    "tightBoundingBox": {
        "lx": -4.9854,
        "ly": 1.0375,
        "lz": -3.4494,
        "ux": -0.7889,
        "uy": 6.7243,
        "uz": 1.1245
    },
    "pointAttributes": [
        "POSITION_CARTESIAN",
        "COLOR_PACKED"
    ],
    "spacing": 0.03,
    "scale": 0.001,
    "hierarchyStepSize": 5
}

```

- Nummerierung erlaubt das Erkennen der Hierarchie (r is the root node, r0 is the first child of the root node, r03 is the fourth child of the first child of the root node) max 8 Kinder.
- Die Hierarchie eines Knotens wird in .hrc-Dateien in Paketen von 5 Bytes gespeichert. Diese Liste von 5-Byte-Paketen ist eine erste Durchquerung des Baums.
- Child nodes are arranged like this: Bild auch gut! (Quelle: https://github.com/PropellerAero/potree-propeller-private/blob/master/docs/file_format.md)

PostgreSQL mit pgPointCloud

- pgPointCloud ist eine Open-Source-PostgreSQL-Erweiterung zum Speichern und Verwenden von Punktwolken Daten mit PostGIS.
- Ermöglicht eine gute Integration mit anderen georäumlichen Daten (Vektor, Raster) in ein gemeinsames Framework: PostGIS
- Mit PDAL-Daten in DB importieren - pgPointCloud writer

```

{
  "pipeline": [
    {
      "type": "readers.las",
      "filename": "/home/lidar/st-helens-small.las",
      "spatialreference": "EPSG:26910"
    },
    {
      "type": "filters.chipper",
      "capacity": 400
    }
  ]
}

```

```

    "type":"writers.pgpointcloud",
    "connection":"host='localhost' dbname='pc' user='lidar' password='lidar'
port='5432'",
    "table":"sthsm",
    "compression":"dimensional",
    "srid":"26910"
  }
]
}

```

- Am besten wenn sich die Patches, also Punkte, die dicht beieinander liegen, nicht mit andern Patches überlappen.
- Ungeordnete Daten müssen zur Patchorganisation durch den "Chipper"-Filter, bevor sie in die DB können.
- Auch das Lesen geht über einen PDAL reader.
- Mit where kann man angeben, was gelesen werden soll.

```

{
  "pipeline":[
    {
      "type":"readers.pgpointcloud",
      "connection":"host='localhost' dbname='pc' user='lidar' password='lidar'
port='5432'",
      "table":"sthsm",
      "column":"pa",
      "spatialreference":"EPSG:26910",
      "where":"PC_Intersects(pa, ST_MakeEnvelope(560037.36, 5114846.45, 562667.31,
5118943.24, 26910))",
    },
    {
      "type":"writers.text",
      "filename":"/home/lidar/st-helens-small-out.txt"
    }
  ]
}

```

- Funktionen: PcPoint (PC_MakePoint, PC_AsText, PC_PCId, PC_Get, PC_MemSize); PcPatch (PC_Patch; PC_MakePatch; PC_NumPoints; PC_PCId; PC_AsText; PC_Summary; PC_Uncompress; PC_Union; PC_Intersects; PC_Explode; PC_PatchAvg; PC_PatchMax; PC_PatchMin; PC_PatchAvg; PC_PatchMax; PC_FilterGreaterThan; PC_FilterLessThan; PC_FilterBetween; PC_FilterEquals; PC_Compress; PC_PointN; PC_IsSorted; PC_Sort; PC_Range; PC_SetPCId; PC_Transform; PC_MemSize), WKB (PC_AsBinary; PC_EnvelopeAsBinary; PC_BoundingDiagonalAsBinary), PostGIS (PC_Intersects; PC_Intersection; Geometry; PC_EnvelopeGeometry; PC_BoundingDiagonalGeometry)
- Ermöglicht das Schreiben von eigenen Loading-Systemen:
<https://pgpointcloud.github.io/pointcloud/development.html> (Quelle:
<https://pgpointcloud.github.io/pointcloud/index.html>)
- Nicht verwenden von Postgis (One point per row means billions of rows) -> zu Patches organisieren und reduzieren auf millions.

- 2 Typen: **PcPatches** sind Sammlungen von PcPoints. **PcPoints** sind Packungen von Punkten (X, Y, Z, ...) in Byte-Arrays gepackt.
- Verwendung:

```
CREATE EXTENSION pointcloud;
CREATE EXTENSION postgis;          -- optional
CREATE EXTENSION pointcloud_postgis; -- optional
```

Nach dem Aktivieren von Pointcloud in einer Datenbank, werden die Ansichten pointcloud_columns (Informationen zu allen in der Datenbank vorhandenen PcPoint- und PcPatch-Spalten) und die Tabelle pointcloud_formats (XML-Dokumente, die definieren, wie Dimensionen in PcPoints codiert werden) zur Datenbank hinzugefügt.

```
SELECT pa FROM patches LIMIT1;-----
----- 0101000000020000000900000002100000000400000060CEFFFFBC9A78560000(1
row)
```

Die Query selektiert einen Patch und returnt einen WBK (Well Known Binary) String, der den Patch repräsentiert.

- Punktwolken in PostgreSQL können in QGIS als 2D-Grenzen (Umfänge der Patches werden angezeigt) zu Testzwecken und zum Debuggen betrachtet werden.
- Mit LOPoCS können die in PostgreSQL/ Pointcloud gespeicherten Punktwolkendaten mit verschiedenen Pointcloud-Webviewern, einschließlich Potree, Caesium und iTowns verwendet werden (Stream Protokolle die aktuell unterstützt werden: Greyhound (Greyhound is an HTTP server that provides dynamic level-of-detail point cloud streaming. Zitat: <https://greyhound.io/administration.html>) und 3D Tiles (3D Tiles is an open specification for sharing, visualizing, fusing, and interacting with massive heterogenous 3D geospatial content across desktop, web, and mobile applications. Zitat <https://github.com/CesiumGS/3d-tiles>)). LOPoCS sollen noch atraktiver werden. Sprache Python, Webframework Flask und basiert auf den Hauptbibliotheken 3D Tiles library (for 3D Tiles) and thelaz-perf library (for Greyhound) -> Weitere Infos zu den LOPoCS: <https://oslandia.github.io/lopocs/>, <https://oslandia.com/en/2016/11/03/locpocs-stream-point-cloud-postgis/>, <https://github.com/Oslandia/lopocs> -> letzter pullrequest 3.1.2020 last Issue 3.6.2020 (Quelle: <https://oslandia.com/wp-content/uploads/2018/05/Lemoine-Oslandia-Pointcloud.pdf>)

Oracle

- fme reader and writer (fme kommerzielles Verarbeitungs/ Umwandlungsprogramm)
- Funktionen: SDO_PC_PKG.CLIP_PC (Clip-Operation für eine Punktwolke), SDO_PC_PKG.CREATE_PC (Erstellt eine Punktwolke mit den Punkten der inptable), SDO_PC_PKG.DROP_DEPENDENCIES (Löscht die Abhängigkeiten zwischen einer Punktwolkenblocktabelle und einer angegebenen Basistabelle und -spalte), SDO_PC_PKG.INIT (Initialisiert eine Punktwolke), SDO_PC_PKG.TO_GEOMETRY (Gibt ein Geometrieobjekt zurück, das eine Punktwolke ganz oder teilweise darstellt) (Quelle: https://docs.oracle.com/cd/B28359_01/appdev.111/b28400/sdo_pc_pkg_ref.htm#SPATL1190)
- Kaum zu aktuellen Infos gekommen

- Kommerzielles Tool

Vergleichstabelle

Kriterium	Entwine (EPT)	Potree	LAZ	PLY	e57	PostgreSQL mit pgPointcloud
Pdal lesbar	Ja	Nein	Ja	Ja	Ja	Ja
Pdal ausgabe	Ja	Nein	Ja	Ja	Ja	Ja
Kachelung	Ja	Ja	Nein	Nein	Nein	Ja
Zugriff optimiert (insbesondere für die Darstellung)	Ja	Ja	Nein	Nein	Nein	Ja über thirdparty LOPoCS
Verbreitet	+	+	++	+	+	-
Aktuell	Ja	Ja	Ja	Ja	Ja	Ja
Ablage in Filestruktur oder DB	Filestruktur	Filestruktur	Filestruktur	Filestrucktur	Filestrucktur	DB Open-Source
Speichergrosse (file: 26935_12355)	27.4 MB	165 MB	23.8 MB	394.8 MB	87.8 MB	6.25 MB*
Exportierbar Ipad	Nein	Nein	3D Scanner App	SiteScape, 3d Scanner App	SiteScape, 3d Scanner App	PDAL lesbare Pointclouds importierbar (Vorteil wenn sie ordered sind)
Exportierbar Geoslam	eher nicht	eher nicht	denke ja	denke ja	denke ja	PDAL lesbare Pointclouds importierbar (Vorteil wenn sie ordered sind)
Exportierbar Leica	eher nicht	eher nicht	denke ja	denke ja	denke ja	PDAL lesbare Pointclouds importierbar (Vorteil wenn sie ordered sind)

- Nicht extra DB aufgesetzt und getestet, sondern berechnet: 384.65 MB LAZ braucht in PostgreSQL mit Batch 101.77 MB (Quelle: Massive point cloud data management: Design, implementation and execution of a point cloud benchmark). Daraus mit 3 Satz abgeleitet: $101.77 \text{ MB} / 384.65 \text{ MB} * 101.77 \text{ MB} = 6.25 \text{ MB}^*$

Vergleichbare Arbeiten

In einer Studie von Peter van Oosterom, Oscar Martinez-Rubib, Milena Ivanovab, Mike Horhammerc, Daniel Geringerc, Siva Ravadac, Theo Tijssena, Martin Kodded, Romulo Goncalves mit dem Titel "Massive point cloud data management: design, implementation and execution of a point cloud benchmark" wurden auch verschiedene Verwaltungsmethoden verglichen. Das Endresumee der Arbeit zeigt, dass es keine Lösung gibt, die für alle Probleme optimal ist. Sie kommen zum Schluss, dass alle ihre getesteten Varianten (PostgreSQL flat table und blocks, Oracle flat table, blocks und Exadata, MonetDB, LAStools) Vor- und Nachteile haben und bei jedem Anwendungsfall individuell entschieden werden muss. Diese Arbeit beinhaltet Vergleiche im Initialisieren, Abfragen und Speichern der Daten.

- https://www.researchgate.net/publication/272373258_Massive_point_cloud_data_management_Design_implementation_and_execution_of_a_point_cloud_benchmark

Resultate

Auf Basis der Daten, die wir im Prototyp verwalten möchten, werden wir eine filebasierte Struktur verwenden. Da unsere Daten im Prototyp noch nicht gemerged werden, sondern als einzelne Dateien gehalten werden, ergibt das Einsetzen einer DB kaum einen Mehrwert. Wir haben auch nicht Teilgebietsabfragen, wie dies im "Massive point cloud data management: design, implementation and execution of a point cloud benchmark" analysiert wurde. Zur Darstellung werden wir zwar eine Kachelung benötigen, aber jeweils die vom Kunden angewählte Pointcloud als Ganzes darstellen.

Welches der oben genannten Formate am besten geeignet wäre, wird am besten im Zusammenhang mit dem Viewer entschieden. Vielleicht ist auch eine Mischvariante sinnvoll. Die Daten werden im kleineren LAZ gespeichert und zur Ansicht in Entwine (oder ähnlichem) vorbereitet und dann optimal dargestellt. Das gibt natürlich Performanceeinbußen.

1.3 Recherche Prozessierung

Research processing

Varianten (Kandidaten)

PDAL

- PDAL bietet eine Reihe von Kommandozeilenanwendungen, mit denen Benutzer bequem Cloud-Daten verarbeiten, filtern, übersetzen und abfragen können.
- Mit Python gut kombinierbar (<https://pdal.io/python.html#python>) und sogar in Python filters integriert. Zudem gibt es eine Erweiterung, um PDAL in Programmen einzusetzen. (<https://pdal.io/>)
- Java (<https://pdal.io/java.html>)
- Viele Dateiformate konvertierbar, auch von Pointcloud in Raster
- Performance nicht optimal wegen der Unterstützung so vieler Formate. Spezifische Tools werden das sicher schneller hinkriegen.
- Kein benutzerfreundliches GUI vorhanden
- Core C++ Software Library
- Can compile and run on Linux, OS X, and Windows
- Tutorials <https://pdal.io/development/writing-reader.html#writing-reader> (<https://pdal.io/about.html#what-tasks-are-pdal-good-at>)
- Applications: delta (Nächsten Punkt suchen), density, ground (Erkenne Boden und nicht Boden), hausdorff (Nächste Distanz zwischen zwei Punktwolken), info, merge, pipeline, random (Generiert eine Punktwolke), sort, split, tile, tindex, translate (<https://pdal.io/apps/index.html>)
- PDAL license (BSD)
- JSON Pipelines: reader writer und filter (Create, Order, Move, Cull, New, Join, Metadata, Mesh, Languages, Other)
- Verbreitet, aktuell und gut unterhalten (Issue von vor 4 Tagen bereits beantwortet, seit januar 7 Pullrequests)

LAStool

- ~50MB
- LAStools are the fastest and most memory efficient solution for batch-scripted multi-core LiDAR processing and can turn billions of LiDAR points into useful products at blazing speeds and with low memory requirements. (<https://rapidlasso.com/lastools/>)
- Einige Funktionen: Klassifizieren, Tile, konvertieren, filtern, Raster, triangulieren, Konturen, Clip oder Polygon Extraktion von LiDAR Daten
- Natives GUI
- Toolboxes in QGIS, ArcGIS und ERDAS IMAGES
- Schnelles LIDAR-Daten processing (Quelle: <https://rapidlasso.com/lastools/>)
- Halb kommerzielles Tool (<https://rapidlasso.com/pricing/>). Ein Teil ist Open Source (LGPL 2.1). Der andere Teil ist eine Closed Source und erfordert eine Lizenz für die meisten kommerziellen oder staatlichen Zwecke. (<https://lastools.github.io/LICENSE.txt>)
- Open Source LGPL"free"
 - laszip
 - lasindex

- lasvalidate
- lasliberate
- lasinfo
- las2las
- lasdiff
- lasmerge
- las2txt & txt2las
- lasprecision
- LASzip (with static linking exception)
- LASlib (with LASzip) - the API used by LAStools.
- "free" aber jetzt (noch) nicht Open Source.
 - demzip
 - demdiff
 - e572las
 - lasview
 - laspublish
 - lasoptimize (<https://lastools.github.io/LICENSE.txt>)
- Weitere nicht kostenlose Tools wurden nicht aufgeführt.

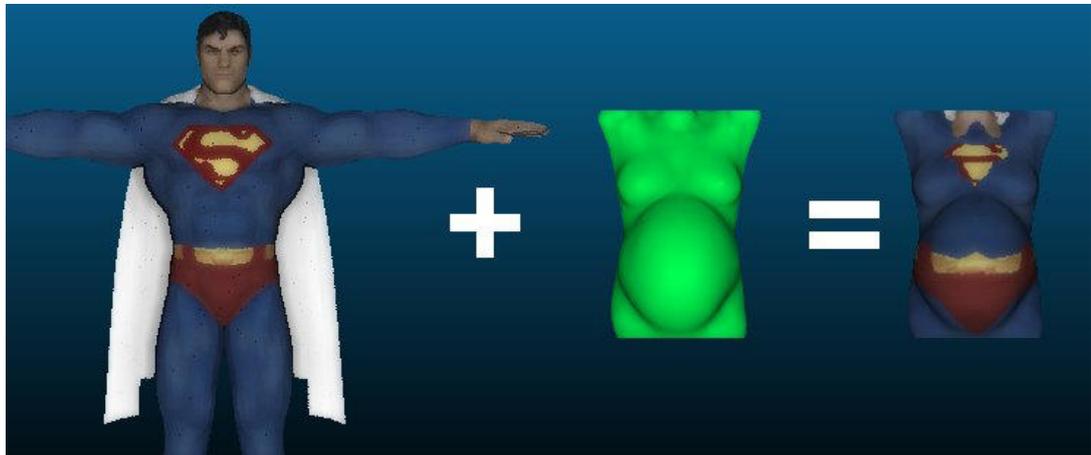
libLAS

- Toolset für die Datenübersetzung
- C/C++ library
- Lesen und Schreiben von LAS-Format
- Ab 2018 wurde libLAS durch das PDAL-Projekt ersetzt und befindet sich im Ruhezustand oder im Wartungsmodus (Quelle: <https://liblas.org/>)
- libLAS has C, C++, .NET, Ruby, and Python APIs.
- Crossplatform, Commandline (<https://liblas.org/features.html>)
- Reprojection

CloudCompare

- Verarbeitungssoftware für 3D-Punktwolken (und Dreiecksnetze/ triangular mesh).
- Entwickelt um einen Vergleich zwischen zwei dichten 3D-Punktwolken (wie sie mit einem Laserscanner erfasst wurden) oder zwischen einer Punktwolke und einem Dreiecksnetz durchzuführen.
- User interface Qt and OpenGL. (<https://www.danielgm.net/cc/>)
- Funktionen:
 - Global Shift and Scale: Zu grosse Koordinaten müssen geshifted werden, da intern mit einem 32 float gearbeitet wird (das spart Platz und ist performanter). Werden aber grosse Koordinaten nicht geshifted, verliert man Genauigkeit. Standardeinstellung kann aber übersteuert werden. https://www.cloudcompare.org/doc/wiki/index.php?title=Global_Shift_and_Scale
 - Clone, merge, rotate und translate (<https://www.cloudcompare.org/doc/wiki/index.php?title=Methods>)
 - Subsample: Dabei werden die Punkte zufällig, Octree-basierend oder spatial ausgedünnt. Die Originaldaten bleiben bestehen. (<https://www.cloudcompare.org/doc/wiki/index.php?title=Edit%5CSubsample>)

- Multiply/Scale: Skaliert die Entität relativ zum Ursprung des Koordinatensystems (jede Koordinate wird direkt mit dem/ den Eingabefaktor(en) multipliziert) - oder skaliert die Entität relativ zu ihrem bounding-box center (<https://www.cloudcompare.org/doc/wiki/index.php?title=Multiply/Scale>)
- Segment: (https://www.cloudcompare.org/doc/wiki/index.php?title=Interactive_Segmentation_Tool)
- Crop: Kann eine oder mehrere Wolken innerhalb einer 3D-Box zuschneiden. Kann auch zum Zuschneiden von Meshes verwendet werden. (<https://www.cloudcompare.org/doc/wiki/index.php?title=Crop>)
- Farben: (Eine einzige Farbe wählen, ausgewählte Objekte kolorieren, indem die aktuelle Farbe mit der angegebenen multipliziert wird, ähnlich wie bei der Level-Methode von Photoshop. Dann anwenden einer Farbrampe (linearer Farbverlauf oder Zyklus) auf die ausgewählten Objekte. Warning: any previous color field (if any) will be overwritten, Konvertiert das aktuelle RGB-Farbfeld in Graustufen (Warnung: Keine Bestätigung erforderlich), Konvertiert das aktuelle RGB-Farbfeld in ein oder mehrere Skalarfelder, Interpolieren Sie von einer anderen Entität



- , Verbessert die RGB-Farbdynamik mit dem zugehörigen Skalarfeld "Intensitäten" (RGB-IHS-Methode), Farbwerte entfernen)
- Normalen berechnen: Um Normalen auf einer Punktwolke zu berechnen, muss die lokale Oberfläche, die durch einen Punkt und seine Nachbarn dargestellt wird, geschätzt werden. Der level of noise und die Anzahl/ Entfernung der Nachbarn ändern das Aussehen dieser Oberfläche:
 1. surface model Wählen: (best fit) plane - robust to noise but very bad with sharp edges and corners 2D triangulation - weak to noise but good with sharp edges quadric (formerly called 'height function') - very good for curvy surfaces
 2. Nachbardistanz wählen - je grösser die Zahl, desto mehr Punkte werden für die Berechnung eingesetzt und umso besser das Modell (Aber auch langsamer).
 3. Die Orientierung der Normalen muss nun noch bestimmt werden. Man kann es weglassen, eine minimum Spanning Tree Methode oder das schnelle robuste Grid-Verfahren verwenden.
 - Normale invertieren, orientieren (Schritt 3), zu HSV Farbe konvertieren oder in ein field dip und dip direction exportieren.
 - Octree berechnen/ resampeln
 - Create Point Cloud
 - Viele Mesh Funktionen
 - Erstellen und Editieren von Planes und noch vieles mehr (<https://www.cloudcompare.org/doc/wiki/index.php?title=Methods>)

- Vergleichen 2er Punktwolken für die Auswertung der Scanner interessant
(https://www.cloudcompare.org/doc/wiki/index.php?title=How_to_compare_two_3D_entities)
- LGPL Lizenz (<https://www.cloudcompare.org/doc/wiki/index.php?title=License>)

Fusion

- Konvertierungs-, Analyse- und Anzeigesoftware
- Dreidimensionale Gelände- und Überdachungsflächenmodelle, sowie LIDAR-Daten die mit herkömmlichen zweidimensionalen Bildern (z.B. Orthofotografien, topografischen Karten, Satellitenbildern, GIS-Shapefiles) verschmolzen werden.
- Überdachungs- und Bodenoberflächenmodelle können hergestellt werden.
- Algorithmen, mit denen der Benutzer einzelne Baumattribute manuell messen oder mithilfe automatisierter Funktionen einzelne Baumattribute über große Bereiche hinweg charakterisieren kann. (<https://www.idaholidar.org/tools/fusion-ldv/>)
- Neueste Version January 21, 2021
- FUSION besteht aus einem grafischen Anzeigefenster und einem Kontrollfenster. Das FUSION-Display zeigt alle Projektdaten mithilfe eines für geografische Informationssysteme typischen 2D-Displays an.
- LDV bietet die 3D-Visualisierungsumgebung für die Untersuchung und Messung räumlich expliziter Datenuntergruppen.
- Befehlszeilenprogramme bieten spezifische Analyse- und Datenverarbeitungsfunktionen, mit denen FUSION für die Verarbeitung großer LIDAR-Erfassungen geeignet ist. (http://forsys.sefs.uw.edu/fusion/fusion_overview.html)
- FUSION wird im US-Landwirtschaftsministerium, Forstdienst, Pacific Northwest Research Station von einem Mitarbeiter der Bundesregierung im Rahmen seiner offiziellen Aufgaben entwickelt. Gemäß Titel 17, Abschnitt 105 des US-amerikanischen Kodex unterliegt diese Software keinem Urheberrechtsschutz und ist gemeinfrei. FUSION ist in erster Linie ein Forschungsinstrument. USDA Forest Service übernimmt keinerlei Verantwortung für die Verwendung durch andere Parteien und übernimmt keine ausdrücklichen oder stillschweigenden Garantien für die Qualität, Zuverlässigkeit oder andere Merkmale. (<http://forsys.sefs.uw.edu/fusion/fusionlatest.html>)

OrfeoToolbox

- Ist anscheinend keine Blackbox
- Open-Source-Projekt für hochmoderne Fernerkundung
- Bash, Python oder QGIS aufrufbare Apps
- Leistungsstarke C++-API
- Last release 2020-10-05
- Es kann hochauflösende optische, multispektrale und Radarbilder im Terabyte-Bereich verarbeiten
- Zugriff über MonteVerdi, QGIS, Python, die Befehlszeile oder C++
- Verfügbar unter Linux, MacOS und Windows
- Nicht weiter ermittelt, sehr viel in Richtung Bilddaten-Funktionen die wir nicht brauchen.

Entwine

- Siehe Storage
- Das Erstellen von Punktwolkendatensätzen für 3D-Kacheln zur Anzeige in Cesium erfolgt in zwei Schritten.

- merge, convert, scan (Aggregieren von Informationen in unindexierten Punktwolken, um ein Entwine Dataset zu erzeugen) (<https://entwine.io/configuration.html>)

Potree converter

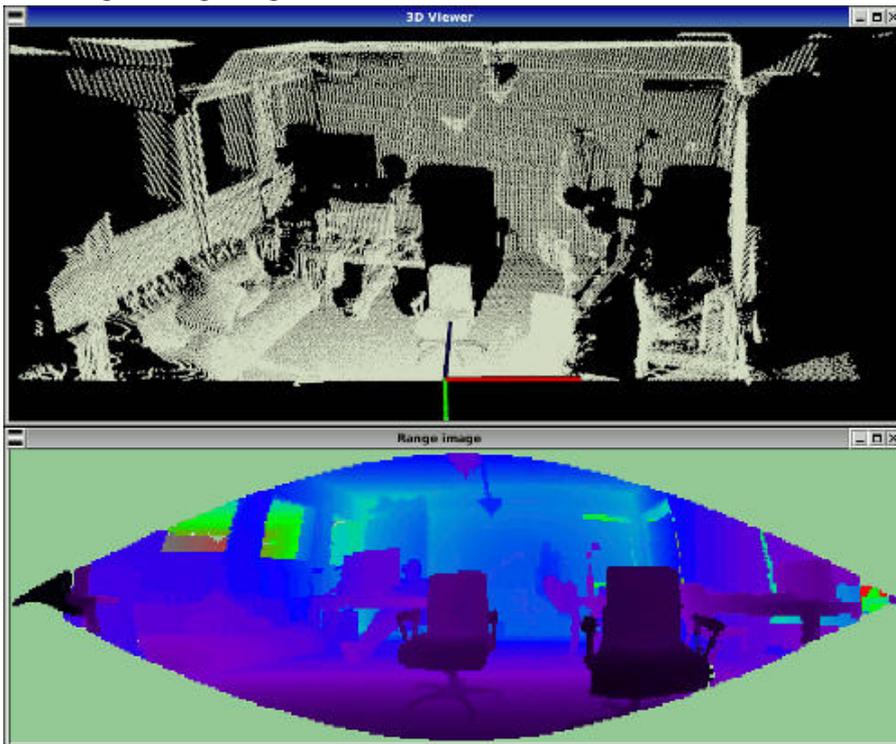
- Funktioniert sicher für LAZ nach PotreeOctree
- Sehr grosses Format
- Zurzeit noch keine Komprimierung
- Keine weitere Funktionalität

Point Cloud Library (PCL)

- Die Point Cloud Library (PCL) ist ein eigenständiges, großes, offenes Projekt für die 2D/ 3D-Bild- und Punktwolkenverarbeitung.
- Reihe modularer Bibliotheken. Der wichtigste Satz freigegebener Module: filters, features, keypoints, registration, kdtree, octree, segmentation, sample_consensus, surface, recognition, io, visualization; Alle: NPPST Core Basic functions for CUDA streams management, NPPST Image Processing, NPPST Signal Processing, Module common, Module features, Module filters, Module geometry, Module io, Module kdtree, Module keypoints, Module octree, Module outofcore, Module recognition, Module registration, Module sample_consensus, Module search, Module segmentation, Module surface, Module visualization
- BSD license
- Plattform spezifische Anwendung (auch spezifische Tutorials): Linux, Mac OS X, Microsoft Windows
- Verbreitet und aktuell und gut unterhalten (Issue von vor 6 Tagen bereits beantwortet, viele offene Pullrequests, einer vor 4 Tagen) (<https://github.com/PointCloudLibrary/pcl> 23.03.2021)
- Algorithmen wie Filterung, Merkmalschätzung, Oberflächenrekonstruktion, Registrierung, Modellanpassung und Segmentierung. Diese Algorithmen können beispielsweise verwendet werden, um Ausreißer aus verrauschten Daten zu filtern, 3D-Punktwolken zusammenzufügen, relevante Teile einer Szene zu segmentieren, Schlüsselpunkte zu extrahieren und Deskriptoren zu berechnen, um Objekte in der Welt anhand ihres geometrischen Erscheinungsbilds zu erkennen und Oberflächen daraus zu erstellen.
- Die Bibliothek pcl_octree bietet effiziente Methoden zum Erstellen einer hierarchischen Baumdatenstruktur aus Punktwolkendaten. Dies ermöglicht räumliche Partitionierungs-, Downsampling- und Suchoperationen für den Punktdatensatz. Jeder Octree-Knoten hat entweder acht Kinder oder keine Kinder. Der Wurzelknoten beschreibt einen kubischen Begrenzungsrahmen, der alle Punkte einkapselt. Auf jeder Baumebene wird dieser Raum durch den Faktor 2 unterteilt, was zu einer erhöhten Voxelauflösung führt.
- Umfangreiche Doku aber nicht alles aktuell (Links führen ins Bichts, oder Aussagen "das muss noch ergänzt werden".) (Quelle: <https://pointclouds.org/documentation/index.html>)
- Die zu Grunde liegenden Datenstrukturen in PCL nutzen SSE-Optimierungen in hohem Maße. Die meisten mathematischen Operationen werden mit einer Open-Source-Vorlagenbibliothek für Linearalgebra implementiert und basieren darauf.
- Wir haben uns entschlossen, jeden Algorithmus aus PCL als eigenständigen Baustein verfügbar zu machen, der leicht mit anderen Blöcken verbunden werden kann.
- Sie garantieren, dass für kritische Anwendungen, die es sich leisten können im selben Prozess ausgeführt zu werden, nicht notwendige Daten kopiert oder serialisiert/ deserialisiert werden müssen. Zu diesem Zweck haben wir Knoten erstellt, bei denen es sich um dynamisch ladbare Plugins handelt,

die wie ROS-Knoten aussehen und funktionieren, jedoch in einem einzigen Prozess (als einzelne oder mehrere Threads).

- PCL verfügt über eine eigene Visualisierungsbibliothek, die auf VTK basiert. Ziel ist es, die Ergebnisse von Algorithmen, die mit solchen überdimensionalen Daten arbeiten, schnell prototypisieren und visualisieren zu können.
- Ein Histogramm-Visualisierungsmodul (PCLHistogramVisualizer) für 2D-plots; Methoden zum Rendern und Festlegen visueller Eigenschaften (Farben, Punktgrößen, Deckkraft usw.) für jeden n-D-Punktwolkendatensatz; Verfahren zum Zeichnen grundlegender 3D-Formen auf dem Bildschirm (z. B. Zylinder, Kugeln, Linien, Polygone usw.) entweder aus Punktmengen oder aus parametrischen Gleichungen; RangeImage visualization modules



- Sehr stark auf die Robotik ausgelegt, deshalb ist das Erkennen von Gegenständen im Innenbereich im Vordergrund. Es muss überprüft werden, wie es mit der Außenbereichserkennung so klappt.
- Planen die Unterstützung von GPUs mithilfe von CUDA und OpenCL. (https://pointclouds.org/assets/pdf/pcl_icra2011.pdf)
- pcl installation nur auf Linux zum laufen gebracht
- pclpy 0.11.0 Linux Not working for now. Contributions are welcome!(<https://pypi.org/project/pclpy/>)

Vergleichstabelle

Kriterium	Entwine	Potree Converter	CloudCompare	LAStool	Pdal	PCL
Daten lesen LAZ	Ja	Ja	Ja	Ja	Ja	Ja
Daten Octree lesen	Nein	Nein	Ja	Ja (nicht EPTLaz)	Ja	Nein
Datenumwandlung in Octree	Ja	Ja	Ja	Nein	Ja(nicht EPT LAZ)	

Kriterium	Entwine	Potree Converter	CloudCompare	LAStool	Pdal	PCL
Punkte Erkennen (Boden, Objekte, Bruchkanten)	Nein	Nein	Nein	Ja	Ja	Ja
2D Raster generierbar	Nein	Nein	Nein	Ja	Ja	Ja
Daten rauschen eliminieren	Nein	Nein	Ja	Ja	Ja	Ja
Objekte entfernen /Pnktwolke bereinigen	Nein	Nein	Ja (Grafisch bearbeiten)	Ja (mit passender Clip Form)	Ja (mit passendem Clip Form)	Ja (mit passendem Clip Form)
Verbreitung	++	+	++	++	++	+
Unterhalten	Aktuell	Aktuell	Aktuell	Aktuell	Aktuell	

Zwischen Resultate

PDAL ist zurzeit leichter Führer vor PCL. PCL ist zwar auch sehr gut, aber wir müssten noch testen, ob dieses Octreeformat Potree lesbar ist (wahrscheinlich nicht). Da wir wissen, dass ept mit Pdal geschrieben und gelesen werden kann und dies Potree lesbar ist, ist Pdal vorne. Interessant zu wissen wäre, welches der beiden Tools die Objektklassierung und 2D ableitbare Extraktion besser machen kann. Das muss für die finale Entscheidung also noch angeschaut werden (neben der Potree Kompatibilität von PCL). Für die Bereinigung mit Objektentfernung wäre ein Verweis auf Cloudcompare sinnvoll. Dies muss somit vom Benutzer installiert sein. Eine Eigenentwicklung einer solchen Funktion würde den Rahmen dieser Arbeit sprengen.

Es stellt sich noch die Frage, ob wir noch etwas Platz sparen könnten, indem wir die LAZ und ept Ablage optimieren. Vielleicht wäre es möglich, dass das nur das Root-Element im entwine Format ist und der Rest erst mit später umgewandelt und nachgeliefert wird (unklar ob das Speicher sparen würde). Oder nur aktuelle/ viel verwendete Punktwolken im ept vorbereiten und die anderen im LAZ format archivieren. Vielleicht sogar Unterscheidung im Frontend anzeigen "schon lange nicht mehr angesehen" und "das Laden der Daten wird länger dauern", da die Aufbereitung in ept noch nicht abgeschlossen ist. Oder kann der Benutzer die Archivierung tätigen und dadurch bestätigen, dass das Laden der Daten nun länger dauert, da er sie nicht mehr regelmässig braucht?

Funktionstest PDAL vs PCL

Funktion	Gewichtung	PDAL	PCL
Plattform unabhängig einfach instalierbar	10%	10 - 1	3 - 0.3
Einfach Anwendbar	15%	9 - 1.35	4 - 0.6
Boden Erkennung für 2D Ableitung	20%	2 - 0.4	2 - 0.4
Objekt Erkennung für 2D Ableitung	20%	2 - 0.4	4 - 0.8

Funktion	Gewichtung	PDAL	PCL
DOM export für 2D Ableitung	15%	10 - 1.5	0 - 0
In Python integrierbar	20%	10 - 2	5 - 1

2 Scanner Vergleich

Vergleich der Geräte

Leica BLK2GO:

Positiv:

- Reichweite
- Leicht
- Kompakt
- App um Wolke vor Download zu betrachten
- Kann kopfüber messen

Neutral:

- Bedienung mit nur einem Knopf -> Einfach aber verwirrend

Negativ:

- Gerät hat nur mit iPhone App verbunden (Android verfügbar aber ging nach 10 Minuten probieren nicht)
- Beim Download kam immer die Meldung "erfolgreich" auch wenn es zum Abbruch gekommen ist
- Cyclone rechnet sehr sehr lange mit Ladebalken, der gerne bei 4% stehen bleibt
- Braucht mindestens so viel Platz auf dem C-Laufwerk, wie die zu verarbeitende Datei gross ist, obwohl in den Einstellungen das Schreiben auf Laufwerk D gesetzt wurde
- Leica Programme sind nicht intuitiv (Beispiel: Kleiner Akzeptieren-Knopf, gefolgt von einem grossen Akzeptieren-Knopf. Beispiel 2: Das Laden der Daten ist langsam, danach sind diese links im Programm gelistet. Der Import-Knopf ist aber rechts (und dauert ebenfalls sehr lange). Auch der direkt-Export startet man über den Import-Knopf rechts.)

GeoSLAM Revo RT:

Positiv:

- Reichweite
- Tablet für Live-Beobachtung

Negativ:

- Eigene Person sichtbar (Start ausserhalb des Messbereiches ist somit besser)
- Start und Ende etwas aufwändig (Start- und Stoppreihenfolge der Geräte muss beachtet werden)
- Gewicht (Macht es in unwegsamem Gelände schwieriger)
- Farbinfo nur mit Lizenz und aufwändig (Bei unserer grössten Datei hat es nicht geklappt)

iPad Pro:

Positiv:

- Leicht
- Günstig
- Schnelle Prozessierung
- Einfacher Export

Negativ:

- Genauigkeit (App abhängig)
- Messdauer
- Reichweite

Vergleich der Datenauswertung

Vergleichspunkt	Leica BLK2GO	GeoSLAM Revo RT	SiteScape (iPad Pro)	Scaniverse (iPad Pro)	3d Scanner App (iPad Pro)	Polycam (iPad Pro)
-----------------	-----------------	-----------------	----------------------------	--------------------------	------------------------------	-----------------------

Vergleichspunkt	Leica BLK2GO	GeoSLAM Revo RT	SiteScape (iPad Pro)	Scaniverse (iPad Pro)	3d Scanner App (iPad Pro)	Polycam (iPad Pro)
Farbige Punktwolke	Ja	Schwierig, GoPro hat einen kleineren Sichtwinkel als die Aufnahme mit dem Gerät. Das Einfärben klappt nur mit lizenzierter Software. ² Das dauert mehrere Minuten, da zuerst neu prozessiert und anschliessend eingefärbt werden muss. (Die grösste Datei, die der Villa Aurum, konnte nicht reprozessiert werden, ohne dass es Probleme gab Bild ¹	Ja	Ja, die Prozessierung kann in Standard, High oder Ultra erfolgen und je nach Scangrösse mehrere Minuten dauern	Ja, die Prozessierung kann in HD, Medium, Fast oder Custom erfolgen und je nach Scangrösse mehrere Minuten dauern	Ja, die Prozessierung kann in fast, space, object oder custom erfolgen und je nach Scangrösse mehrere Minuten dauern
Download ¹ Zeit pro Punktwolke	30min	10min	3min	5min	3min	3min
Download ¹ Software	BLK Download Manager, Cyclon	Website des Feld Computer, GeoSLAM Hub (Weitere zur Auswertung möglich)	iTunes	iTunes	iTunes	iTunes
Exportierbare Formate	b2g (Software Export in e57)	geoslam/ply (Software Export in LAZ, LAS, E57, PLY, TXT)	ply, e57	Send in Message, Upload in Sketchfab oder Export in FBX, GTF, USDZ, OBJ, STL, PLY	Punktwolken Export: XYZ Color, PLY, PTS, LAS, LAS Geo-Referenced, e57, PCD, XYZ no Color, XYZ color, space delimitated oder Modell Export: OBJ, GLTF, GLB, STL, DAE oder Upload in Sketchfab und send AR iMessage	Teilen: Polycam Weblink erstellen, Order Video, iMessage und Sketchfab. Exportieren: OBJ, GBL, DAE, USDZ, STL, DAE, DXF, PLY, XYZ, PTS

¹Download -> Punktwolke lesbar auf unserem PC.

²Danke an GeoSLAM für die Unterstützung und die Lizenz. Achtung beim Export aus GeoSLAM Hub gehen nicht eingefärbte Punkte mit Point Color ZebCam verloren. Nicht aber beim Export Point Color ZebCam + Shaded!

Wir schätzen die Reichweite der Scanner, bzw deren Sensor auf 15m beim GeoSLAM und BLK2GO und auf 4m beim iPad Pro.

Download Schritte

Wie bringt man die Punktwoke in einem für uns lesbaren Format auf den PC?

Leica BLK2GO:

1. BLK mit PC verbinden
2. Mit dem BLK Manager die b2g-Dateien auf den PC laden oder direkt in Cyclone importieren
3. In Cyclone mit dem Direktexport zu e57 konvertieren (Punktwolke kann vorher bearbeitet werden)

GeoSLAM ZEB Revo-RT

1. Feldcomputer starten, mit dem Netz von Feldcomputer verbinden und die IP abrufen
2. Von dort die Punktwolke in PLY oder .geoSLAM downloaden.
3. Import der Punktwolke in die Software GeoSLAM Hub. Dort kann man die Punktwolke je nach Dateiformat mit zusätzlichen Informationen ausstatten wie z.B. Farbe anhand der Zeit, Höhe, Shape, SLAM Kondition, Farbwerte der GoPro (siehe farbige Punktwolke), Zeitstempel relativ oder absolut, Smooth und Punktnormalen (Ausser in LAS/LAZ)

SiteScape (iPad)

1. Datelexport in Filesystem des iPads
2. Unter Windows: Synchronisieren mit iTunes und Datei herunterladen

Scaniverse (iPad)

1. Prozessieren (Standard, High oder Ultra)
2. Datelexport in Filesystem des iPads

3. Unter Windows: Synchronisieren mit iTunes und Datei herunterladen

3d Scanner App (iPad)

1. Prozessieren (HD, Medium, Fast oder Custom)
2. Datelexport in Filesystem des iPads
3. Unter Windows: Synchronisieren mit iTunes und Datei herunterladen

Polycam (iPad)

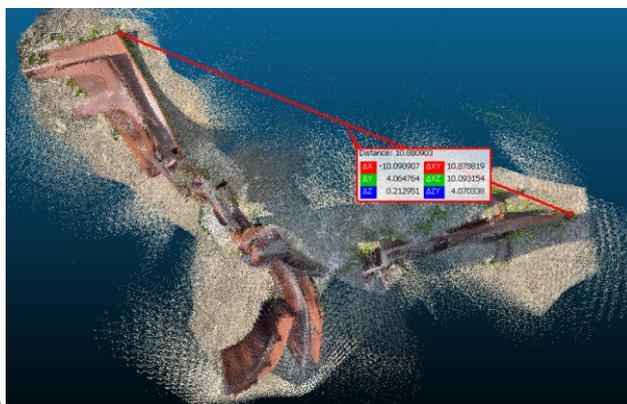
1. Prozessieren (Fast, Space, Object oder Custom)
2. Datelexport in Filesystem des iPads
3. Unter Windows: Synchronisieren mit iTunes und Datei herunterladen

Vergleich Messung Skulptur

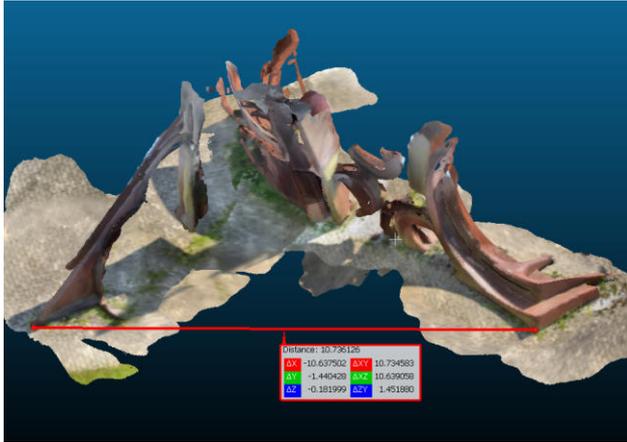
Vergleichspunkt	Leica BLK2GO	GeoSLAM Revo RT	SiteScape (iPad Pro)	Scaniverse (iPad Pro)	3d Scanner App (iPad Pro)	Polycam (iPad Pro)
Geschwindigkeit 1 Messung	2 min	4 min	15 min	15 min	15min	10min
Anzahl Punkte	11718645	1311176	2870169	1215852	686211	1351140
Box-Dimension im CloudCompare [XxYxZ]	4.2x5.0x3.9	87.3x43.3x10.0	16.4x16.5x8.4	14.9x9.6x5.2	13.4x5.2x8.5	13.8x12.4x5.7
Punktabstand ¹	0.023	0.055	0.047	0.020	0.030	0.062
Outliers	Sehr wenige ^{Bild6}	Wenige fast ausschliesslich nur weit aussen	Viele ^{Bild4}	Viele Bereiche verschoben und Löcher. Nicht viele einzelne Outliers ^{Bild2}	Einige Bereiche verschoben und Löcher. Nicht viele einzelne Outliers ^{Bild5}	Viele Bereiche verschoben und Löcher. Nicht viele einzelne Outliers ^{Bild3}
3d Completion	100%	100%	90%	80%	75%	75%
2D Grundriss ermittelbar	Ja	Ja	Ja	Ja	Ja	Schwer
Höhe ermittelbar	Ja	Ja	Nein	Nein	Nein	Nein
Georeferenziert	Nein	Nein	Nein	Nein	Ja mittels GPS, 5m Genauigkeitsrange	Nein
Länge von A nach B	11.05	11.19	10.88	10.74	10.94	10.56

¹Je drei Abstände gemessen und gemittelt.

Messung von A nach B



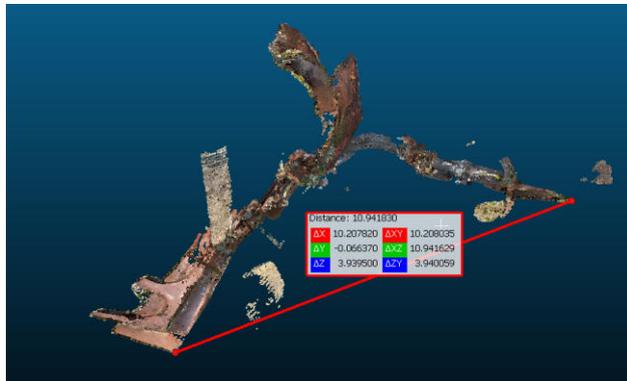
iPad Pro: SiteScape



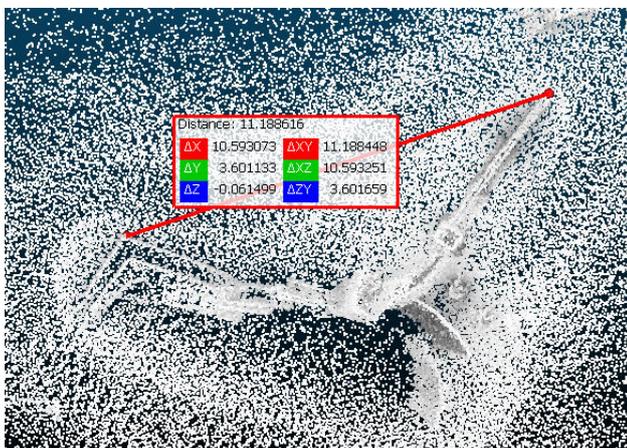
iPad Pro: Scaniverse



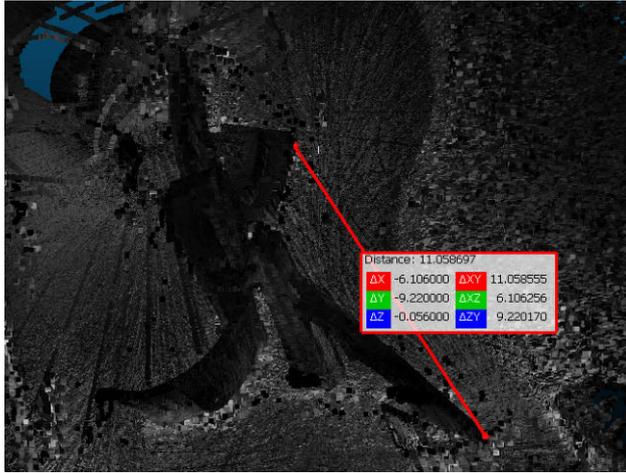
iPad Pro: Polycam



iPad Pro: 3d Scanner App



GeoSLAM Rev0 RT



Leica BLK2GO

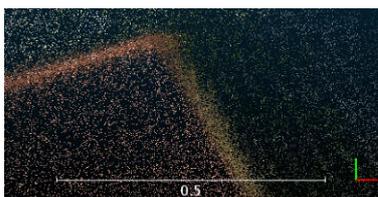
iPad Pro: SiteScape



Front:



Top:



Punkte:

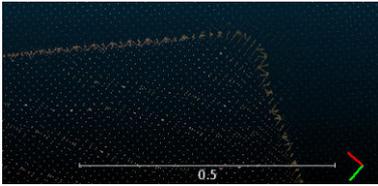
iPad Pro: Scaniverse



Front:



Top:



Punkte:

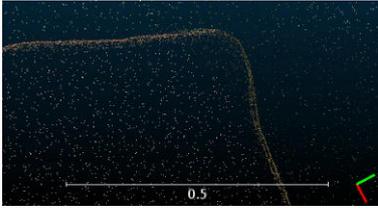
iPad Pro: Polycam



Front:



Top:



Punkte:

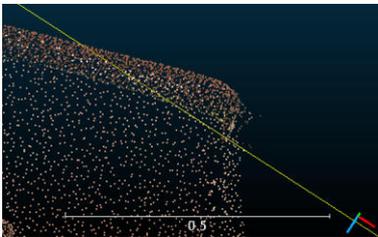
iPad Pro: 3d Scanner App



Front:



Top:



Punkte:

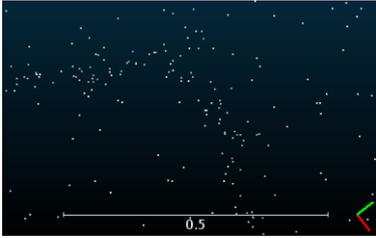
GeoSLAM Revo RT



Front:



Top:



Punkte:

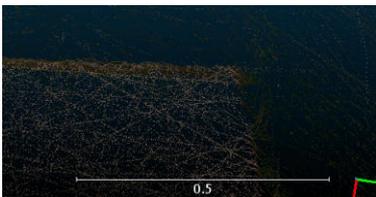
Leica BLK2GO



Front:



Top:



Punkte:

Vergleich Messung Burgmauer

Vergleichspunkt	Leica BLK2GO	GeoSLAM Revo RT^4	Sitescape (iPad Pro)	3d Scanner App (iPad Pro)	Scaniverse (iPad Pro)	Polycam (iPad Pro)	Structure from Motion
Geschwindigkeit 1 Messung	5min	5min	20min	20min	20min	20min	20min
Punktanzahl	23346161	2634453	18471492	2234417	1038461	815460	32939936
Box-Dimension im CloudCompare [XxYxZ]	72.29x45.81x28.53	62.58x41.58x23.37	30.98x9.44x8.12	36.38x9.48x5.63	30.48x5.67x5.44	31.81x12.94x5.45	37.60x13.87x8.15

Vergleichspunkt	Leica BLK2GO	GeoSLAM Revo RT^4	Sitescape (iPad Pro)	3d Scanner App (iPad Pro)	Scaniverse (iPad Pro)	Polycam (iPad Pro)	Structure from Motion
Punktabstand ¹	0.014	0.040	0.029	0.014	0.018	0.025	0.006
Outliers	Ja aber nicht dramatisch	Ja aber nicht dramatisch	Ja aber nicht dramatisch	Kaum	Kaum	Outliers und Verschiebungen	Kaum
3d Completion	95%	95%	90%	70%	85%	85%	100%
2D Grundriss ermittelbar	Ja	Ja	Ja	Nein	Ja	Ja	Ja
Höhe bei jedem Mauerabschnitt ermittelbar	Fast	Ja	Nein	Nein	Nein	Nein wegen Verschiebung	Nein
Georeferenziert	Nachträglich mit 4 Referenzpunkten	Nachträglich mit 4 Referenzpunkten	Nachträglich mit 5 Referenzpunkten	Nachträglich mit 4 Referenzpunkten	Nachträglich mit 4 Referenzpunkten	Nachträglich mit 4 Referenzpunkten	Nachträglich mit 10 Referenzpunkten
Messung A nach B 17.21 m	17.25	17.52 ³	16.79	Nicht möglich	16.99 ²	17.01 ²	17.24
Messung C nach D 2.92 m	2.92	2.90 ³	2.94	2.90	2.93	2.90	2.93
Messung E nach F 3.31 m	3.33	3.41 ³	3.24	3.17	3.33	2.28	3.32

¹ Je drei Abstände gemessen und gemittelt.

² Ein Punkt nicht erkennbar -> nächst möglichen verwendet.

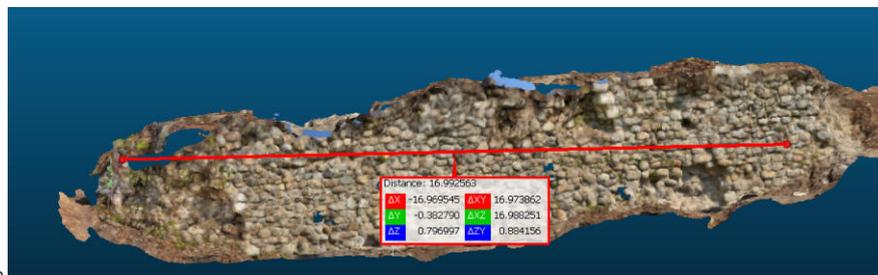
³ Das GeoSLAM hat in diesem Test schlechte Resultate erzielt, da die Punktdichte und eher schlechte Einfärbung das Erkennen der Referenzpunkte unmöglich gemacht hat. Deshalb wurde bei der Villa das Schritttempo angepasst, damit eine bessere Punktdichte und besser Analyseresultate erzielt werden konnten.

Messung von A nach B

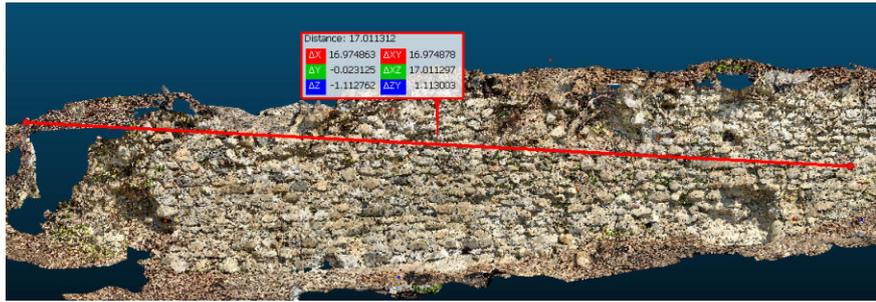
In Realität ca 17.21 m.



iPad Pro: SiteScape



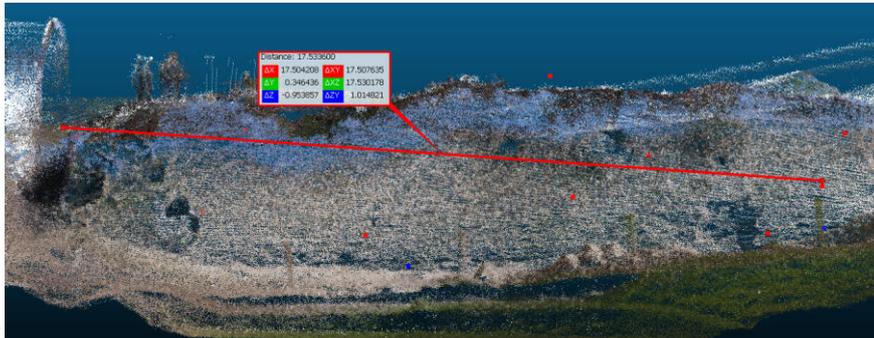
iPad Pro: Scaniverse



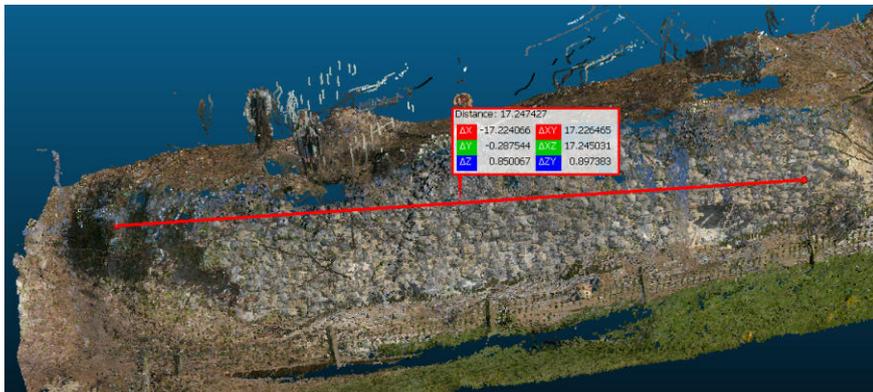
iPad Pro: Polycam



iPad Pro: 3d Scanner App



GeoSLAM Revo RT



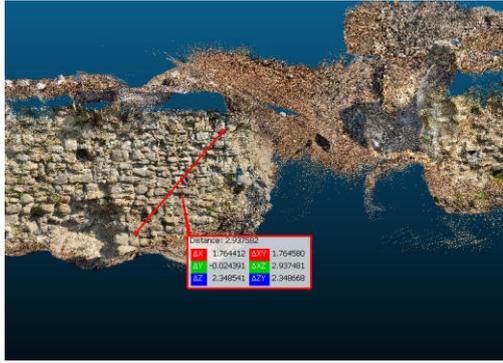
Leica BLK2GO



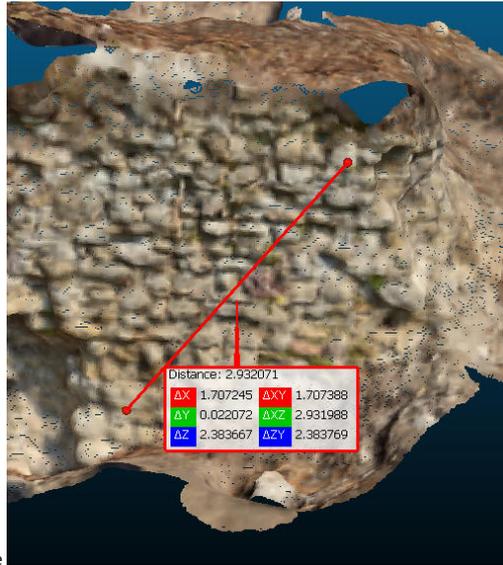
Structure from Motion:

Messung von C nach D

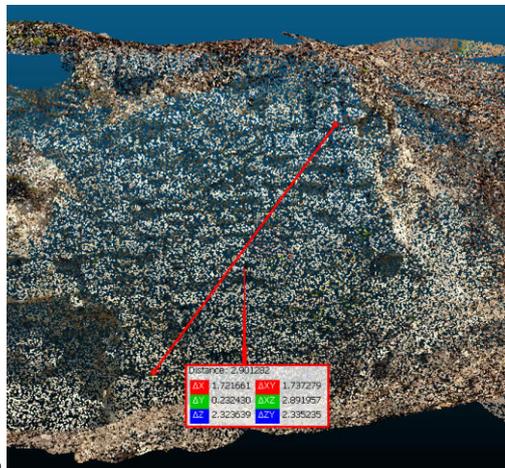
In Realität ca 2.92 m.



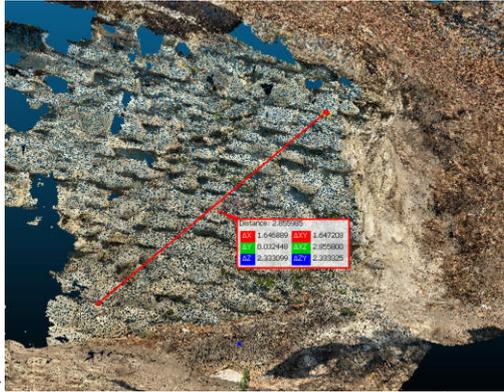
iPad Pro: SiteScape



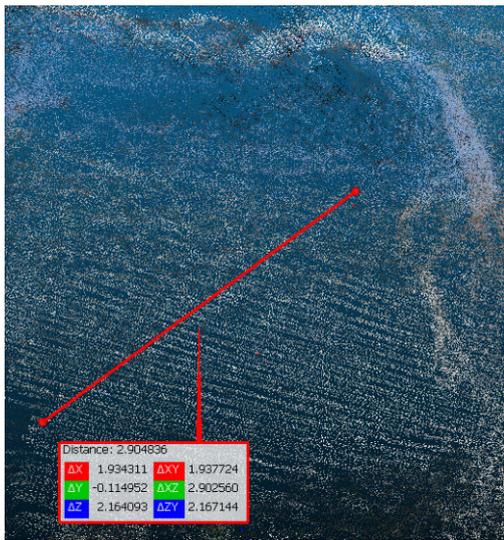
iPad Pro Scaniverse



iPad Pro Polycam



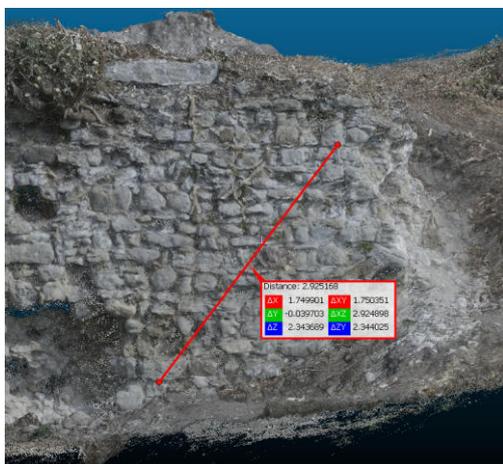
iPad Pro 3d Scanner



GeoSLAM Revo RT



Leica BLK2GO

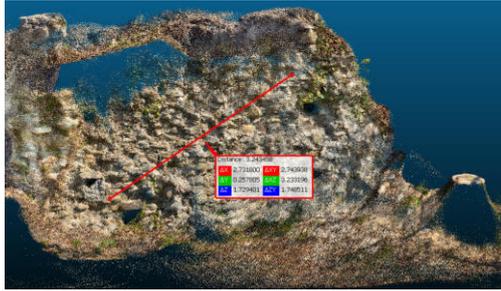


Structure from Motion

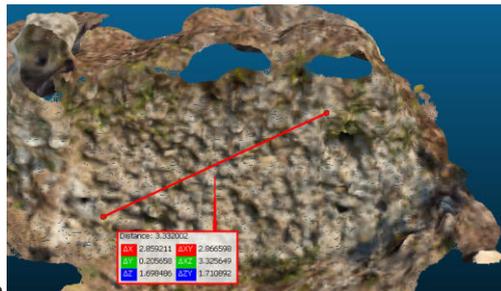
Messung von E nach F

In Realität ca 3.31 m.

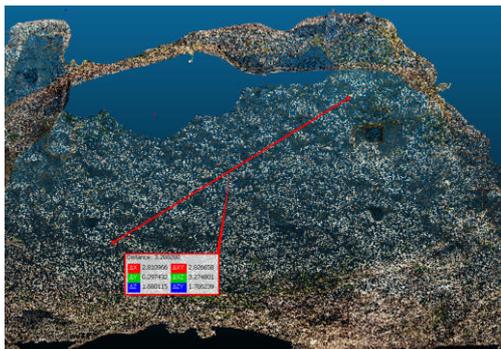
iPad Pro: SiteScape



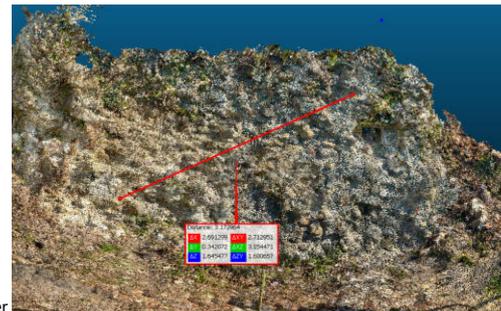
iPad Pro Scaniverse



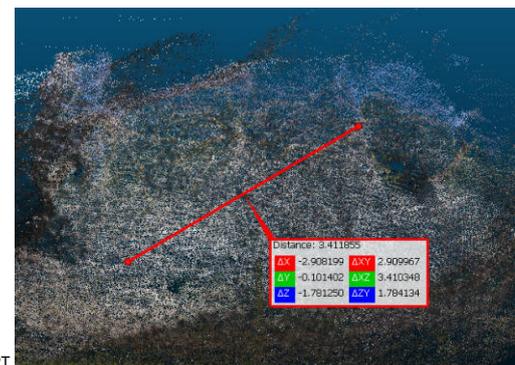
iPad Pro Polycam

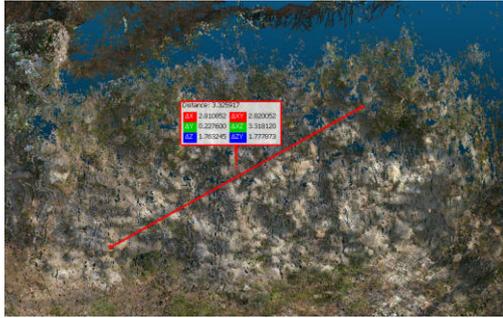


iPad Pro 3d Scanner

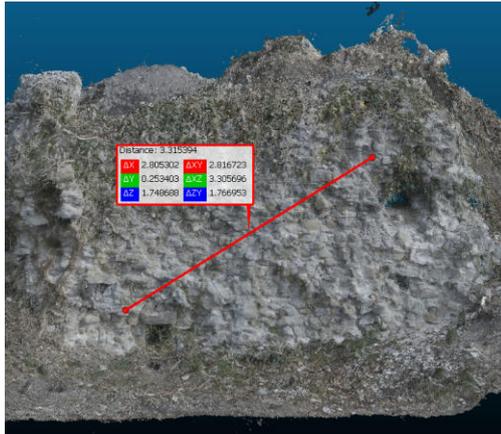


GeoSLAM Revo RT





Leica BLK2GO



Structure from Motion

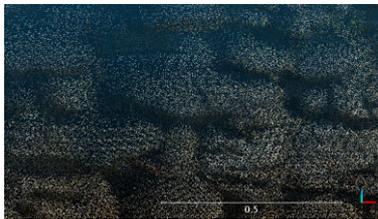
iPad Pro: SiteScape



Front:



Top:



Punkte:

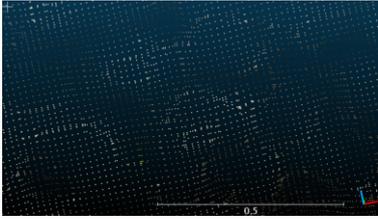
iPad Pro: Scaniverse



Front:



Top:



Punkte:

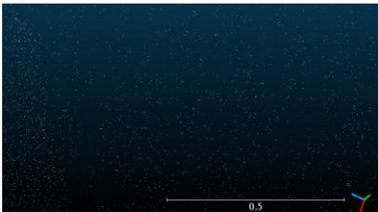
iPad Pro: Polycam



Front:

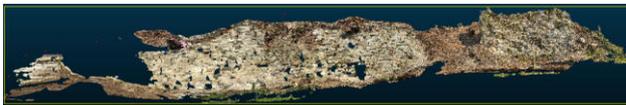


Top:



Punkte:

iPad Pro: 3d Scanner App



Front:



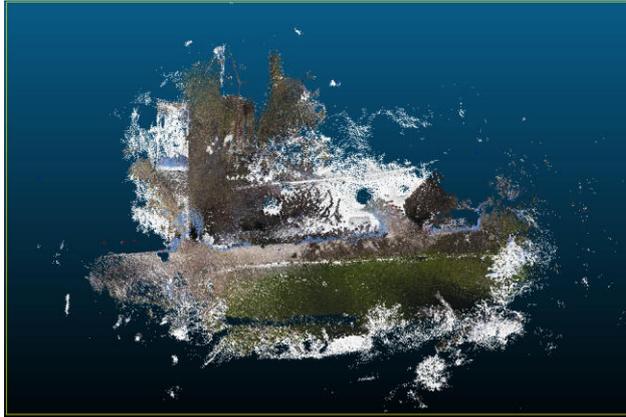
Top:

Punkte: ![3d Scanner App Punkte](

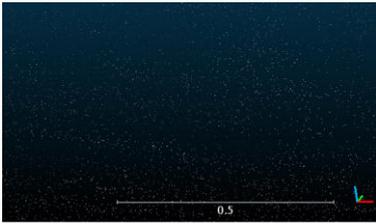
GeoSLAM Revo RT



Front:



Top:



Punkte:

Leica BLK2GO



Front:

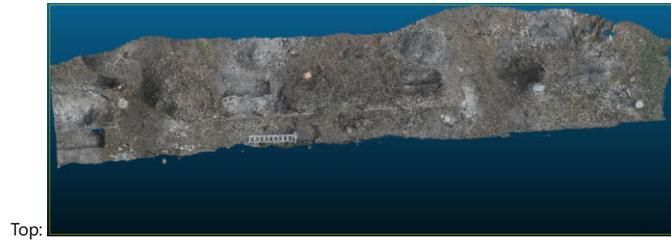


Top:



Punkte:

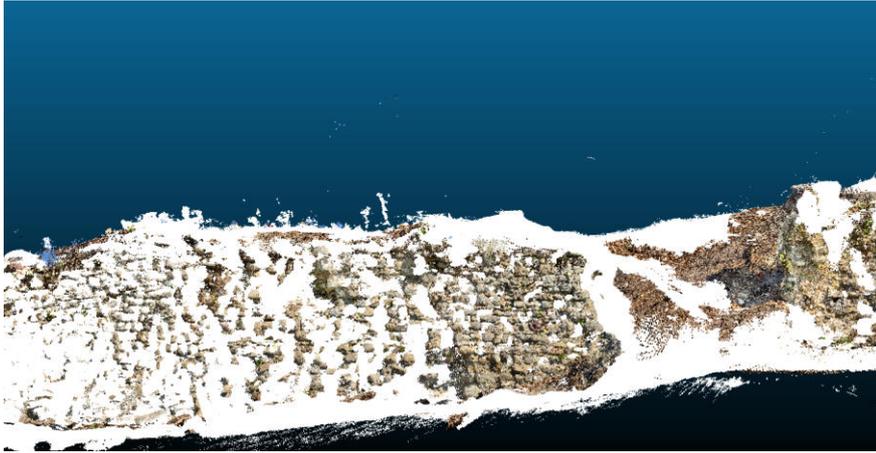
Structure for Motion



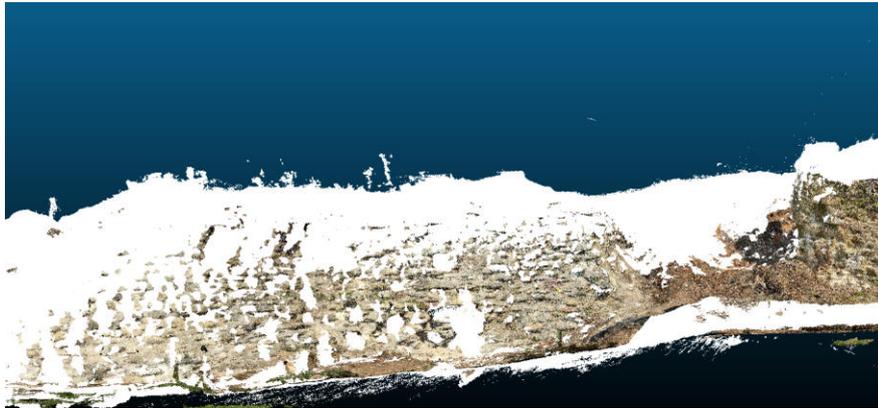
Überlagerungen

In diesem Abschnitt wurden die referenzierten Punktwolken übereinander dargestellt. (Weiss sind die Referenzdaten, die in farbiger Variante hier zu sehen ist.)

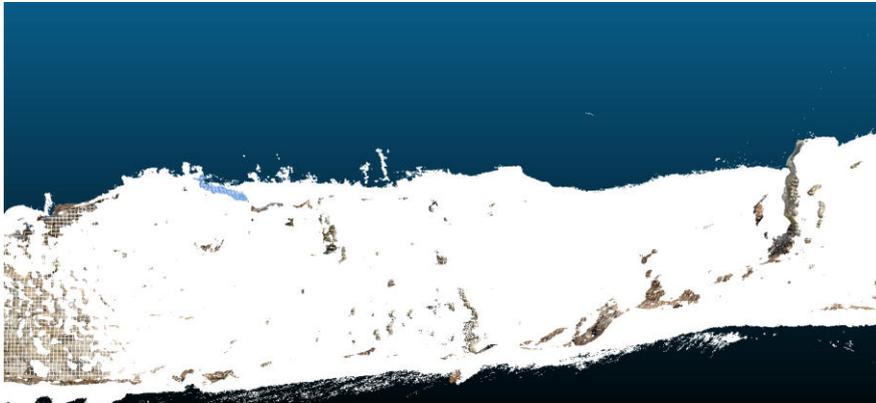




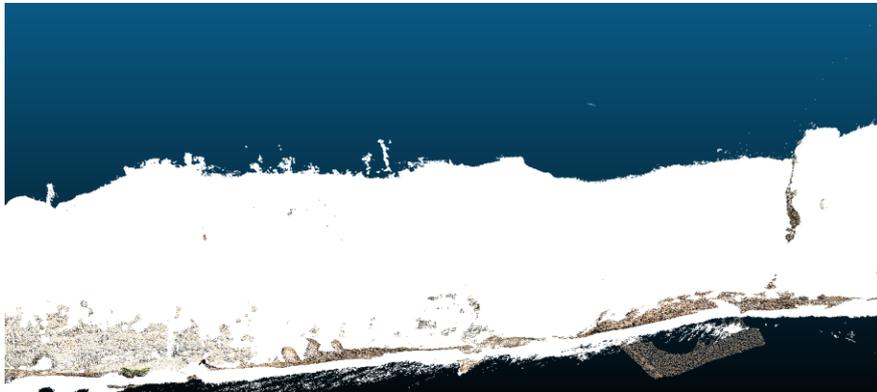
Sitescape (iPad Pro):



3d Scanner App (iPad Pro):



Scaniverse (iPad Pro):



Polycam (iPad Pro):

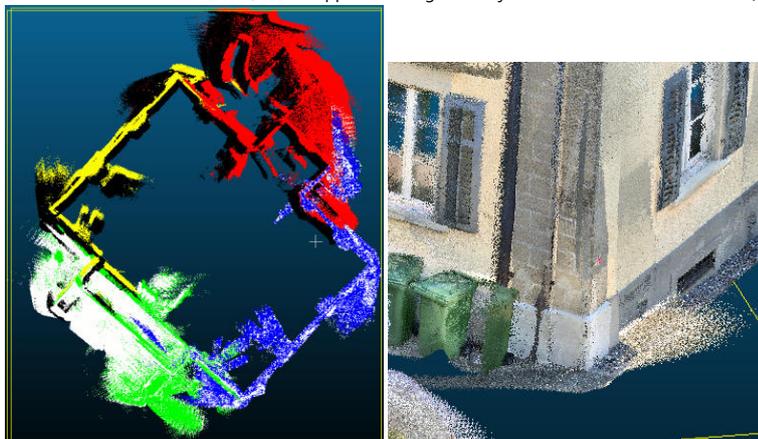
Vergleich Messung Villa

Vergleichspunkt	Leica BLK2GO	GeoSLAM Revo RT	Sitescape (iPad Pro) ¹	3d Scanner App (iPad Pro)	Scaniverse (iPad Pro)	Polycam (iPad Pro)
Geschwindigkeit 1 Messung	6min	11min	20min	25min	15min	15min
Punktzahl	37186415	1285745	37807598	6357459	1223093	932700
Box-Dimension im CloudCompare [XxYxZ]	66.40x60.26x22.80	82.37x91.25x24.04	24.56x20.53x10.49	21.51x18.83x8.16	19.41x22.28x7.60	23.29x20.42x8.14
Outliers	Wenige	Ja	Ja	Viele Verschiebungen bei Boden und Fassaden zueinander.	Viele Verschiebungen beim Eingang (Start und Ende).	Viele Verschiebungen beim Eingang (Start und Ende).
3d Completion	99%	99%	95%	95%	90%	90%
2D Grundriss ermittelbar	Ja	Ja	Ja	Grob ja genau nein	Ja	Grob ja genau nein
Fassaden Höhe ermittelbar	Ja	Ja	Nein (zu kleiner Scanbereich)	Nein (Shifts und nicht alles im Scanbereich)	Nein (Scanbereich zu klein)	Nein (Scanbereich zu klein)
Messpunkte Vergleich ²	Sehr gut, Abweichungen von 0.06	Messpunkte nicht erkennbar, aber passt dennoch sehr gut	Passt gut, Abweichungen von 0.39	Passt sehr schlecht, Abweichungen von 1.05 (Ursache dafür werden die vielen unsaubereren Verschiebungen sein.)	Passt gut, Abweichungen von 0.46	Passt schlecht, Abweichungen von 0.6
Messungen unterteilt	Nein	Nein	Ja 4 Teile	Nein	Nein	Nein
Messung A nach B ³ 10.68 m	10.72	10.69	10.36	10.11	10.41	10.37
Messung C nach D ³ 10.64 m	10.67	10.60	10.24	9.99	10.33	10.42
Messung E nach F 13.47 m	13.42	13.41	13.15	12.92	13.15 (Beide Punkte geschätzt nicht erkannt)	12.97 (Ein Punkt geschätzt nicht erkannt)

²Passen die Referenzpunkte vom Tachy zur Aufnahme?

³ Für die Fassadenlänge wurden aus den schiefen Messungen nur die Längen in XY Richtung, ohne die Höhe Z verwendet

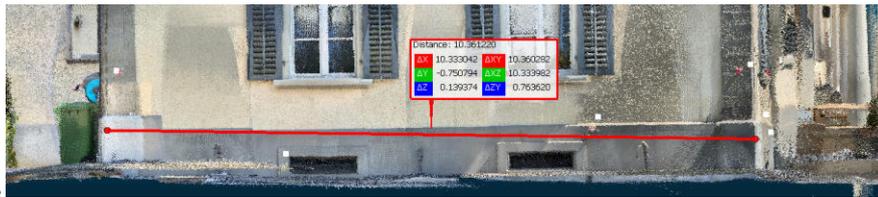
¹ Selber referenzierte Einzelteile, da die App nicht das ganze Objekt auf einmal scannen konnte. (Schwarzweiss das eigene, farbig das Automatische):



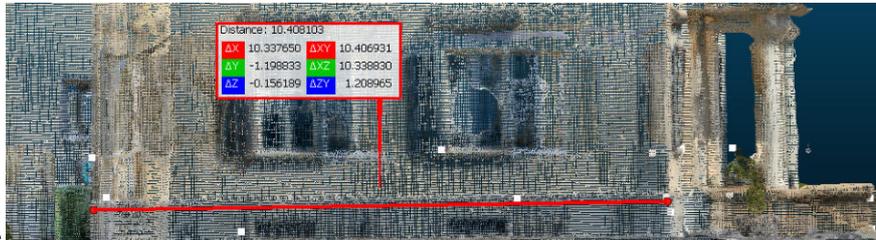
Bei mehreren Messungen wurde das beste Resultat verwendet.

Messung von A nach B

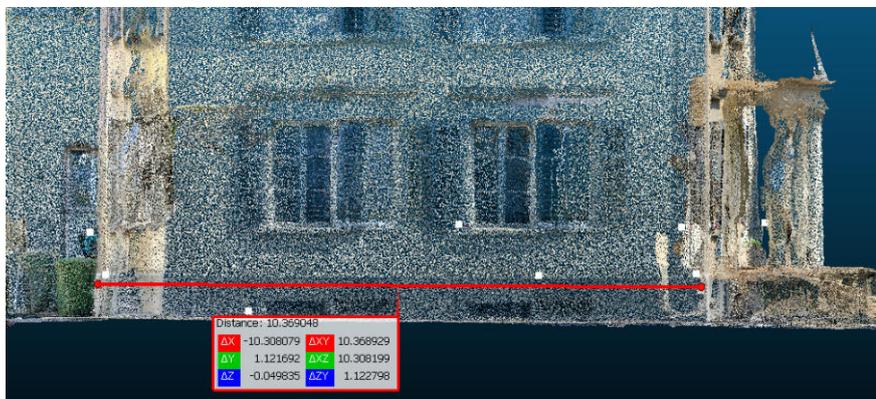
In Realität ca 10.68 m (Aus AV).



iPad Pro: SiteScape



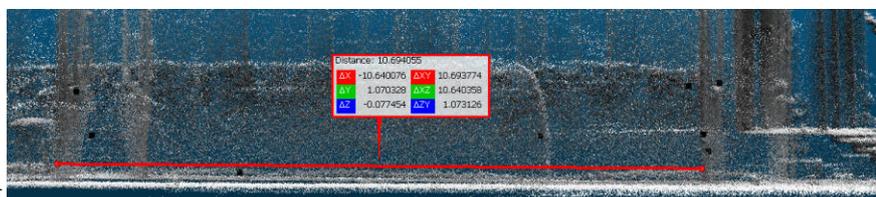
iPad Pro: Scaniverse



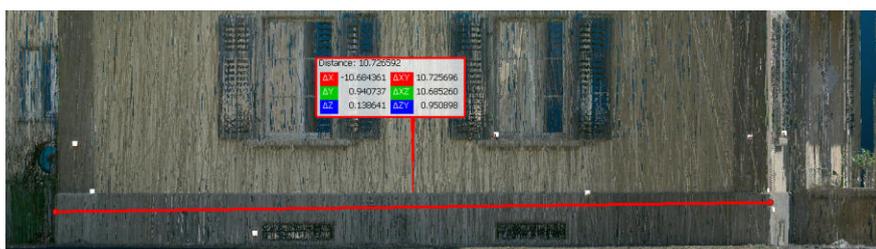
iPad Pro: Polycam



iPad Pro: 3d Scanner App



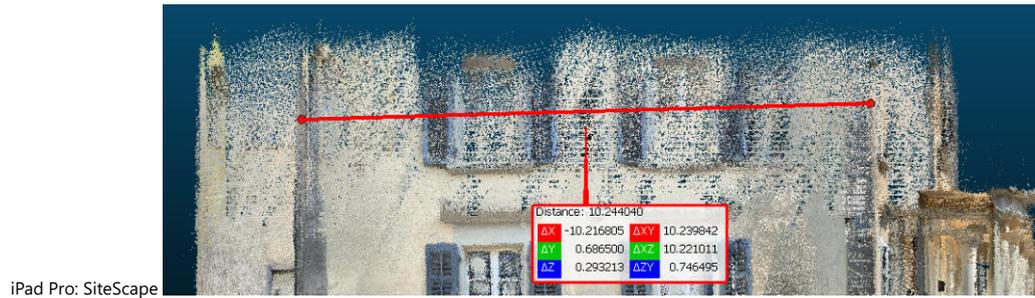
GeoSLAM Revo RT



Leica BLK2GO

Messung von C nach D

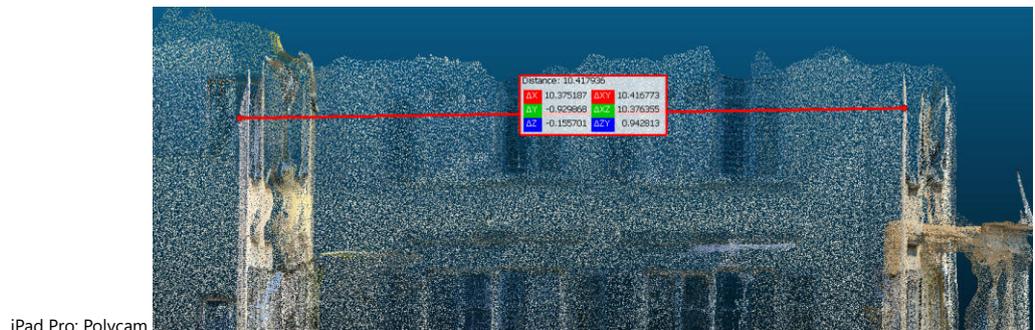
In Realität ca 10.64 m (AV minus 2x 2cm Sockel).



iPad Pro: SiteScape



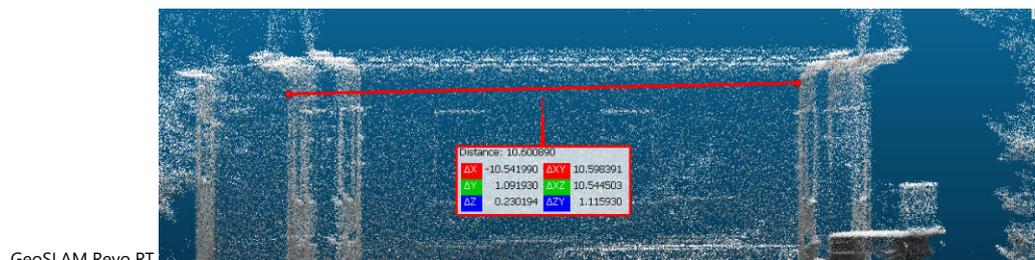
iPad Pro: Scaniverse



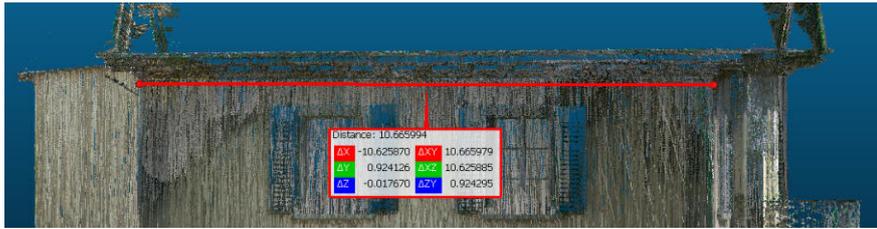
iPad Pro: Polycam



iPad Pro: 3d Scanner App



GeoSLAM Rev0 RT

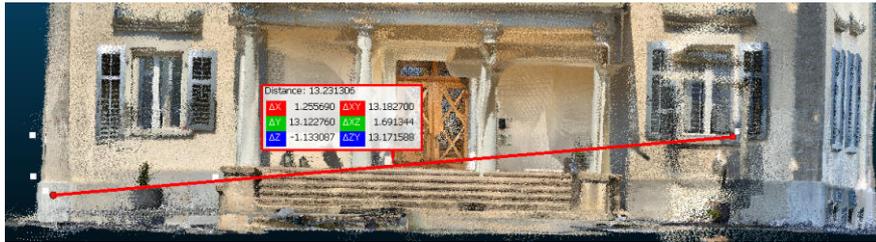


Leica BLK2GO

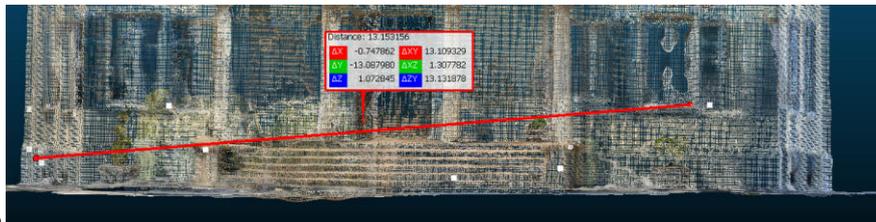
Messung von E nach F

In Realität ca 13.47 m.

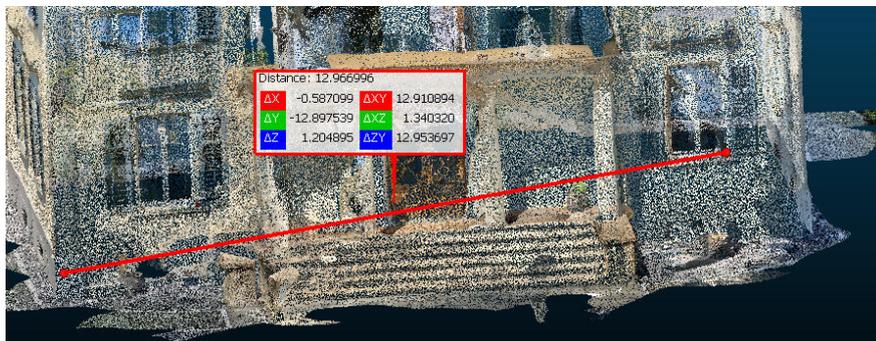
iPad Pro: SiteScape



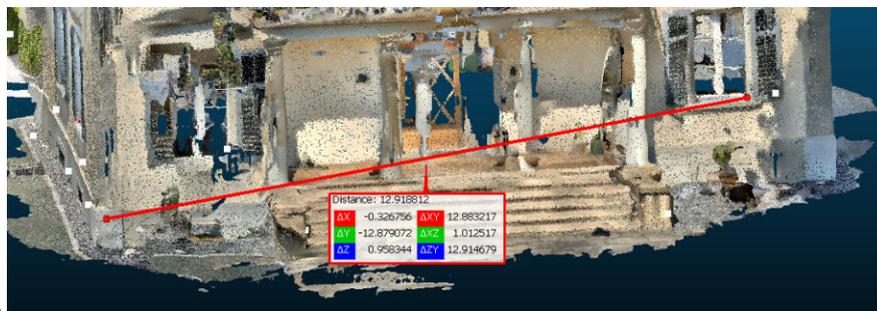
iPad Pro: Scaniverse

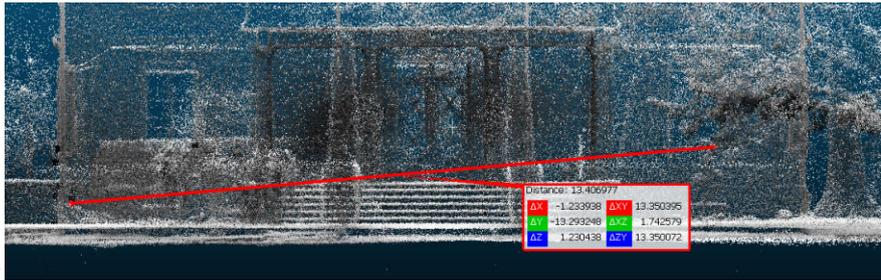


iPad Pro: Polycam



iPad Pro: 3d Scanner App



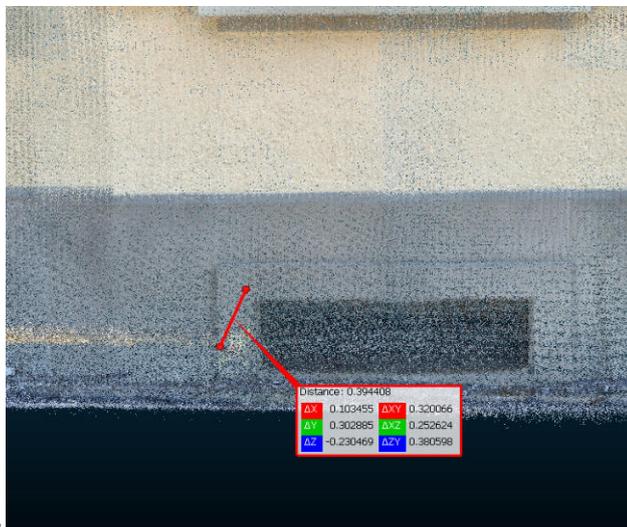


GeoSLAM Revo RT

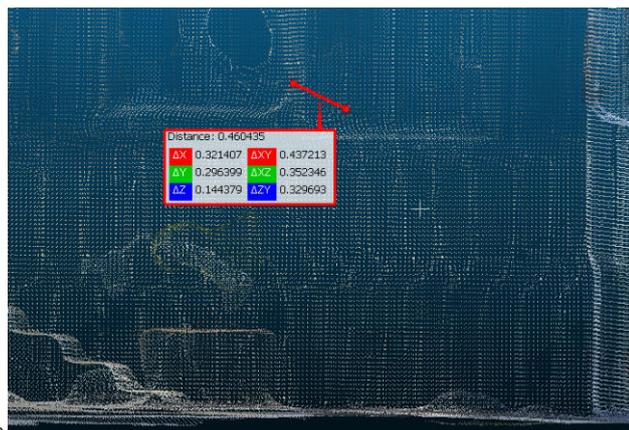


Leica BLK2GO

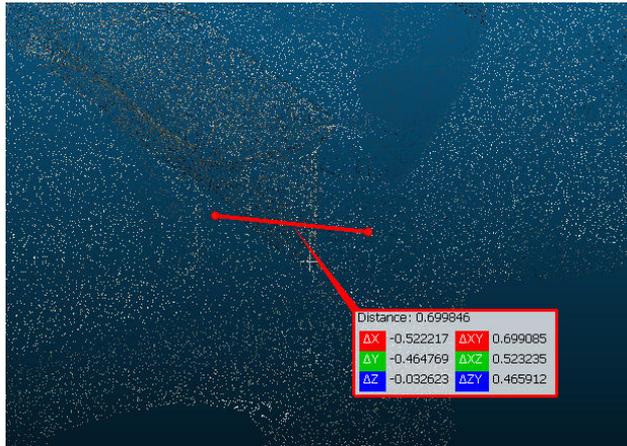
Abstände von Objekt zu Tachymeter-Referenzdaten:



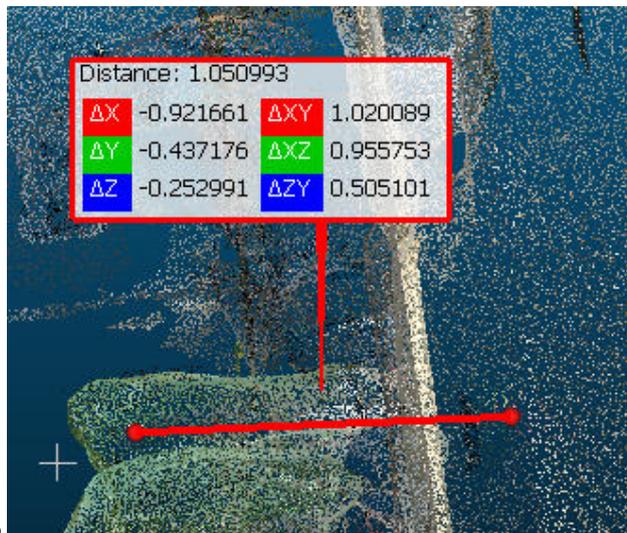
iPad Pro: SiteScape



iPad Pro: Scaniverse



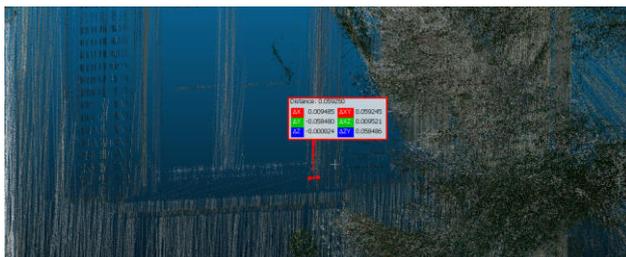
iPad Pro: Polycam



iPad Pro: 3d Scanner App

GeoSLAM Revo RT

Kein Bild, da die Referenzpunkte nicht sichtbar sind und für die Referenzierung geschätzt wurden. Dennoch passen die Messungen gut.



Leica BLK2GO

iPad Pro: SiteScape



Front:



Side:



Top:



Punkte:

iPad Pro: Scaniverse



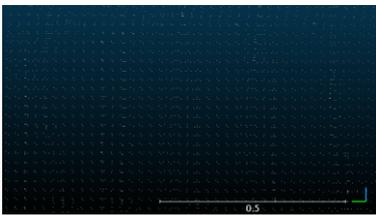
Front:



Side:

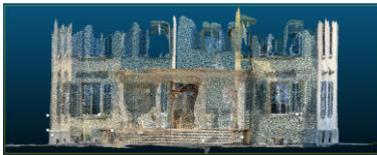


Top:



Punkte:

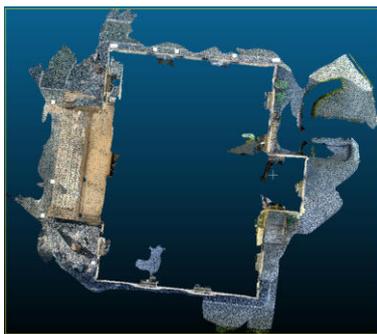
iPad Pro: Polycam



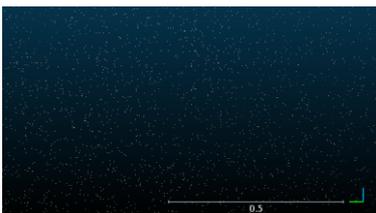
Front:



Side:



Top:



Punkte:

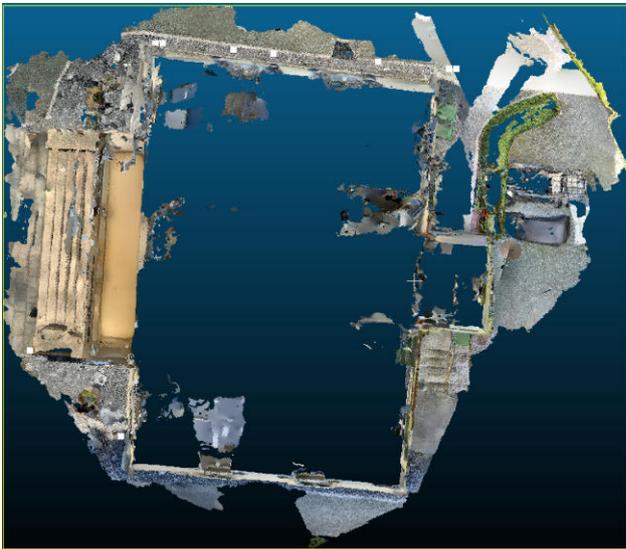
iPad Pro: 3d Scanner App



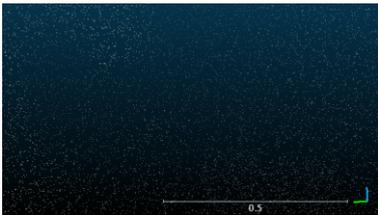
Front:



Side:



Top:



Punkte:

GeoSLAM Revo RT



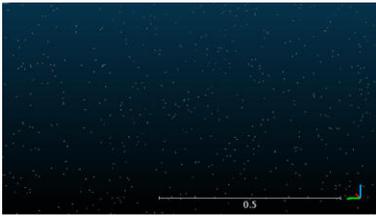
Front:



Side:



Top:



Punkte:

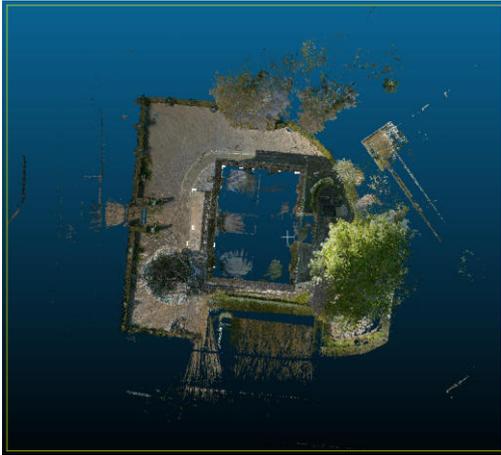
Leica BLK2GO



Front:

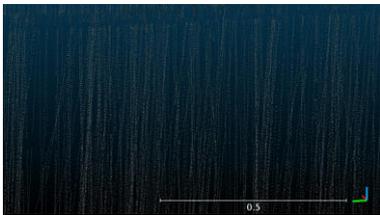


Side:



Top:

Punkte:



Verweise

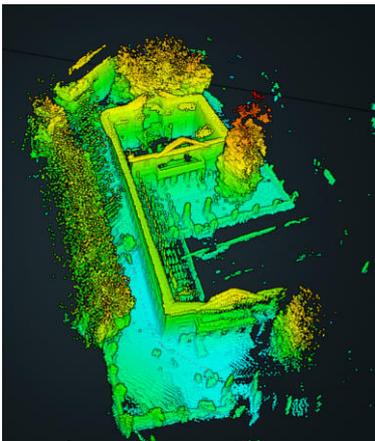


Bild1

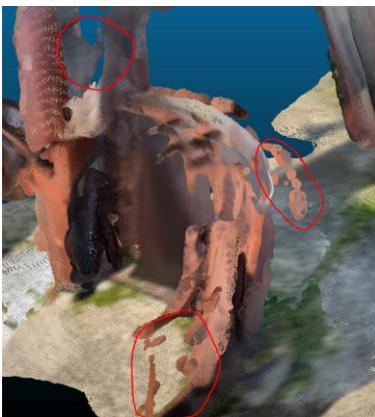


Bild2



Bild3



Bild4

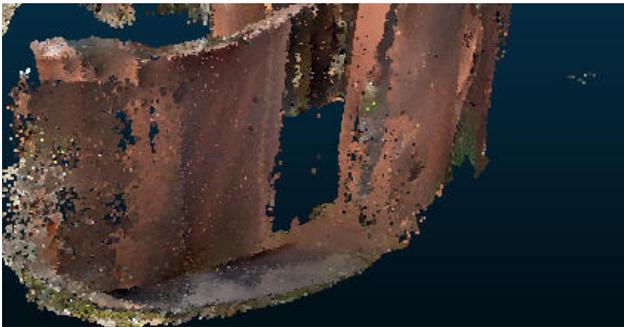


Bild5



Bild6

3 2D Berechnungen

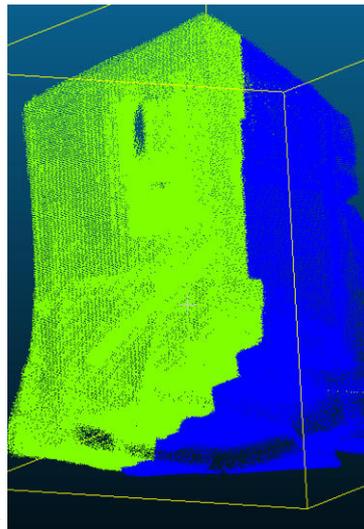
2D Ansicht

PDAL Boden-Filter Tests

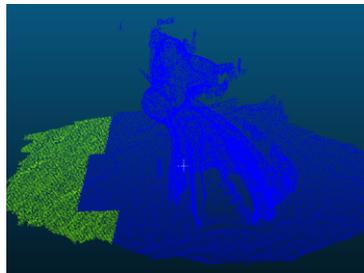
- `filters.csf`: Der Cloth Simulation Filter (CSF) klassifiziert Bodenpunkte basierend auf dem in [Zhang2016] beschriebenen Ansatz. Zhang2016: Zhang, Wuming et al. "Eine benutzerfreundliche LiDAR-Datenfiltermethode in der Luft, die auf einer Stoffsimulation basiert." *Remote Sensing* 8.6 (2016): 501.

```
(pdal pipeline filters.csf Warning) Both NumberOfReturns and ReturnNumber are filled with 0's. Proceeding without any further return filtering.  
PDAL: bad allocation
```

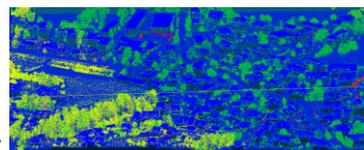
- `filters.smrf`: Der Simple Morphological Filter (SMRF) klassifiziert Bodenpunkte basierend auf dem in [Pingel2013] beschriebenen Ansatz. Pingel2013: Pingel, Thomas J., Keith C. Clarke und William A. McBride. "Ein verbesserter einfacher morphologischer Filter fe 21–30.



Gebäude aufnahme iPad:



Skulptur aufnahme iPad:

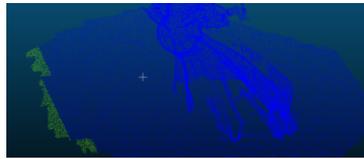


Uetikon aufnahme Kt ZH:

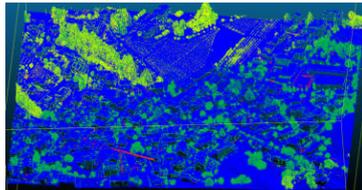
- `filters.pmf`: Der Progressive Morphological Filter (PMF) ist eine Methode zur Segmentierung von Boden- und Nicht-Bodenrenditen. Dieser Filter ist eine Implementierung der in [Zhang2003] beschriebenen

Methode. Zhang2003: Zhang, Keqi et al. "Ein progressiver morphologischer Filter zum Entfernen von Nichtgrundmessungen aus LIDAR-Daten in der Luft." *Geowissenschaften und Fernerkundung, IEEE Transactions on* 41.4 (2003): 872-882.

Skulptur aufnahme iPad:

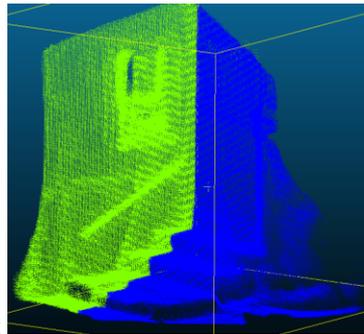


Uetikon aufnahme Kt ZH:



- filters.skewnessbalancing: Skewness Balancing klassifiziert Bodenpunkte basierend auf dem in [Bartels2010] beschriebenen Ansatz. Bartels2010: Bartels, Marc und Hong Wei. "Schwellenfreie Objekt- und Bodenpunktrennung in LIDAR-Daten." *Mustererkennungsschreiben* 31.10 (2010): 1089-1099.

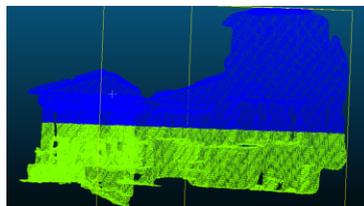
Gebäude aufnahme iPad:



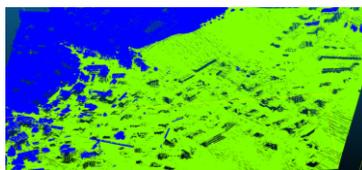
Skulptur aufnahme iPad:



Gebäude2 aufnahme iPad:



Uetikon aufnahme Kt ZH:



Die Test sind alle Fehlgeschlagen und scheinen auf Airborne Laserscanning ausgelegt zu sein.

PCL Boden-Filter Tests



Die Test sind alle ergebnislos geblieben und wuren nach ca 15 bis 30 min abgebrochen

PCL Read PCD and Write GeoTIFF

```
#include <iostream>
#include <pcl/io/pcd_io.h>
#include <pcl/io/png_io.h>
#include <pcl/point_types.h>
#include <pcl/range_image/range_image.h>
#include <pcl/visualization/common/float_image_utils.h>

int main (int argc, char** argv) {

    pcl::PointCloud<pcl::PointXYZ>::Ptr pointCloud(new
    pcl::PointCloud<pcl::PointXYZ>);
    if (pcl::io::loadPCDFile<pcl::PointXYZ>("../test.pcd", *pointCloud) == -1) /**
    load the file
    {
        PCL_ERROR("Couldn't read file test.pcd \n");
        return (-1);
    }

    float angularResolution = (float) ( 1.0f * (M_PI/180.0f)); // 1.0 degree in
    radians
    float maxAngleWidth      = (float) (360.0f * (M_PI/180.0f)); // 360.0 degree in
    radians
    float maxAngleHeight    = (float) (180.0f * (M_PI/180.0f)); // 180.0 degree in
    radians
    Eigen::Affine3f sensorPose = (Eigen::Affine3f)Eigen::Translation3f(1.0f, 4.5f,
    1.0f);
    pcl::RangeImage::CoordinateFrame coordinate_frame =
    pcl::RangeImage::CAMERA_FRAME;
    float noiseLevel=0.00;
    float minRange = 0.0f;
    int borderSize = 1;

    pcl::RangeImage rangeImage;
    rangeImage.createFromPointCloud(*pointCloud, angularResolution, maxAngleWidth,
    maxAngleHeight,
                                sensorPose, coordinate_frame, noiseLevel,
    minRange, borderSize);

    std::cout << rangeImage << "\n";

    float* ranges = rangeImage.getRangesArray ();
```

```

    unsigned char* rgb_image = pcl::visualization::FloatImageUtils::getVisualImage
    (ranges, rangeImage.width, rangeImage.height);

    pcl::io::saveRgbPNGFile("ha.png", rgb_image, rangeImage.width,
    rangeImage.height);

    return (0);
}

```

PCL region_growing_segmentation.cpp

```

#include <iostream>
#include <vector>
#include <pcl/point_types.h>
#include <pcl/io/pcd_io.h>
#include <pcl/search/search.h>
#include <pcl/search/kdtree.h>
#include <pcl/features/normal_3d.h>
#include <pcl/visualization/cloud_viewer.h>
#include <pcl/filters/passthrough.h>
#include <pcl/segmentation/region_growing.h>

int
main (int argc, char** argv)
{
    pcl::PointCloud<pcl::PointXYZ>::Ptr cloud (new pcl::PointCloud<pcl::PointXYZ>);
    if ( pcl::io::loadPCDFile <pcl::PointXYZ> ("../test.pcd", *cloud) == -1)
    {
        std::cout << "Cloud reading failed." << std::endl;
        return (-1);
    }

    pcl::search::Search<pcl::PointXYZ>::Ptr tree (new
    pcl::search::KdTree<pcl::PointXYZ>);
    pcl::PointCloud <pcl::Normal>::Ptr normals (new pcl::PointCloud <pcl::Normal>);
    pcl::NormalEstimation<pcl::PointXYZ, pcl::Normal> normal_estimator;
    normal_estimator.setSearchMethod (tree);
    normal_estimator.setInputCloud (cloud);
    normal_estimator.setKSearch (50);
    normal_estimator.compute (*normals);

    pcl::IndicesPtr indices (new std::vector <int>);
    pcl::PassThrough<pcl::PointXYZ> pass;
    pass.setInputCloud (cloud);
    pass.setFilterFieldName ("z");
    pass.setFilterLimits (0.0, 1.0);
    pass.filter (*indices);

    pcl::RegionGrowing<pcl::PointXYZ, pcl::Normal> reg;
    reg.setMinClusterSize (50);

```

```

reg.setMaxClusterSize (1000000);
reg.setSearchMethod (tree);
reg.setNumberOfNeighbours (30);
reg.setInputCloud (cloud);
//reg.setIndices (indices);
reg.setInputNormals (normals);
reg.setSmoothnessThreshold (3.0 / 180.0 * M_PI);
reg.setCurvatureThreshold (1.0);

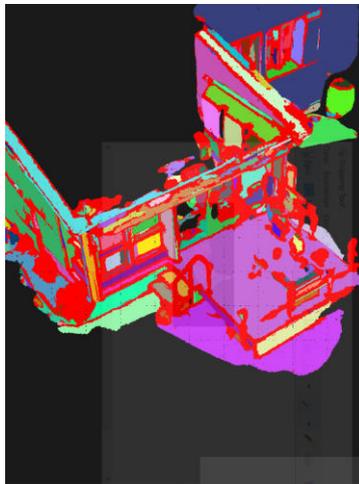
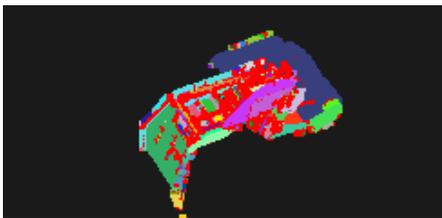
std::vector <pcl::PointIndices> clusters;
reg.extract (clusters);

std::cout << "Number of clusters is equal to " << clusters.size () << std::endl;
std::cout << "First cluster has " << clusters[0].indices.size () << " points."
<< std::endl;
std::cout << "These are the indices of the points of the initial" <<
std::endl << "cloud that belong to the first cluster:" << std::endl;
int counter = 0;
while (counter < clusters[0].indices.size ())
{
std::cout << clusters[0].indices[counter] << ", ";
counter++;
if (counter % 10 == 0)
std::cout << std::endl;
}
std::cout << std::endl;

pcl::PointCloud <pcl::PointXYZRGB>::Ptr colored_cloud = reg.getColoredCloud ();
pcl::visualization::CloudViewer viewer ("Cluster viewer");
viewer.showCloud(colored_cloud);
while (!viewer.wasStopped ())
{
}

return (0);
}

```



DOM PDAL (END LÖSUNG)

VIIIA Test mit GEOSLAM

```

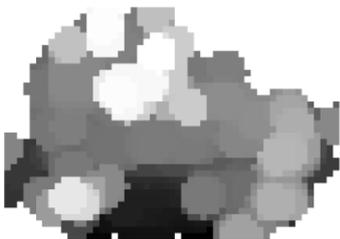
"filename":"output3.tif",
"resolution":0.05,
"output_type":"max",
"radius":0.05,
"type": "writers.gdal"

"filename":"output1.tif",
"resolution":0.1,
"output_type":"max",
"radius":0.15,
"type": "writers.gdal"

"filename":"output.tif",
"resolution":1.0,
"output_type":"max",
"radius":3.0,
"type": "writers.gdal"

```

BURG Test mit GeoSLAM



```

"" "filename":"burgOutput3.tif", "resolution":1.0, "output_type":"max",

```



```

"radius":3.0, "type": "writers.gdal" ""

```

```

"" "filename":"burgOutput1.tif",

```



```

"resolution":0.1, "output_type":"max", "radius":0.15, "type": "writers.gdal" ""

```

```

"filename":"burgOutput3.tif", "resolution":0.05, "output_type":"max", "radius":0.05, "type": "writers.gdal" ""

```

Skulptur Test GeoSLAM

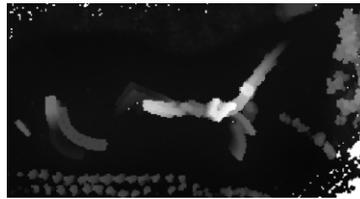


``` "filename": "burgOutput3.tif", "resolution": 0.05, "output\_type": "max",



"radius": 0.05, "type": "writers.gdal" ```

``` "filename": "burgOutput31.tif",



"resolution": 0.05, "output_type": "max", "radius": 0.1, ```

"filename": "skulpturOutput1.tif", "resolution": 0.1, "output_type": "max", "radius": 0.15, "type": "writers.gdal" ```



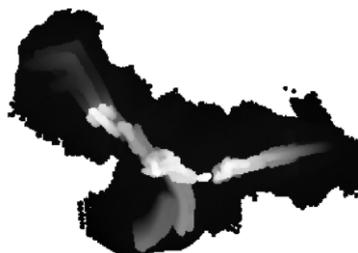
``` "filename": "skulpturOutput.tif", "resolution": 1.0, "output\_type": "max",

"radius": 3.0, "type": "writers.gdal" ```

## Test IPAD (Auswahl nicht alle!)

Einstellungen immer:

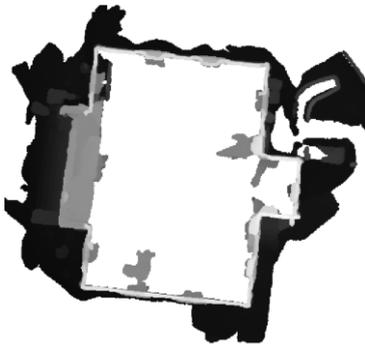
```
"resolution": 0.05,
"output_type": "max",
"radius": 0.1,
"type": "writers.gdal"
```



Sitescape:

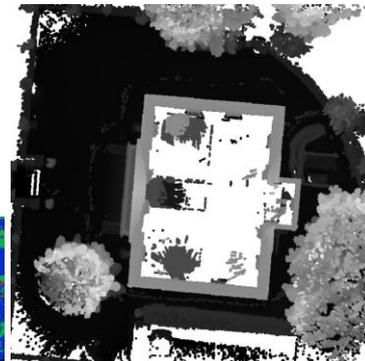
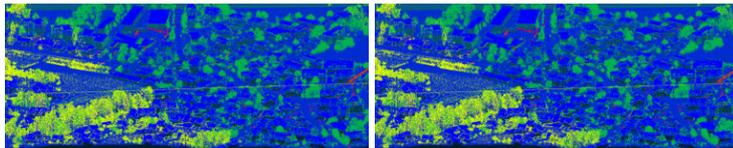


Scanniverse:



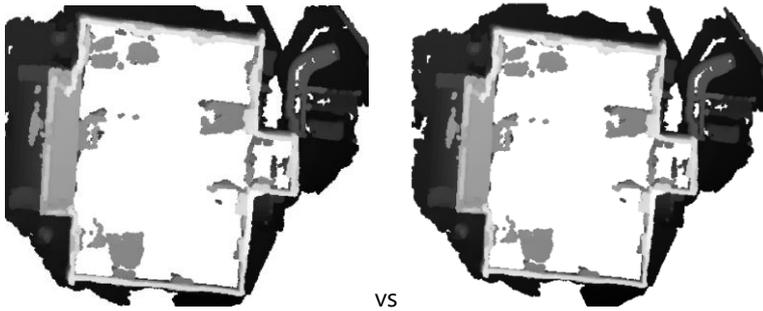
Polycam:

### Test BLK2GO

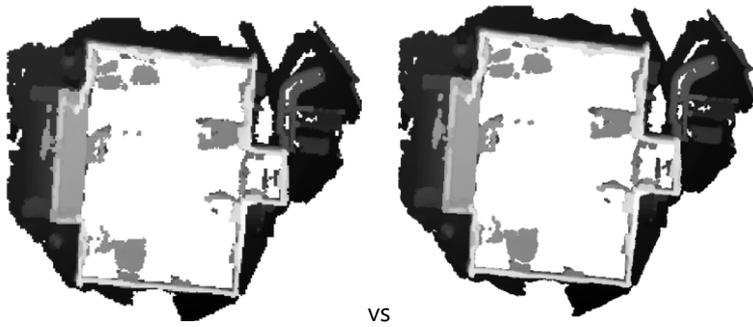


optimierungen:

```
{
 "type": "filters.outlier",
 "method": "statistical",
 "mean_k": 12,
 "multiplier": 2.0
}
VS
{
 "type": "filters.outlier",
 "method": "statistical",
 "mean_k": 12,
 "multiplier": 1.0 <-Default=2
},
```



```
{
 "filename": "villa3dScannerApp111.tif",
 "resolution": 0.1,
 "output_type": "max",
 "radius": 0.1,
 "type": "writers.gdal"
}
VS
"resolution": 0.05,
"output_type": "max",
"radius": 0.1,
"type": "writers.gdal"
```



## **4 Qualitätsmanagement**

### **4.1 Qualitätsmassnahmen**

# Qualitätsmassnahmen

| Massnahme                   | Zeitraum                                                                                                                    | Ziel                                                 | Beschreibung                                                                                                                                                                                                              |
|-----------------------------|-----------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Unit Tests                  | Gesamte Konstruktionsphase                                                                                                  | 80 % Code Coverage                                   | Die einzelnen Komponenten der Software werden getestet.                                                                                                                                                                   |
| Integration Tests           | Ab Meilenstein "Prototyp"                                                                                                   | Einzelne Komponenten arbeiten fehlerfrei miteinander | Jede Möglichkeit mindestens einmal getestet                                                                                                                                                                               |
| Usability Tests             | Einer auf Papier, um Navigationsschwierigkeiten zu erkennen (Phase End of Elaboration) und einer bei "Alpha-Version fertig" | Applikation ist User Friendly                        | Unabhängige Personen testen die App                                                                                                                                                                                       |
| System Tests                | Ab Meilenstein "Prototyp"                                                                                                   | Funktionsumfang testen                               | Mit den System-Tests wird sichergestellt, dass alle Functional Requirements erfüllt werden. Als Grundlage für die Tests dienen die formulierten Use-Cases. Die System-Tests werden von einem Tester manuell durchgeführt. |
| Stabilität-Test             | Ab Alpha Version                                                                                                            | Zuverlässigkeit der Applikation prüfen               | Dabei werden Extrembelastungen gemäss den Requirements geprüft                                                                                                                                                            |
| Kontinuierliche Integration | Ganzes Projekt                                                                                                              | Kein grosser Integrationsaufwand                     | Wenn am Projekt gearbeitet wird, wird jeden Tag einmal (sofern sinnvoll) ein Merge Request gemacht                                                                                                                        |
| Offene Kommunikation        | Gesamtes Projekt                                                                                                            | Problembhebung einfach und schnell erledigen         | Schnelle und ehrliche Kommunikation                                                                                                                                                                                       |
| Code Reviews                | Gesamtes Projekt                                                                                                            | Steigerung der Qualität unserer Arbeiten             | Code wird bei jedem Merge durch den Anderen reviewed                                                                                                                                                                      |

| <b>Massnahme</b>      | <b>Zeitraum</b>            | <b>Ziel</b>                              | <b>Beschreibung</b>                                                                                                                             |
|-----------------------|----------------------------|------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------|
| Dokumentationsreview  | Gesamtes Projekt           | Steigerung der Qualität unserer Arbeiten | Dokumentation wird bei jedem Merge durch den Anderen reviewed                                                                                   |
| Code Style Guidelines | Gesamte Konstruktionsphase | Verbesserung der Code Qualität           | Die Code Style Guidelines werden mit jedem Pipeline-Durchlauf geprüft. Dadurch wird die Qualität und die Verständlichkeit des Codes verbessert. |

## Definition of Done

### Source

1. Der Code Läuft ohne Errors und Warnings
2. Unit Tests wurden für den Code geschrieben
3. Unit Tests sind durchgelaufen
4. Definierte Code Guidelines und Standards wurden eingehalten
5. Code dokumentiert
6. Code wurde reviewed oder im Pair Programming erstellt
7. Nicht funktionale Anforderungen erfüllt
8. Integriert in den Master Branch

### Documentation

1. Rechtschreibprüfung erfolgreich
2. Inhalt auf Richtigkeit geprüft
3. Punkte 1-2 wurden von einem weiteren Teammitglied reviewed
4. Integriert in den Master Branch

## 4.2 Systemtests

# Prototyp 27.04.2021

| Was                                                                                                                                                                                                                                                                                                                                            | Implementiert | Kommentar                                              | Verbesserung |
|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------|--------------------------------------------------------|--------------|
| <b>1. Registrieren:</b> Der Gast registriert sich und wird somit zu einem registrierten Benutzer.                                                                                                                                                                                                                                              | Nein          | ---                                                    | ---          |
| <b>2. Einloggen:</b> Der Gast loggt sich ein.                                                                                                                                                                                                                                                                                                  | Nein          | ---                                                    | ---          |
| <b>3. Punktwolke visualisieren:</b> Der Benutzer kann die eigenen oder die ihm freigegebenen Punktwolken visualisieren. Die Punkte der gewünschten Punktwolke werden dargestellt und der Benutzer kann die Darstellung anpassen. Die Visualisierung lässt sich rotieren, verschieben und drehen. Ausserdem kann man dabei Messungen vornehmen. | Nein          | Implementation läuft noch nicht, ist aber Vorbereitet. | ---          |
| <b>4. Punktwolken-Übersicht einsehen:</b> Der Benutzer kann in einer Übersicht die für ihn verfügbaren Punktwolken ansehen.                                                                                                                                                                                                                    | Nein          | ---                                                    | ---          |
| <b>5. Punktwolke in 2D visualisieren:</b> Der Benutzer kann die für ihn verfügbaren Punktwolken in 2D visualisieren.                                                                                                                                                                                                                           | Nein          | ---                                                    | ---          |
| <b>6. Punktwolke in 2D exportieren:</b> Der Benutzer kann die für ihn verfügbaren Punktwolken in 2D exportieren.                                                                                                                                                                                                                               | Nein          | ---                                                    | ---          |
| <b>7. Punktwolke herunterladen:</b> Der Benutzer kann die für ihn verfügbaren Punktwolken herunterladen.                                                                                                                                                                                                                                       | Nein          | ---                                                    | ---          |
| <b>8. Visualisierung teilen:</b> Der Benutzer kann einen Link zu seiner Visualisierung mit der gewünschten Darstellung generieren und diesen mit anderen Benutzern teilen. Dabei muss natürlich die Berechtigung für die entsprechende Punktwolke vorhanden sein.                                                                              | Nein          | ---                                                    | ---          |
| <b>9. Punktwolke hochladen:</b> Der registrierte und eingeloggte Benutzer kann Punktwolken in verschiedenen Dateiformaten hochladen.                                                                                                                                                                                                           | Nein          | ---                                                    | ---          |

| Was                                                                                                                                                                                                              | Implementiert | Kommentar | Verbesserung |
|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------|-----------|--------------|
| <b>10. Punktwolke freigeben:</b> Der registrierte und eingeloggte Benutzer kann Punktwolken anderen registrierten Benutzern freigeben oder die Punktwolke öffentlich und somit auch für Gäste zugänglich machen. | Nein          | ---       | ---          |
| <b>11. Punktwolke löschen:</b> Der registrierte und eingeloggte Benutzer kann seine eigenen Punktwolken löschen.                                                                                                 | Nein          | ---       | ---          |
| <b>12. Punktwolke bereinigen:</b> Der registrierte und eingeloggte Benutzer kann seine eigenen Punktwolken extern bereinigen, in dem er ungewollte Objekte in der Punktwolke entfernt.                           | Nein          | ---       | ---          |

## Alpha 07.05.2021

| Was                                                                                                                                                                                                                                                                                                                                            | Implementiert | Kommentar                                                  | Verbesserung                                                   |
|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------|------------------------------------------------------------|----------------------------------------------------------------|
| <b>1. Registrieren:</b> Der Gast registriert sich und wird somit zu einem registrierten Benutzer.                                                                                                                                                                                                                                              | Nein          | Implementation vorbereitet, aber noch keine Funktionalität | ---                                                            |
| <b>2. Einloggen:</b> Der Gast loggt sich ein.                                                                                                                                                                                                                                                                                                  | Nein          | Implementation vorbereitet, aber noch keine Funktionalität | ---                                                            |
| <b>3. Punktwolke visualisieren:</b> Der Benutzer kann die eigenen oder die ihm freigegebenen Punktwolken visualisieren. Die Punkte der gewünschten Punktwolke werden dargestellt und der Benutzer kann die Darstellung anpassen. Die Visualisierung lässt sich rotieren, verschieben und drehen. Ausserdem kann man dabei Messungen vornehmen. | Ja            | Defaultpunktwolke kann angezeigt und angepasst werden.     | Verschiedene Punktwolken darstellen und Potree Menu entfernen. |
| <b>4. Punktwolken-Übersicht einsehen:</b> Der Benutzer kann in einer Übersicht die für ihn verfügbaren Punktwolken ansehen.                                                                                                                                                                                                                    | Nein          | ---                                                        | ---                                                            |
| <b>5. Punktwolke in 2D visualisieren:</b> Der Benutzer kann die für ihn verfügbaren Punktwolken in 2D visualisieren.                                                                                                                                                                                                                           | Nein          | ---                                                        | ---                                                            |

| Was                                                                                                                                                                                                                                                               | Implementiert | Kommentar | Verbesserung |
|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------|-----------|--------------|
| <b>6. Punktwolke in 2D exportieren:</b> Der Benutzer kann die für ihn verfügbaren Punktwolken in 2D exportieren.                                                                                                                                                  | Nein          | ---       | ---          |
| <b>7. Punktwolke herunterladen:</b> Der Benutzer kann die für ihn verfügbaren Punktwolken herunterladen.                                                                                                                                                          | Nein          | ---       | ---          |
| <b>8. Visualisierung teilen:</b> Der Benutzer kann einen Link zu seiner Visualisierung mit der gewünschten Darstellung generieren und diesen mit anderen Benutzern teilen. Dabei muss natürlich die Berechtigung für die entsprechende Punktwolke vorhanden sein. | Nein          | ---       | ---          |
| <b>9. Punktwolke hochladen:</b> Der registrierte und eingeloggte Benutzer kann Punktwolken in verschiedenen Dateiformaten hochladen.                                                                                                                              | Nein          | ---       | ---          |
| <b>10. Punktwolke freigeben:</b> Der registrierte und eingeloggte Benutzer kann Punktwolken anderen registrierten Benutzern freigeben oder die Punktwolke öffentlich und somit auch für Gäste zugänglich machen.                                                  | Nein          | ---       | ---          |
| <b>11. Punktwolke löschen:</b> Der registrierte und eingeloggte Benutzer kann seine eigenen Punktwolken löschen.                                                                                                                                                  | Nein          | ---       | ---          |
| <b>12. Punktwolke bereinigen:</b> Der registrierte und eingeloggte Benutzer kann seine eigenen Punktwolken extern bereinigen, in dem er ungewollte Objekte in der Punktwolke entfernt.                                                                            | Nein          | ---       | ---          |

## Beta 21.05.2021

| Was                                                                                               | Implementiert | Kommentar | Verbesserung        |
|---------------------------------------------------------------------------------------------------|---------------|-----------|---------------------|
| <b>1. Registrieren:</b> Der Gast registriert sich und wird somit zu einem registrierten Benutzer. | Ja            | ---       | Passwort verlängern |
| <b>2. Einloggen:</b> Der Gast loggt sich ein.                                                     | Ja            | ---       | ---                 |

| Was                                                                                                                                                                                                                                                                                                                                            | Implementiert | Kommentar                                                   | Verbesserung                                                                              |
|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------|-------------------------------------------------------------|-------------------------------------------------------------------------------------------|
| <b>3. Punktwolke visualisieren:</b> Der Benutzer kann die eigenen oder die ihm freigegebenen Punktwolken visualisieren. Die Punkte der gewünschten Punktwolke werden dargestellt und der Benutzer kann die Darstellung anpassen. Die Visualisierung lässt sich rotieren, verschieben und drehen. Ausserdem kann man dabei Messungen vornehmen. | Ja            | Defaultpunktwolke kann angezeigt und angepasst werden.      | Verschiedene Punktwolken darstellen.                                                      |
| <b>4. Punktwolken-Übersicht einsehen:</b> Der Benutzer kann in einer Übersicht die für ihn verfügbaren Punktwolken ansehen.                                                                                                                                                                                                                    | Nein          | Vorbereitet aber noch nicht funktionsfähig                  | ---                                                                                       |
| <b>5. Punktwolke in 2D visualisieren:</b> Der Benutzer kann die für ihn verfügbaren Punktwolken in 2D visualisieren.                                                                                                                                                                                                                           | Nein          | ---                                                         | ---                                                                                       |
| <b>6. Punktwolke in 2D exportieren:</b> Der Benutzer kann die für ihn verfügbaren Punktwolken in 2D exportieren.                                                                                                                                                                                                                               | Nein          | ---                                                         | ---                                                                                       |
| <b>7. Punktwolke herunterladen:</b> Der Benutzer kann die für ihn verfügbaren Punktwolken herunterladen.                                                                                                                                                                                                                                       | Nein          | ---                                                         | ---                                                                                       |
| <b>8. Visualisierung teilen:</b> Der Benutzer kann einen Link zu seiner Visualisierung mit der gewünschten Darstellung generieren und diesen mit anderen Benutzern teilen. Dabei muss natürlich die Berechtigung für die entsprechende Punktwolke vorhanden sein.                                                                              | Nein          | ---                                                         | ---                                                                                       |
| <b>9. Punktwolke hochladen:</b> Der registrierte und eingeloggte Benutzer kann Punktwolken in verschiedenen Dateiformaten hochladen.                                                                                                                                                                                                           | Minimal       | Funktion in Overview eingefügt und unterstützt nur LAS, LAZ | Funktion in Formular übertragen, Errorhandling einfügen und mehrere Formate unterstützen. |
| <b>10. Punktwolke freigeben:</b> Der registrierte und eingeloggte Benutzer kann Punktwolken anderen registrierten Benutzern freigeben oder die Punktwolke öffentlich und somit auch für Gäste zugänglich machen.                                                                                                                               | Ja            | Kann beim Upload angegeben werden.                          | ---                                                                                       |

| Was                                                                                                                                                                                    | Implementiert | Kommentar | Verbesserung |
|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------|-----------|--------------|
| <b>11. Punktwolke löschen:</b> Der registrierte und eingeloggte Benutzer kann seine eigenen Punktwolken löschen.                                                                       | Nein          | ---       | ---          |
| <b>12. Punktwolke bereinigen:</b> Der registrierte und eingeloggte Benutzer kann seine eigenen Punktwolken extern bereinigen, in dem er ungewollte Objekte in der Punktwolke entfernt. | Nein          | ---       | ---          |

## Final 04.06.2021

| Was                                                                                                                                                                                                                                                                                                                                            | Implementiert | Kommentar                               | Verbesserung |
|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------|-----------------------------------------|--------------|
| <b>1. Registrieren:</b> Der Gast registriert sich und wird somit zu einem registrierten Benutzer.                                                                                                                                                                                                                                              | Ja            | ---                                     | ---          |
| <b>2. Einloggen:</b> Der Gast loggt sich ein.                                                                                                                                                                                                                                                                                                  | Ja            | ---                                     | ---          |
| <b>3. Punktwolke visualisieren:</b> Der Benutzer kann die eigenen oder die ihm freigegebenen Punktwolken visualisieren. Die Punkte der gewünschten Punktwolke werden dargestellt und der Benutzer kann die Darstellung anpassen. Die Visualisierung lässt sich rotieren, verschieben und drehen. Ausserdem kann man dabei Messungen vornehmen. | Ja            | ---                                     | ---          |
| <b>4. Punktwolken-Übersicht einsehen:</b> Der Benutzer kann in einer Übersicht die für ihn verfügbaren Punktwolken ansehen.                                                                                                                                                                                                                    | Ja            | ---                                     | ---          |
| <b>5. Punktwolke in 2D visualisieren:</b> Der Benutzer kann die für ihn verfügbaren Punktwolken in 2D visualisieren.                                                                                                                                                                                                                           | Ja            | ---                                     | ---          |
| <b>6. Punktwolke in 2D exportieren:</b> Der Benutzer kann die für ihn verfügbaren Punktwolken in 2D exportieren.                                                                                                                                                                                                                               | Ja            | ---                                     | ---          |
| <b>7. Punktwolke herunterladen:</b> Der Benutzer kann die für ihn verfügbaren Punktwolken herunterladen.                                                                                                                                                                                                                                       | Ja            | ---                                     | ---          |
| <b>8. Visualisierung teilen:</b> Der Benutzer kann einen Link zu seiner Visualisierung mit der gewünschten Darstellung generieren und diesen mit anderen Benutzern teilen. Dabei muss natürlich die Berechtigung für die entsprechende Punktwolke vorhanden sein.                                                                              | Nein          | Wure aus zeitlichen Gründen weggelassen | ---          |

| Was                                                                                                                                                                                                              | Implementiert | Kommentar          | Verbesserung |
|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------|--------------------|--------------|
| <b>9. Punktwolke hochladen:</b> Der registrierte und eingeloggte Benutzer kann Punktwolken in verschiedenen Dateiformaten hochladen.                                                                             | Ja            | e57, las, laz, ply | ---          |
| <b>10. Punktwolke freigeben:</b> Der registrierte und eingeloggte Benutzer kann Punktwolken anderen registrierten Benutzern freigeben oder die Punktwolke öffentlich und somit auch für Gäste zugänglich machen. | Ja            | ---                | ---          |
| <b>11. Punktwolke löschen:</b> Der registrierte und eingeloggte Benutzer kann seine eigenen Punktwolken löschen.                                                                                                 | Ja            | ---                | ---          |
| <b>12. Punktwolke bereinigen:</b> Der registrierte und eingeloggte Benutzer kann seine eigenen Punktwolken extern bereinigen, in dem er ungewollte Objekte in der Punktwolke entfernt.                           | Ja            | ---                | ---          |

### **4.3 Usability Test**

# Usability Test

---

## Testkonzept

Das [Figma Projekt](#) wird einem Usability-Test unterzogen. Dabei werden wir das [Wireframe](#) ausgewählten Testpersonen vorlegen, welche das UI und die Funktionalität kritisch beurteilen. Die Rückmeldungen werden wir für die Erarbeitung der Webseite verwenden.

## Test Ziele

- Ist der Aufbau verständlich?
- Ist die Navigation verständlich?
- Sind die wichtigsten Funktionalitäten abgedeckt?

## Testsznarien

| Nr | Kurzbeschrieb         | Szenario                                                                                                                        | Fragestellung (Hypothese)                                                                      |
|----|-----------------------|---------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------|
| 1  | Punktwolke ansehen    | Sie interessieren sich für die Grabung in Neubühl und möchten diese Punktwolke im grösstmöglichen Bildschirm sehen.             | Merkt er, dass er auf der Grabung Neubühl ist und die Seiten einklappen kann?                  |
| 2  | Punktwolke darstellen | Sie möchten die Punktdichte erhöhen, um mehr Details zu erkennen.                                                               | Findet der Tester die Einstellungen?                                                           |
| 3  | Darstellung teilen    | Sie möchten diese Darstellung an ihren Kollegen senden.                                                                         | Findet der Tester den teilen-Knopf und versteht er, was geteilt wird?                          |
| 4  | Punktwolke hochladen  | Sie haben selber eine schöne Aufnahme ihres Quartiers erstellt und möchten diese hochladen.                                     | Versteht, dass er sich dazu einloggen muss? Findet er den Upload-Knopf? Verständliche Angaben? |
| 5  | Punktwolke bearbeiten | Sie bemerken, dass Ihnen die Beschreibung nicht passt und wollen sie verbessern.                                                | Bearbeiten gefunden?                                                                           |
| 6  | 2D-Ansicht            | Sie möchten sehen, was für 2D-Strukturen in ihren Daten ableitbar sind, um diese anschliessend in einen 2D-Plan zu integrieren. | Wechsel zu 2D-Ansicht gefunden?                                                                |
| 7  | 2D-Ansicht downloaden | Nun möchten sie die Ansicht exportieren, um sie in QGIS zu importieren.                                                         | Findet Download                                                                                |
| 8  | 3D-Download           | Zum Abschluss möchten sie die Grabung Neubühl als Punktwolke exportieren.                                                       | Findet den LAZ-Datei Download und versteht es?                                                 |

## Testvorbereitungen

[Figma Projekt Präsentationsmodus](#)

## Auswertung Usability Test 1 (Figma)

### Protokolle

- [Protokoll 1](#)
- [Protokoll 2](#)

### Beibehalten

- Aufteilung der Seite
- Übersichtlichkeit
- Farbige Ecken mit der Freigabe

### Verbessern

| <b>Problem</b>                          | <b>Änderungsvorschlag Testpersonen</b>                                   | <b>Änderungsvorschlag Team</b>                                                                                                                      | <b>Umsetzung</b>                                                      |
|-----------------------------------------|--------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------|
| Rechte Seite wegklicken                 | Auch einen Pfeil über den Reitern                                        | -                                                                                                                                                   | Möglich, aber nicht zwingend                                          |
| Upload Funktionalität sichtbarer machen | -                                                                        | Button auch ausgegraut für Gast sichtbar machen und bei Klick aufs Login weiterleiten.                                                              | Wird umgesetzt                                                        |
| Download Punktwolken und Raster         | -                                                                        | Klar beschreiben, dass dies die Rohdaten sind und keine Darstellungsoptionen, die für den Download verwendet werden. Sonst Teilen-Option verwenden. | wird umgesetzt                                                        |
| Upload Button                           | Nicht in Header, sondern links als erster Listeneintrag.                 | -                                                                                                                                                   | Möglich, aber nicht zwingend                                          |
| Freigabe                                | Farbige Ecken cool, weitere Farben für andere Freigabe stufen erwünscht. | -                                                                                                                                                   | Wird umgesetzt                                                        |
| Mehrere gleiche Scans                   | Qualitätsraiting, wenn mehrere gleiche Scanobjekte vorhanden sind.       | -                                                                                                                                                   | Wird nicht umgesetzt, da in der Vision ein Merge gemacht werden soll. |
| Knöpfe                                  | Die Buttons schöner gestalten (nicht schwarz beschriften).               | -                                                                                                                                                   | Wird umgesetzt                                                        |

| <b>Problem</b> | <b>Änderungsvorschlag<br/>Testpersonen</b>                                  | <b>Änderungsvorschlag Team</b> | <b>Umsetzung</b>                                                                         |
|----------------|-----------------------------------------------------------------------------|--------------------------------|------------------------------------------------------------------------------------------|
| Überladung     | Rechte Seite besser eingeklappt bei Start.                                  | -                              | Wird offen gelassen und die Überladung akzeptiert. So sieht man direkt, was möglich ist. |
| Ausdehnung     | Grösse nennen oder weglassen. Beispiel um zu verstehen was das sein sollte. | -                              | Wird weggelassen.                                                                        |

## Auswertung Usability Test 2 (Final Version)

### Protokolle

- [Protokoll 1](#)
- [Protokoll 2](#)

### Unbedingt beibehalten

### Verbesserungen

| <b>Problem</b>                       | <b>Änderungsvorschlag<br/>Testperson</b>                                           | <b>Änderungsvorschlag<br/>Team</b>                                                                                          | <b>Umsetzung</b>                                                                                                                                      |
|--------------------------------------|------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------|
| Springen nach Edit                   | Mindestens Aufklappen der Liste, um sichtbar zu machen, wo man hin gesprungen ist. | Nicht springen bei Edit oder Listen-Sortierung so anpassen, dass man nach dem Refresh automatisch auf dem Richtigen landet. | Wird aus zeitlichen Gründen nicht mehr umgesetzt, sollte aber mit der Weiterentwicklung angegangen werden.                                            |
| Sichtbarkeit der eigenen Punktwolken | -                                                                                  | Suchen/ filtern nach eigenen (Zurzeit kommen auch die, die mir freigegeben wurden, mit der Eingabe meines Benutzernamen)    | Dies ist Funktionalität, die es bereits gibt und in diesem Umfang wird nicht weiter daran gearbeitet.                                                 |
| Edit-Button sichtbarer machen        | -                                                                                  | Auffälligere Darstellung                                                                                                    | Wird nicht umgesetzt, da nur eine der Testpersonen zuerst in den Potree Einstellungen gesucht hat und die Knöpfe bereits dominant und zu oberst sind. |

| <b>Problem</b>             | <b>Änderungsvorschlag<br/>Testperson</b>                                                                  | <b>Änderungsvorschlag<br/>Team</b>              | <b>Umsetzung</b>                                                                                                                                                               |
|----------------------------|-----------------------------------------------------------------------------------------------------------|-------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Scrolling                  | -                                                                                                         | Scroll stoppen, bevor man nicht mehr weg kommt. | Wird nicht mehr umgesetzt, da das Scrolling von Potree kommt. Ein Merge-Request wird vielleicht resultieren, wenn dies nicht bereits mit einer Variable definiert werden kann. |
| Umstellung<br>Point Budget | -                                                                                                         | Auf die Punktwolke abgestimmter Slider.         | Wird nicht mehr umgesetzt, ist aber eine tolle Weiterentwicklung.                                                                                                              |
| TIFF<br>Informationen      | Beschreiben, dass das TIFF nicht im Fotoviewer dargestellt werden kann, sondern nur in einem GIS Programm | -                                               | Wird umgesetzt.                                                                                                                                                                |

# Usability Test 1

---

## Einleitung

Sie können keine Fehler machen. Alles, was Sie nicht können, machen auch andere falsch und muss eliminiert werden, bevor die Webseite benutzt werden kann. Die Aufzeichnungen werden am Ende des Projekts gelöscht. Das schriftliche Protokoll und die Auswertung über alle Tests werden abgelegt und stehen dem Projektteam sowie dem Betreuer zu Verfügung. Wenn Sie zustimmen, würden wir neben der Gesamtauswertung auch das anonymisierte Protokoll als Anhang in die Dokumentation einfügen.

Der Test dient der Verbesserung des Punktwolken-Browsers.

1. Sind Sie damit einverstanden?

- Ja

2. Auch mit dem Anhängen des Protokolls?

- Ja

## Vorgehen

Zu Beginn werde ich Ihnen einige Fragen zu Ihren Vorkenntnissen stellen. Anschliessend geht es in den eigentlichen Test über. Ich werde Ihnen einige Szenarien vorlesen und Sie dürfen gerne jede ihrer Überlegungen laut äussern, während Sie dies umsetzen. Ihre mündlichen Inputs helfen uns, um zusehen, ob unsere Ideen auch bei den Benutzern ankommen.

Haben Sie noch fragen?

- Nein
- Wie schätzen Sie ihre Vorkenntnisse mit GIS-Programmen ein?
- Sehr gut, verwende ich häufig. Welche:
- Habe ich schon mal gebraucht. Welche:
- Kenne ich, habe aber noch nie eines eingesetzt.
- Kenne ich nicht. Bemerkung: Nur vom Hören Sagen selber keinen weiteren Bezug.
- Wissen Sie, was eine Punktwolke ist?
- Ja, sehr genau.
- Ja in etwa.
- Nein. Bemerkungen: Brauchte keine weiterführenden Informationen dazu, um das Tool zu verstehen.
- Benutzen sie online Kartendienste (Google, OSM, ...)?

- Ja sehr häufig. Welche: *Wenn ich nicht weiss wohin oder von wo die Rede ist, dann schaue ich immer auf Google Maps nach*
- Ja manchmal. Welche:
- Nie Bemerkungen:

#### 1 | Punktwolke ansehen:

- Dauer bis ausgeführt: Mittel
- Merkt dass dies bereits offen ist.
- Inhalte gelesen
- Seiten eingeklappt
- Angaben haben gefehlt:
- Übersicht unordentlich
- Weitere Bemerkungen: Links öffentlich gelesen, Grabung Neubühl gerne Umrandung als angewält wahrgenommen, rechts nicht bemerkt, dass man dies wegeklicken kann, würde erwarten, dass man im Tool dann zoomen und rotieren kann (auf Wireframe nicht abbildbar), hätte Regler "Field of View" auch noch ausprobiert (auf Wireframe nichts verändert).

#### 2 | Punktwolke darstellen:

- Dauer bis ausgeführt: Schnell
- Funktion ohne Eingriff gefunden
- Zuerst Links gesucht
- Zuerst in Header gesucht
- Weitere Bemerkungen: Regler sofort rechts gefunden.

#### 3 | Darstellung teilen:

- Dauer bis ausgeführt: Schnell
- Funktion ohne Eingriff gefunden
- Verwirrt
- Klar was und wie geteilt wird
- Symbol nicht erkannt
- Weitere Bemerkungen: Wenn nicht da, dann Downloadknopf testen und so senden -> (Interpretation: Nicht allen klar, dass der Download LAZ Rohdaten ohne Einstellungen sind)

#### 4 | Punktwolke hochladen:

- Dauer bis ausgeführt: Mittel
- Funktion ohne Eingriff gefunden
- Login ohne Eingriff versucht
- Unklare Attribute: Erst nach Hinweis für Login gefunden, Datumsformat vorgegeben?, Ausdehnung unklar und nach Erklärung den Sinn nicht eingesehen, hätte wenn dann lieber Grösse anstatt nicht Ausdehnung.
- Weitere Bemerkungen: War nicht bewusst, dass sie nicht eingeloggt war. Aber nachher klar, sei ja bei vielen Tools so. Freigabemöglichkeiten selbständig richtig interpretiert.

#### 5 | Punktwolke bearbeiten:

- Dauer bis ausgeführt: Schnell
- Funktion ohne Eingriff gefunden
- Unklare Attribute:
- weitere Bemerkungen:

#### 6 | 2D-Ansicht:

- Dauer bis ausgeführt: Schnell
- Ansicht ohne Eingriff gefunden
- Ansicht verstanden
- Unklar was dies sein sollte?
- Inhalte gelesen
- Weitere Bemerkungen: Hat gefragt, ob man da nun die Umrisse erkennen sollte (Wireframe Bild ist auch noch schwer zu erkennen, hoffen noch auf bessere Ableitungen).

#### 7 | 2D-Ansicht downloaden:

- Dauer bis ausgeführt: Schnell
- Ansicht ohne Eingriff gefunden
- Knopfaufschrift unklar
- Weitere Bemerkungen: Transparenz einstellen, dann download. Möchte den bearbeiten-Knopf auch da.

#### 8 | 3D-Download:

- Dauer bis ausgeführt: Schnell
- Liste richtige Punktwolke gefunden
- Funktion ohne Eingriff gefunden
- Beschriftung unklar
- Weitere Bemerkungen: Vermutete, dass LAZ ein Punktwolkenformat ist, ist sich aber nicht sicher.

#### Wireframe verbessern

\*Dauer: schnell, angemessen, langsam -> langsam kann ein Hinweis auf eine Unverständlichkeit sein, die verbessert werden muss. Sehr übersichtlich links rechts wechseln nein.

- Wie sind Sie mit der Navigation zurecht gekommen?
- ++
- +
- /
- -
- - - Bemerkung: Sehr übersichtlich, gute Trennung von Auswahl (links) und Einstellungen der Punktwolke (rechts). Upload links bei der Liste erwartet, z.B. als erster Eintrag. Pfeil zum zuklappen in 3D und 2D separat. Suchen nach Ort und Datum würde wahrscheinlich am meisten verwendet.
- Waren die Angaben in den Previews für Sie aufschlussreich genug?
- ++

- +
- -
- - - Bemerkungen: Nicht zu viel, berechtigte Angaben beim Hochladen und nicht alles muss man angeben. Zum Suchen genug Informationen in den Kacheln.
- Wie gefällt Ihnen das Design? (Achtung Geschmacksfrage)
- ++
- +
- /
- -
- - - Bemerkungen: Gut, Knöpfe etwas schöner und nicht schwarz beschriftet. Farbige Ecken sind cool und unterschiedliche Farben je nach Typ noch informativer, vielleicht noch Qualitätsrating, wenn mehrere gleiche Scanobjekte vorhanden sind.
- Was war besonders gut: Alles besprochen
- Was besonders schlecht: Alles besprochen
- Probleme aus dem Test aufgreifen: Alles besprochen

# Test

## Einleitung

Sie können keine Fehler machen. Alles, was Sie nicht können, machen auch andere falsch und muss eliminiert werden, bevor die Webseite benutzt werden kann. Die Aufzeichnungen werden am Ende des Projekts gelöscht. Das schriftliche Protokoll und die Auswertung über alle Tests werden abgelegt und stehen dem Projektteam sowie dem Betreuer zu Verfügung. Wenn Sie zustimmen, würden wir neben der Gesamtauswertung auch das anonymisierte Protokoll als Anhang in die Dokumentation einfügen.

Der Test dient der Verbesserung des Punktwolken-Browsers.

1. Sind Sie damit einverstanden?

- Ja

2. Auch mit dem Anhängen des Protokolls?

- Ja

## Vorgehen

Zu Beginn werde ich Ihnen einige Fragen zu Ihren Vorkenntnissen stellen. Anschliessend geht es in den eigentlichen Test über. Ich werde Ihnen einige Szenarien vorlesen und Sie dürfen gerne jede ihrer Überlegungen laut äussern, während Sie dies umsetzen. Ihre mündlichen Inputs helfen uns, um zusehen, ob unsere Ideen auch bei den Benutzern ankommen.

Haben Sie noch fragen?

- Nein
- Wie schätzen Sie ihre Vorkenntnisse mit GIS-Programmen ein?
  - Sehr gut, verwende ich häufig. Welche:
  - Habe ich schon mal gebraucht. Welche: QGIS
  - Kenne ich, habe aber noch nie eines eingesetzt.
  - Kenne ich nicht. Bemerkung:
- Wissen Sie, was eine Punktwolke ist?
  - Ja, sehr genau.
  - Ja in etwa.
  - Nein. Bemerkungen:
- Benutzen sie online Kartendienste (Google, OSM, ...)?
  - Ja sehr häufig. Welche: Leaflet beim Arbeiten, OSM beim Arbeiten, Google um Gegend zu erkunden.

- Ja manchmal. Welche:
- Nie Bemerkungen:

#### 1 | Punktwolke ansehen:

- Dauer bis ausgeführt: Mittel
- Merkt dass dies bereits offen ist.
- Inhalte gelesen
- Seiten eingeklappt
- Angaben haben gefehlt:
- Übersicht unordentlich
- weitere Bemerkungen: Klickt mehrfach und es ändert nichts, nachher bemerkt, dass es bereits angewählt ist. Umrandung nicht verstanden, wirkt überladen (alles offen beim Start).

#### 1 | Punktwolke darstellen:

- Dauer bis ausgeführt: Schnell
- Funktion ohne Eingriff gefunden
- Zuerst Links gesucht
- Zuerst in Header gesucht
- Weitere Bemerkungen: Ohne probleme. Durch das Probieren des ersten Schrittes erkannt, dass die Einstellungen rechts sind und die Auswahl links.

#### 3 | Darstellung teilen:

- Dauer bis ausgeführt: Schnell
- Funktion ohne Eingriff gefunden
- Verwirrt
- Klar was und wie geteilt wird
- Symbol nicht erkannt
- Weitere Bemerkungen: Share oder Download je nachdem.

#### 4 | Punktwolke hochladen:

- Dauer bis ausgeführt: Schnell
- Funktion ohne Eingriff gefunden
- Login ohne Eingriff versucht
- Unklare Attribute: Ausdehnung
- Weitere Bemerkungen: Upload hinterlegen sonst kommt man nicht auf die Idee, ein Login zu machen, da man dies meist nicht möchte.

#### 5 | Punktwolke bearbeiten:

- Dauer bis ausgeführt: Schnell
- Funktion ohne Eingriff gefunden
- Unklare Attribute: Ausdehnung
- weitere Bemerkungen: Titel klar, Genauigkeit des Ortes einfach Textfeld, Beschreibung gut um zu erklären für was oder warum die Aufnahme gemacht wurde, Freigabe Klar, Bereinigen kleiner machen,

Performance verbessern, Ausdehnung Abstand zwischen den Sachen oder Skalierung der Wolke anpassen

#### 6 | 2D-Ansicht:

- Dauer bis ausgeführt: Schnell
- Ansicht ohne Eingriff gefunden
- Ansicht verstanden
- Unklar was dies sein sollte?
- Inhalte gelesen
- Weitere Bemerkungen: 2D klicken und dann hat man eine 2D-Darstellung

#### 7 | 2D-Ansicht downloaden:

- Dauer bis ausgeführt: Schnell
- Ansicht ohne Eingriff gefunden
- Knopfaufschrift unklar
- Weitere Bemerkungen: Herunterladen im tif Format

#### 8 | 3D-Download:

- Dauer bis ausgeführt: Schnell
- Liste richtige Punktwolke gefunden
- Funktion ohne Eingriff gefunden
- Beschriftung unklar
- Weitere Bemerkungen: Download LAZ würde ich jetzt als Export verwenden, oder gibt es zu Export noch eine Funktion mit anderen Formaten?

\*Dauer: schnell, angemessen, langsam -> langsam kann ein Hinweis auf eine Unverständlichkeit sein, die verbessert werden muss.

- Wie sind Sie mit der Navigation zurecht gekommen?
- ++
- +
- /
- -
- - - Bemerkung: Alles verständlich. Offene Auswahl am Start. Ecke mit anderen Farben. Upload finden könnte man verbessern. Position in Header weniger versteckt (auch eingeklappt sichtbar).
- Waren die Angaben in den Previews für Sie aufschlussreich genug?
- ++
- +
- -

- - - Bemerkungen: Nicht mehr Angaben im Formular, sonst überfüllt -> weitere Informationen in Beschreibung ablegen. Informationen in der Darstellung ok. Am Anfang nicht alles gelesen, da sehr viel aber nachher schnell zurecht gefunden. Beschreibung und Titel sind die wichtigsten Informationen, Erfassungsdatum interessanter als Uploaddatum.
- Wie gefällt Ihnen das Design? (Achtung Geschmacksfrage)
- ++
- +
- /
- -
- - - Bemerkungen: Überfüllt, wie es sich bei GIS Programmen gehört. Nach kurzer Zeit aber klar und gut unterteilt. Ziel sind Fachleute im GIS, die sind sich das gewohnt.
- Was war besonders gut: Alles besprochen
- Was besonders schlecht: Alles besprochen
- Probleme aus dem Test aufgreifen: Alles besprochen

# Test

## Einleitung

Sie können keine Fehler machen. Alles, was Sie nicht können, machen auch andere falsch und muss eliminiert werden, bevor die Webseite benutzt werden kann. Die Aufzeichnungen werden am Ende des Projekts gelöscht. Das schriftliche Protokoll und die Auswertung über alle Tests werden abgelegt und stehen dem Projektteam, sowie dem Betreuer zur Verfügung. Wenn Sie zustimmen, würden wir neben der Gesamtauswertung auch das anonymisierte Protokoll als Anhang in die Dokumentation einfügen.

Der Test dient der Verbesserung des Punktwolken-Browsers.

1. Sind Sie damit einverstanden?

- Ja

2. Auch mit dem Anhängen des Protokolls?

- Ja

## Vorgehen

Zu Beginn werde ich Ihnen einige Fragen zu Ihren Vorkenntnissen stellen. Anschliessend geht es in den eigentlichen Test über. Ich werde Ihnen einige Szenarien vorlesen und Sie dürfen gerne jede ihrer Überlegungen laut äussern, während Sie dies umsetzen. Ihre mündlichen Inputs helfen uns, um zusehen, ob unsere Ideen auch bei den Benutzern ankommen.

Haben Sie noch fragen?

- Nein
- Wie schätzen Sie ihre Vorkenntnisse mit GIS-Programmen ein?
  - Sehr gut, verwende ich häufig. Welche:
  - Habe ich schon mal gebraucht. Welche:
  - Kenne ich, habe aber noch nie eines eingesetzt.
  - Kenne ich nicht. Bemerkung:
- Wissen Sie, was eine Punktwolke ist?
  - Ja, sehr genau.
  - Ja in etwa.
  - Nein. Bemerkungen:
- Benutzen sie online Kartendienste (Google, OSM, ...)?
  - Ja sehr häufig. Welche: Wenn ich nicht weiss wohin oder von wo die Rede ist, dann schaue ich immer auf Google Maps nach

- Ja manchmal. Welche:
- Nie Bemerkungen:

#### 1 | Punktwolke ansehen:

- Dauer bis ausgeführt: Schnell
- Merkt dass dies bereits offen ist.
- Inhalte gelesen: Links, ja rechts nein
- Seiten eingeklappt
- Angaben haben gefehlt:
- Übersicht unordentlich
- Weitere Bemerkungen: gescrollt und gedreht, echt cool, nach kurzem Ausprobieren Funktionsweise schnell begriffen. (Intuitiv bedienbar)

#### 2 | Punktwolke darstellen:

- Dauer bis ausgeführt: Schnell
- Zuerst Links gesucht
- Zuerst in Header gesucht
- Weitere Bemerkungen: Umstellung ändert zu wenig da die Punktwolke gar nicht so viele Punkte hat? Umstellung aber gefunden, zuerst aber versehentlich auf 2D geklickt aber selbständig bemerkt, dass das nun falsch war.

#### 3 | Darstellung teilen: Nicht umgesetzt (nicht sehr traurig, da man ja die Punktwolken freigeben kann reicht das ja)

#### 4 | Punktwolke hochladen:

- Dauer bis ausgeführt: Schnell
- Funktion ohne Eingriff gefunden
- Login ohne Eingriff versucht
- Unklare Attribute:
- Weitere Bemerkungen: Sehr gut Knopf mit Information. Login und Upload ohne Probleme Formular alle Felder befüllt. Rückmeldung sehr gut beim Upload und auch das Formular gerade auszufüllen, später füllt es doch eh niemand mehr aus. Aber dennoch nicht zu viele Zwangsfelder, wenn es mal schnell gehen muss.

#### 5 | Punktwolke bearbeiten:

- Dauer bis ausgeführt: Schnell
- Funktion ohne Eingriff gefunden
- Unklare Attribute:
- Weitere Bemerkungen: Erst unten in Potree Einstellungen gesucht...

#### 6 | 2D-Ansicht: Noch nicht dargestellt

- Dauer bis ausgeführt:
- Ansicht ohne Eingriff gefunden
- Ansicht verstanden
- Unklar was dies sein sollte?

- Inhalte gelesen
- Weitere Bemerkungen: Nicht wahnsinig schlimm, dass es seine Zeit braucht, wenn ich mir vorstelle, was alles berechnet werden muss. Sehr gut mit dem Cache beim 2. Mal ist es ja sofort da.

#### 7 | 2D-Ansicht downloaden:

- Dauer bis ausgeführt: Schnell
- Ansicht ohne Eingriff gefunden
- Knopfaufschrift unklar
- Weitere Bemerkungen: Anzeige im Fotoviewer nicht möglich, vielleicht für Anfänger beschreiben, damit diese nicht erschrecken, da Fotoviewer von einer kaputten Datei spricht.

#### 8 | 3D-Download:

- Dauer bis ausgeführt: Schnell
- Liste richtige Punktwolke gefunden
- Funktion ohne Eingriff gefunden
- Beschriftung unklar
- Weitere Bemerkungen:

\*Dauer: schnell, angemessen, langsam -> langsam kann ein Hinweis auf eine Unverständlichkeit sein, die verbessert werden muss.

- Wie sind Sie mit der Navigation zurecht gekommen?
- ++
- +
- /
- -
- - - Bemerkung:
- Waren die Angaben in den Previews für Sie aufschlussreich genug?
- ++
- +
- -
- - - Bemerkungen:
- Wie gefällt Ihnen das Design? (Achtung Geschmacksfrage)
- ++
- +
- /
- -

- - - Bemerkungen:
- Was war besonders gut: Upload verbessert, Logout noch kurz getestet auch gut gelöst, drehen um farbigen Punkt sehr angenehm, Seiten wegklickbar, Coole Farben in der Liste, klar wo man aktuell ist und wer Einsicht in diese Punktwolke hat, Schnell lernbar gut bedienbar, nicht mega langsam nichts sehr schlecht, Link zu Potree Bedienungsanleitung
- Was besonders schlecht: Visuell nicht erkennbar, was aktuell bei den Konfigurationen rechts ausgewählt ist (2D oder 3D), meine Punktwolken in der Liste sichtbar machen
- Probleme aus dem Test aufgreifen: -

# Test

## Einleitung

Sie können keine Fehler machen. Alles, was Sie nicht können, machen auch andere falsch und muss eliminiert werden, bevor die Webseite benutzt werden kann. Die Aufzeichnungen werden am Ende des Projekts gelöscht. Das schriftliche Protokoll und die Auswertung über alle Tests werden abgelegt und stehen dem Projektteam, sowie dem Betreuer zur Verfügung. Wenn Sie zustimmen, würden wir neben der Gesamtauswertung auch das anonymisierte Protokoll als Anhang in die Dokumentation einfügen.

Der Test dient der Verbesserung des Punktwolken-Browsers.

1. Sind Sie damit einverstanden?

- Ja

2. Auch mit dem Anhängen des Protokolls?

- Ja

## Vorgehen

Zu Beginn werde ich Ihnen einige Fragen zu Ihren Vorkenntnissen stellen. Anschliessend geht es in den eigentlichen Test über. Ich werde Ihnen einige Szenarien vorlesen und Sie dürfen gerne jede ihrer Überlegungen laut äussern, während Sie dies umsetzen. Ihre mündlichen Inputs helfen uns, um zusehen, ob unsere Ideen auch bei den Benutzern ankommen.

Haben Sie noch fragen?

- Nein
- Wie schätzen Sie ihre Vorkenntnisse mit GIS-Programmen ein?
  - Sehr gut, verwende ich häufig. Welche:
  - Habe ich schon mal gebraucht. Welche: QGIS
  - Kenne ich, habe aber noch nie eines eingesetzt.
  - Kenne ich nicht. Bemerkung:
- Wissen Sie, was eine Punktwolke ist?
  - Ja, sehr genau.
  - Ja in etwa.
  - Nein. Bemerkungen:
- Benutzen sie online Kartendienste (Google, OSM, ...)?
  - Ja sehr häufig. Welche: Leaflet beim Arbeiten, OSM beim Arbeiten, Google um Gegend zu erkunden.

- Ja manchmal. Welche:
- Nie Bemerkungen:

#### 1 | Punktwolke ansehen:

- Dauer bis ausgeführt: Schnell
- Merkt dass dies bereits offen ist.
- Inhalte gelesen
- Seiten eingeklappt
- Angaben haben gefehlt:
- Übersicht unordentlich
- Weitere Bemerkungen: Gescrollt und gedreht, Scrolling über Touchpad stoppt nicht

#### 2 | Punktwolke darstellen:

- Dauer bis ausgeführt: Schnell
- Zuerst Links gesucht
- Zuerst in Header gesucht
- Weitere Bemerkungen: Da wir im 3D sind, wird es wohl unter 3D sein. Umstellung ändert zu wenig, da die Punktwolke gar nicht so viele Punkte hat.

#### 3 | Darstellung teilen: Nicht umgesetzt

#### 4 | Punktwolke hochladen:

- Dauer bis ausgeführt: Schnell
- Funktion ohne Eingriff gefunden
- Login ohne Eingriff versucht
- Unklare Attribute:
- Weitere Bemerkungen: Sehr gut: Knopf mit Info, dass man eingeloggt sein muss, Login und Upload ohne Probleme

#### 5 | Punktwolke bearbeiten:

- Dauer bis ausgeführt: Schnell
- Funktion ohne Eingriff gefunden
- Unklare Attribute:
- Weitere Bemerkungen: Problemlos, Liste zugeklappt und nach Reload auf anderer Punktwolke gelandet -> etwas verwirrt

#### 6 | 2D-Ansicht: Noch nicht Dargestellt

- Dauer bis ausgeführt:
- Ansicht ohne Eingriff gefunden
- Ansicht verstanden
- Unklar was dies sein sollte?
- Inhalte gelesen
- Weitere Bemerkungen:

#### 7 | 2D-Ansicht downloaden: Noch ohne Funktionalität

- Dauer bis ausgeführt: Schnell (Bereich gefunden ohne Inhaltsanzeige)
- Ansicht ohne Eingriff gefunden
- Knopfaufschrift unklar
- Weitere Bemerkungen:

#### 8 | 3D-Download: Noch ohne Funktionalität

- Dauer bis ausgeführt: Schnell (Knopf gefunden ohne Funktionalität)
- Liste richtige Punktwolke gefunden
- Funktion ohne Eingriff gefunden
- Beschriftung unklar
- Weitere Bemerkungen:

\*Dauer: schnell, angemessen, langsam -> langsam kann ein Hinweis auf eine Unverständlichkeit sein, die verbessert werden muss.

- Wie sind Sie mit der Navigation zurecht gekommen?
  - ++
  - +
  - /
  - -
  - - - Bemerkung:
- Waren die Angaben in den Previews für Sie aufschlussreich genug?
  - ++
  - +
  - -
  - - - Bemerkungen:
- Wie gefällt Ihnen das Design? (Achtung Geschmacksfrage)
  - ++
  - +
  - /
  - -
  - - - Bemerkungen:
- Was war besonders gut: Upload verbessert
- Was besonders schlecht:-
- Probleme aus dem Test aufgreifen:-



## **4.4 Performance Test**

# Performance Test

---

## Test 1: Login

### Code

```
from datetime import datetime
import requests

startTime = datetime.now()
x = 1
failed = 0
while x <= 10:
 requestStart = datetime.now()
 response = requests.post("http://152.96.56.35:40000/login/token-auth/", data=
{"username": "denis", "password": "denisdenisdd"})
 print(datetime.now() - requestStart)
 print(x, ": ", response.status_code)
 if response.status_code != 200:
 failed += 1
 x +=1
```

### Resultate

```
0:00:00.528080 1 : 200 0:00:00.447407 2 : 200 0:00:00.366087 3 : 200 0:00:00.437593 4 : 200 0:00:00.517057 5 :
200 0:00:00.398788 6 : 200 0:00:00.398401 7 : 200 0:00:00.409562 8 : 200 0:00:00.358628 9 : 200
0:00:00.444730 10 : 200
```

## Test 2: List

### Code

```
from datetime import datetime
import requests

startTime = datetime.now()
token = requests.post("http://152.96.56.35:40000/login/token-auth/", data=
{"username": "denis", "password": "denisdenisdd"}).json()['token']
x = 1
faile = 0
while x <= 10:
 requestStart = datetime.now()
 response = requests.get("http://152.96.56.35:40000/api/pointclouds/", headers=
{'Authorization': 'JWT ' + token})
 print(datetime.now() - requestStart)
 print(x, ": ", response.status_code)
 if response.status_code != 200:
```

```
 faile += 1
 x +=1
```

## Resultate

0:00:00.119862 1 : 200 0:00:00.139078 2 : 200 0:00:00.096489 3 : 200 0:00:00.112922 4 : 200 0:00:00.117424 5 :  
200 0:00:00.110843 6 : 200 0:00:00.126211 7 : 200 0:00:00.138144 8 : 200 0:00:00.114462 9 : 200  
0:00:00.120735 10 : 200

## Test 3: Upload

### Code

```
from datetime import datetime
import requests

import json
import os
import io

startTime = datetime.now()
token = requests.post("http://152.96.56.35:40000/login/token-auth/", data=
{"username": "denis", "password": "denisdenisdd"}).json()['token']
x = 1
failed = 0
while x <= 10:
 requestStart = datetime.now()

 pointcloud = open('./fileToUpload.las', 'rb')
 total_size = os.path.getsize('./fileToUpload.las')

 slice_size = 1024 * 1024 * 16
 start = 0

 url = 'http://152.96.56.35:40000/api/upload/'

 while total_size > start:
 if total_size - (start + slice_size) < 0:
 end = total_size
 else:
 end = start + slice_size

 if(pointcloud.tell() + slice_size > total_size):
 chunk = pointcloud.read(total_size - pointcloud.tell())
 else:
 chunk = pointcloud.read(slice_size)
 chunk = io.BytesIO(chunk)
 contentrange = "bytes {0}-{1}/{2}".format(start, end-1, total_size)

 response = requests.put(url, data={"filename": 'fileToUpload.las'}, files=
{'file': chunk}, headers={'Authorization': 'JWT ' + token, 'Content-Range':
```

```
contentrange}))

 url = response.json()['url']
 start = response.json()['offset']

 response = requests.post(url, headers={'Authorization': 'JWT ' + token}, data=
{'name': 'fileToUpload'})

 print(datetime.now() - requestStart)
 print(x, ": ", response.status_code)
 if response.status_code != 201:
 failed += 1
 x +=1
```

## Resultate

```
0:00:12.353331 1 : 201 0:00:16.632231 2 : 201 0:00:22.326787 3 : 201 0:00:12.921507 4 : 201 0:00:14.309420 5 :
201 0:00:12.684259 6 : 201 0:00:11.295275 7 : 201 0:00:11.331599 8 : 201 0:00:14.031294 9 : 201
0:00:20.374778 10 : 201
```

## **5 Risiken**

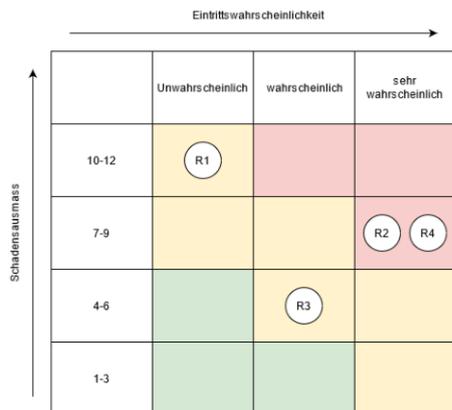
# Risiken

Gewichteter Schaden: 1-12 Eintrittswahrscheinlichkeit: unwahrscheinlich, wahrscheinlich, sehr wahrscheinlich

## Stand Projekt Initialisierung

| Nr | Titel                   | Beschreibung                                                                                                         | Eintritts-wahrscheinlichkeit | Gewichteter Schaden | Vorbeugung                                 | Verhalten beim Eintreten                                                                      |
|----|-------------------------|----------------------------------------------------------------------------------------------------------------------|------------------------------|---------------------|--------------------------------------------|-----------------------------------------------------------------------------------------------|
| R1 | Datenverlust            | Menschliches Versagen, Hardware Fehler, etc                                                                          | unwahrscheinlich             | 12                  | Versionierung aller Daten auf GitLab       | Alten Bestand aus GitLab History beziehen                                                     |
| R2 | Unbekannte Technologien | Das Team muss sich in unbekannte Technologien Lidarsensoren, Point Clouds und deren Geodatenverarbeitung einarbeiten | sehr wahrscheinlich          | 9                   | Ausführliche Recherche                     | Zusammenarbeit, gegenseitiges Helfen, Referenzpersonen zu Rate ziehen (IFS, GIS Spezialisten) |
| R3 | Programmiersprachen     | Das Team muss die vorgegeben Programmiersprachen Python und C++ mit wenig Vorkenntnissen/ Erfahrung verwenden        | wahrscheinlich               | 6                   | Ausführliche Recherche                     | Zusammenarbeit, gegenseitiges Helfen, Referenzpersonen zu Rate ziehen                         |
| R4 | Zeitplanung             | Das Team wird mit der Software im vorgegebenen Zeitraum nicht fertig                                                 | sehr wahrscheinlich          | 9                   | Projekt gut planen und Zeitplan optimieren | Mehr Zeit investieren und Features reduzieren                                                 |

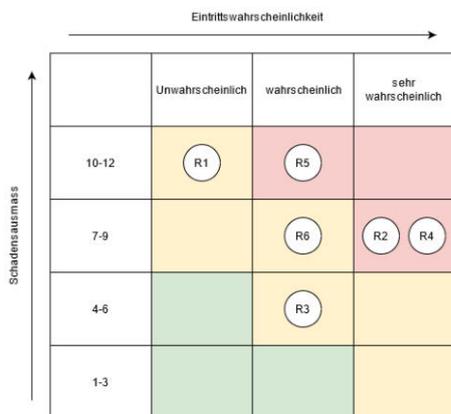
**Summe** 36



## Stand Ende von Inception

| Nr | Titel        | Beschreibung                                | Eintritts-wahrscheinlichkeit | Gewichteter Schaden | Vorbeugung                           | Verhalten beim Eintreten                  |
|----|--------------|---------------------------------------------|------------------------------|---------------------|--------------------------------------|-------------------------------------------|
| R1 | Datenverlust | Menschliches Versagen, Hardware Fehler, etc | unwahrscheinlich             | 12                  | Versionierung aller Daten auf GitLab | Alten Bestand aus GitLab History beziehen |

| Nr           | Titel                                | Beschreibung                                                                                                         | Eintrittswahrscheinlichkeit | Gewichteter Schaden | Vorbeugung                                                            | Verhalten beim Eintreten                                                                      |
|--------------|--------------------------------------|----------------------------------------------------------------------------------------------------------------------|-----------------------------|---------------------|-----------------------------------------------------------------------|-----------------------------------------------------------------------------------------------|
| R2           | Unbekannte Technologien              | Das Team muss sich in unbekannte Technologien Lidarsensoren, Point Clouds und deren Geodatenverarbeitung einarbeiten | sehr wahrscheinlich         | 9                   | Ausführliche Recherche                                                | Zusammenarbeit, gegenseitiges Helfen, Referenzpersonen zu Rate ziehen (IFS, GIS Spezialisten) |
| R3           | Programmiersprachen                  | Das Team muss die vorgegeben Programmiersprachen Python und C++ mit wenig Vorkenntnissen/ Erfahrung verwenden        | wahrscheinlich              | 6                   | Ausführliche Recherche                                                | Zusammenarbeit, gegenseitiges Helfen, Referenzpersonen zu Rate ziehen                         |
| R4           | Zeitplanung                          | Das Team wird mit der Software im vorgegebenen Zeitraum nicht fertig                                                 | sehr wahrscheinlich         | 9                   | Projekt gut planen und Zeitplan optimieren                            | Mehr Zeit investieren und Features reduzieren                                                 |
| R5           | Schlechte Dokumentation              | Die Dokumentation der Evaluationen, Tests und Erfahrungen mit Point Cloud-Technologien verfehlt das Ziel.            | wahrscheinlich              | 10                  | Requirements ausarbeiten und frühe Kontrolle der Zielerreichung       | Erheblicher Mehraufwand leisten                                                               |
| R6           | Kontinuierliche Anforderungsänderung | Die Requirements ändern sich oft                                                                                     | wahrscheinlich              | 7                   | Bestehende Tools gut evaluieren und Core-Requirements gut ausarbeiten | Mehraufwand leisten                                                                           |
| <b>Summe</b> |                                      |                                                                                                                      |                             | <b>53</b>           |                                                                       |                                                                                               |



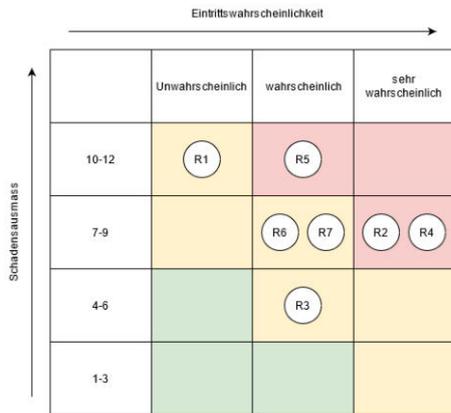
Stand Ende von Elaboration A

| Nr | Titel | Beschreibung | Eintrittswahrscheinlichkeit | Gewichteter Schaden | Vorbeugung | Verhalten beim Eintreten |
|----|-------|--------------|-----------------------------|---------------------|------------|--------------------------|
|----|-------|--------------|-----------------------------|---------------------|------------|--------------------------|

| Nr           | Titel                                   | Beschreibung                                                                                                                                                    | Eintritts-<br>wahrscheinlichkeit | Gewichteter<br>Schaden | Vorbeugung                                                                                                                                                                            | Verhalten beim<br>Eintreten                                                                                     |
|--------------|-----------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------|------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------|
| R1           | Datenverlust                            | Menschliches<br>Versagen, Hardware<br>Fehler, etc                                                                                                               | unwahrscheinlich                 | 12                     | Versionierung<br>aller Daten auf<br>GitLab                                                                                                                                            | Alten Bestand aus<br>GitLab History<br>beziehen                                                                 |
| R2           | Unbekannte<br>Technologien              | Das Team muss sich in<br>unbekannte<br>Technologien<br>Lidarsensoren, Point<br>Clouds und deren<br>Geodatenverarbeitung<br>einarbeiten                          | sehr wahrscheinlich              | 9                      | Ausführliche<br>Recherche                                                                                                                                                             | Zusammenarbeit,<br>gegenseitiges<br>Helfen,<br>Referenzpersonen<br>zu Rate ziehen<br>(IFS, GIS<br>Spezialisten) |
| R3           | Programmiersprachen                     | Das Team muss die<br>vorgegeben<br>Programmiersprachen<br>Python und C++ mit<br>wenig<br>Vorkenntnissen/<br>Erfahrung verwenden                                 | wahrscheinlich                   | 6                      | Ausführliche<br>Recherche                                                                                                                                                             | Zusammenarbeit,<br>gegenseitiges<br>Helfen,<br>Referenzpersonen<br>zu Rate ziehen                               |
| R4           | Zeitplanung                             | Das Team wird mit der<br>Software im<br>vorgegebenen<br>Zeitraum nicht fertig                                                                                   | sehr wahrscheinlich              | 9                      | Projekt gut<br>planen und<br>Zeitplan<br>optimieren                                                                                                                                   | Mehr Zeit<br>investieren und<br>Features<br>reduzieren                                                          |
| R5           | Schlechte<br>Dokumentation              | Die Dokumentation<br>der Evaluationen, Tests<br>und Erfahrungen mit<br>Point Cloud-<br>Technologien verfehlt<br>das Ziel.                                       | wahrscheinlich                   | 10                     | Requirements<br>ausarbeiten<br>und frühe<br>Kontrolle der<br>Zielerreichung                                                                                                           | Erheblicher<br>Mehraufwand<br>leisten                                                                           |
| R6           | Kontinuierliche<br>Anforderungsänderung | Die Requirements<br>ändern sich oft                                                                                                                             | wahrscheinlich                   | 7                      | Bestehende<br>Tools gut<br>evaluieren und<br>Core-<br>Requirements<br>gut<br>ausarbeiten                                                                                              | Mehraufwand<br>leisten                                                                                          |
| R7           | Schlechte<br>Dokumentation extern       | Die Dokumentation<br>der Tools, welche wir<br>verwenden wollen,<br>sind schlecht. Die<br>externen Tools und<br>Libraries sind<br>undokumentierte<br>Blackboxen. | wahrscheinlich                   | 7                      | Im Voraus bei<br>der<br>Entscheidung<br>schon<br>schauen, wie<br>gut die Tools<br>dokumentiert<br>sind.<br>Hochkomplexe<br>Tools ohne<br>Doku eher<br>nicht in<br>Betracht<br>ziehen. | Deutlicher<br>Mehraufwand<br>und<br>Kontaktaufnahme<br>Entwickler                                               |
| <b>Summe</b> |                                         |                                                                                                                                                                 |                                  | <b>60</b>              |                                                                                                                                                                                       |                                                                                                                 |

Infos:

- R2: In Elaboration B kann, nach dem definitiven Entscheid der zu verwendenden Technologien, das Risiko R2 in der Eintrittswahrscheinlichkeit runtergestuft werden.
- R4: Kann noch nicht heruntergestuft werden, erst wenn wir mit dem Programmieren begonnen haben (aber auch da nur eventuell).
- R5: Das Risiko wird beim Voranschreiten in der Arbeit immer weiter sinken, da wir Zwischenstände zur Kontrolle unserem Betreuer vorstellen werden.
- R6: In der Elaboration A konnten die Requirements zum Delivery 2 mal gefestigt werden, das Risiko bleibt aber bestehen.



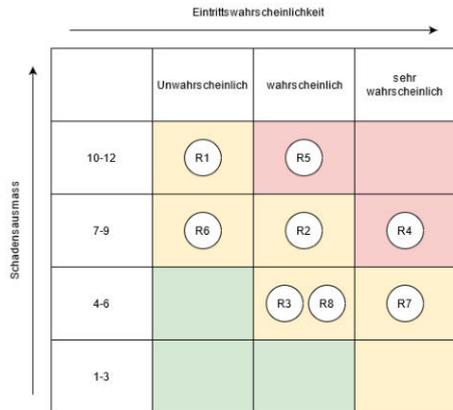
Stand Ende von Elaboration B

| Nr | Titel                   | Beschreibung                                                                                                         | Eintrittswahrscheinlichkeit | Gewichteter Schaden | Vorbeugung                                 | Verhalten beim Eintreten                                                                      |
|----|-------------------------|----------------------------------------------------------------------------------------------------------------------|-----------------------------|---------------------|--------------------------------------------|-----------------------------------------------------------------------------------------------|
| R1 | Datenverlust            | Menschliches Versagen, Hardware Fehler, etc                                                                          | unwahrscheinlich            | 12                  | Versionierung aller Daten auf GitLab       | Alten Bestand aus GitLab History beziehen                                                     |
| R2 | Unbekannte Technologien | Das Team muss sich in unbekannte Technologien Lidarsensoren, Point Clouds und deren Geodatenverarbeitung einarbeiten | wahrscheinlich              | 9                   | Ausführliche Recherche                     | Zusammenarbeit, gegenseitiges Helfen, Referenzpersonen zu Rate ziehen (IFS, GIS Spezialisten) |
| R3 | Programmiersprachen     | Das Team muss die vorgegeben Programmiersprachen Python und C++ mit wenig Vorkenntnissen/ Erfahrung verwenden        | wahrscheinlich              | 6                   | Ausführliche Recherche                     | Zusammenarbeit, gegenseitiges Helfen, Referenzpersonen zu Rate ziehen                         |
| R4 | Zeitplanung             | Das Team wird mit der Software im vorgegebenen Zeitraum nicht fertig                                                 | sehr wahrscheinlich         | 9                   | Projekt gut planen und Zeitplan optimieren | Mehr Zeit investieren und Features reduzieren                                                 |

| Nr           | Titel                                   | Beschreibung                                                                                                                                                    | Eintritts-<br>wahrscheinlichkeit | Gewichteter<br>Schaden | Vorbeugung                                                                                                                                                                            | Verhalten beim<br>Eintreten                                       |
|--------------|-----------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------|------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------|
| R5           | Schlechte<br>Dokumentation              | Die Dokumentation<br>der Evaluationen, Tests<br>und Erfahrungen mit<br>Point Cloud-<br>Technologien verfehlt<br>das Ziel.                                       | wahrscheinlich                   | 10                     | Requirements<br>ausarbeiten<br>und frühe<br>Kontrolle der<br>Zielerreichung                                                                                                           | Erheblicher<br>Mehraufwand<br>leisten                             |
| R6           | Kontinuierliche<br>Anforderungsänderung | Die Requirements<br>ändern sich oft                                                                                                                             | unwahrscheinlich                 | 7                      | Bestehende<br>Tools gut<br>evaluieren und<br>Core-<br>Requirements<br>gut<br>ausarbeiten                                                                                              | Mehraufwand<br>leisten                                            |
| R7           | Schlechte<br>Dokumentation extern       | Die Dokumentation<br>der Tools, welche wir<br>verwenden wollen,<br>sind schlecht. Die<br>externen Tools und<br>Libraries sind<br>undokumentierte<br>Blackboxen. | sehr wahrscheinlich              | 5                      | Im Voraus bei<br>der<br>Entscheidung<br>schon<br>schauen, wie<br>gut die Tools<br>dokumentiert<br>sind.<br>Hochkomplexe<br>Tools ohne<br>Doku eher<br>nicht in<br>Betracht<br>ziehen. | Deutlicher<br>Mehraufwand<br>und<br>Kontaktaufnahme<br>Entwickler |
| R8           | Hohe Komplexität                        | Die von uns<br>erarbeitete Architektur<br>lässt sich schwer<br>umsetzen.                                                                                        | wahrscheinlich                   | 6                      | Frühe<br>Anbindung<br>aller Tools und<br>Aufsetzen der<br>Codebasis.                                                                                                                  | Mit Begründung:<br>Architektur<br>ändern.                         |
| <b>Summe</b> |                                         |                                                                                                                                                                 |                                  | <b>64</b>              |                                                                                                                                                                                       |                                                                   |

## Infos:

- R2: Von sehr wahrscheinlich auf wahrscheinlich heruntergestuft, da schon viel Recherche betrieben wurde.
- R4: Kann noch nicht heruntergestuft werden, erst wenn wir mit dem Programmieren begonnen haben (aber auch da nur eventuell).
- R6: Die Requirements zu Delivery 2 und 3 scheinen relativ sicher zu sein -> Eintrittswahrscheinlichkeit von wahrscheinlich zu unwahrscheinlich herabgestuft.
- R7: Potree ist nicht sehr ausführlich dokumentiert. Das wird sehr wahrscheinlich zu etwas Mehraufwand führen und deshalb wurde R7 auf sehr wahrscheinlich hochgestuft. Da die komplexen Tools aber relativ gut dokumentiert sind, wurde der gewichtete Schaden auf 5 heruntergestuft.
- R8: Neu hinzu kommt "Hohe Komplexität", da die Umsetzung des Tools so wie geplant doch sehr anspruchsvoll werden könnte. Mit allfälligen Architekturänderungen bei Problemen kann dem aber entgegen gewirkt werden, weswegen der Schaden auf eine 6 eingeschätzt wird.



### Stand Ende von Prototyp

Es wird erst Ende von Alpha wieder eine Einschätzung gemacht.

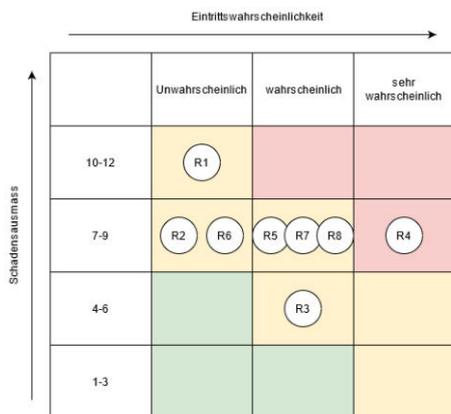
### Stand Ende von Alpha

| Nr | Titel                   | Beschreibung                                                                                                         | Eintritts-wahrscheinlichkeit | Gewichteter Schaden | Vorbeugung                                                      | Verhalten beim Eintreten                                                                      |
|----|-------------------------|----------------------------------------------------------------------------------------------------------------------|------------------------------|---------------------|-----------------------------------------------------------------|-----------------------------------------------------------------------------------------------|
| R1 | Datenverlust            | Menschliches Versagen, Hardware Fehler, etc                                                                          | unwahrscheinlich             | 12                  | Versionierung aller Daten auf GitLab                            | Alten Bestand aus GitLab History beziehen                                                     |
| R2 | Unbekannte Technologien | Das Team muss sich in unbekannte Technologien Lidarsensoren, Point Clouds und deren Geodatenverarbeitung einarbeiten | unwahrscheinlich             | 9                   | Ausführliche Recherche                                          | Zusammenarbeit, gegenseitiges Helfen, Referenzpersonen zu Rate ziehen (IFS, GIS Spezialisten) |
| R3 | Programmiersprachen     | Das Team muss die vorgegeben Programmiersprachen Python und C++ mit wenig Vorkenntnissen/ Erfahrung verwenden        | wahrscheinlich               | 6                   | Ausführliche Recherche                                          | Zusammenarbeit, gegenseitiges Helfen, Referenzpersonen zu Rate ziehen                         |
| R4 | Zeitplanung             | Das Team wird mit der Software im vorgegebenen Zeitraum nicht fertig                                                 | sehr wahrscheinlich          | 9                   | Projekt gut planen und Zeitplan optimieren                      | Mehr Zeit investieren und Features reduzieren                                                 |
| R5 | Schlechte Dokumentation | Die Dokumentation der Evaluationen, Tests und Erfahrungen mit Point Cloud-Technologien verfehlt das Ziel.            | wahrscheinlich               | 8                   | Requirements ausarbeiten und frühe Kontrolle der Zielerreichung | Erheblicher Mehraufwand leisten                                                               |

| Nr           | Titel                                | Beschreibung                                                                                                                               | Eintrittswahrscheinlichkeit | Gewichteter Schaden | Vorbeugung                                                                                                                                     | Verhalten beim Eintreten                              |
|--------------|--------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------|---------------------|------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------|
| R6           | Kontinuierliche Anforderungsänderung | Die Requirements ändern sich oft                                                                                                           | unwahrscheinlich            | 7                   | Bestehende Tools gut evaluieren und Core-Requirements gut ausarbeiten                                                                          | Mehraufwand leisten                                   |
| R7           | Schlechte Dokumentation extern       | Die Dokumentation der Tools, welche wir verwenden wollen, sind schlecht. Die externen Tools und Libraries sind undokumentierte Blackboxen. | wahrscheinlich              | 7                   | Im Voraus bei der Entscheidung schon schauen, wie gut die Tools dokumentiert sind. Hochkomplexe Tools ohne Doku eher nicht in Betracht ziehen. | Deutlicher Mehraufwand und Kontaktaufnahme Entwickler |
| R8           | Hohe Komplexität                     | Die von uns erarbeitete Architektur lässt sich schwer umsetzen.                                                                            | wahrscheinlich              | 9                   | Frühe Anbindung aller Tools und Aufsetzen der Codebasis.                                                                                       | Mit Begründung: Architektur ändern.                   |
| <b>Summe</b> |                                      |                                                                                                                                            |                             | <b>67</b>           |                                                                                                                                                |                                                       |

Infos:

- R2: Von wahrscheinlich auf unwahrscheinlich heruntergestuft, da nicht mehr viele neue Technologien dazu kommen sollten.
- R3: Würden wir zwar gerne herunterstufen, es können aber immer wieder neue Herausforderungen hinzukommen (Upload?)
- R4: Kann noch nicht heruntergestuft werden, da wir leicht in Verzug gekommen sind.
- R5: Wurde von Schaden 10 auf 8 heruntergestuft. Es wurde schon einiges mit dem Betreuer besprochen und wir scheinen auf gutem Wege zu sein. Die Eintrittswahrscheinlichkeit bleibt, der Schaden wird aber bei Eintritt nicht so gross sein.
- R7: Potree konnte integriert werden. Die Eintrittswahrscheinlichkeit wurde deshalb heruntergestuft.
- R8: Eine Architekturänderung wäre "schädlicher" als nur 6. Neu: 9.



Stand Ende von Beta

Es wird erst Ende von Final wieder eine Einschätzung gemacht.

## Stand Ende von Final

| Nr | Titel                                | Beschreibung                                                                                                         | Eintrittswahrscheinlichkeit | Gewichteter Schaden | Vorbeugung                                                            | Verhalten beim Eintreten                                                                      |
|----|--------------------------------------|----------------------------------------------------------------------------------------------------------------------|-----------------------------|---------------------|-----------------------------------------------------------------------|-----------------------------------------------------------------------------------------------|
| R1 | Datenverlust                         | Menschliches Versagen, Hardware Fehler, etc                                                                          | unwahrscheinlich            | 12                  | Versionierung aller Daten auf GitLab                                  | Alten Bestand aus GitLab History beziehen                                                     |
| R2 | Unbekannte Technologien              | Das Team muss sich in unbekannte Technologien Lidarsensoren, Point Clouds und deren Geodatenverarbeitung einarbeiten | unwahrscheinlich            | 4                   | Ausführliche Recherche                                                | Zusammenarbeit, gegenseitiges Helfen, Referenzpersonen zu Rate ziehen (IFS, GIS Spezialisten) |
| R3 | Programmiersprachen                  | Das Team muss die vorgegeben Programmiersprachen Python und C++ mit wenig Vorkenntnissen/ Erfahrung verwenden        | -                           | -                   | Ausführliche Recherche                                                | Zusammenarbeit, gegenseitiges Helfen, Referenzpersonen zu Rate ziehen                         |
| R4 | Zeitplanung                          | Das Team wird mit der Software im vorgegebenen Zeitraum nicht fertig                                                 | unwahrscheinlich            | 9                   | Projekt gut planen und Zeitplan optimieren                            | Mehr Zeit investieren und Features reduzieren                                                 |
| R5 | Schlechte Dokumentation              | Die Dokumentation der Evaluationen, Tests und Erfahrungen mit Point Cloud-Technologien verfehlt das Ziel.            | unwahrscheinlich            | 8                   | Requirements ausarbeiten und frühe Kontrolle der Zielerreichung       | Erheblicher Mehraufwand leisten                                                               |
| R6 | Kontinuierliche Anforderungsänderung | Die Requirements ändern sich oft                                                                                     | unwahrscheinlich            | 7                   | Bestehende Tools gut evaluieren und Core-Requirements gut ausarbeiten | Mehraufwand leisten                                                                           |

| Nr           | Titel                             | Beschreibung                                                                                                                               | Eintritts-<br>wahrscheinlichkeit | Gewichteter<br>Schaden | Vorbeugung                                                                                                                                     | Verhalten beim<br>Eintreten                           |
|--------------|-----------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------|------------------------|------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------|
| R7           | Schlechte<br>Dokumentation extern | Die Dokumentation der Tools, welche wir verwenden wollen, sind schlecht. Die externen Tools und Libraries sind undokumentierte Blackboxen. | -                                | -                      | Im Voraus bei der Entscheidung schon schauen, wie gut die Tools dokumentiert sind. Hochkomplexe Tools ohne Doku eher nicht in Betracht ziehen. | Deutlicher Mehraufwand und Kontaktaufnahme Entwickler |
| R8           | Hohe Komplexität                  | Die von uns erarbeitete Architektur lässt sich schwer umsetzen.                                                                            | -                                | -                      | Frühe Anbindung aller Tools und Aufsetzen der Codebasis.                                                                                       | Mit Begründung: Architektur ändern.                   |
| R9           | Zeitplanung Doku                  | Das Team wird mit der Dokumentation im vorgegebenen Zeitraum nicht fertig                                                                  | unwahrscheinlich                 | 8                      | Weniger relevante Abschnitte zum Schluss schreiben.                                                                                            | Mehraufwand leisten                                   |
| <b>Summe</b> |                                   |                                                                                                                                            |                                  | <b>48</b>              |                                                                                                                                                |                                                       |

## Infos:

- R2: Wenn jetzt noch unbekannte Technologien auftreten, dann ist die Auswirkung sehr gering, da fast alles fertig implementiert ist.
- R3: Kann nicht mehr eintreten (Beim Refactoring und Kleinigkeiten dürfte es keine Probleme mehr geben)
- R4: Es fehlt nicht mehr viel. Es ist unwahrscheinlich, dass wir nicht fertig werden.
- R5: Auf unwahrscheinlich heruntergestuft, weil schon viel darüber mit dem Betreuer geredet wurde.
- R7: Es müssen keine externen Tools mehr eingebunden/ angezapft werden.
- R8: Architekturänderungen gibt es nicht mehr.
- R9: Neu dazugekommen - unwahrscheinlich weil wir schon einiges an "roher" Doku haben. Schaden nur eine 8, weil für die relevantesten Themen schon Entwürfe geschrieben wurden.

 3d options

## **6 Organisation**

### **6.1 Besprechungsnotizen**

# Protokolle für Meetings mit Herrn Keller

---

## 10.06.2021

- Blog Post -> Schauen wir nach der Abgabe
- Abstract -> Ist praktisch fertig

## 03.06.2021

- Erarbeitetes Requirement share View werden wir aus zeitlichen Gründen weglassen -> OK
- Dokument unterzeichnen -> Dokument per PDF an Herrn Keller senden Weitere Verbesserungen:
- In Punktwolken Liste die Fields gross schreiben (Macht Description da Sinn?)
- Release Type umbenennen in "Access type"
- Potree Informationen in 3D List (rechts) unterdrücken
- Potree Schrift und eigene gleich gross in Liste (rechts)
- Pointcloud-Browser -> Pointcloud browser
- Point clouds bei Liste (links) umbenennen in "list"
- Buttons mit Grossbuchstaben anfangen
- Geotiff.js anschauen

## 20.05.2021

- Lizenz für Pointcloudbrowser und Pointcloudpipeline (BSD-2?) ->  
[https://en.wikipedia.org/wiki/ISC\\_license](https://en.wikipedia.org/wiki/ISC_license) (auch Open-BSD genannt) für Pointcloudpipeline und BSD-2 für Pointcloudbrowser (wegen Potree muss ja BSD-2)
- Wer ist GitLab Verantwortliche(r)? -> Nicola Jordan anfragen (Update: Er meint, dass wir "von Hand" deployen sollen, da es nicht so scheint, als würde der Fehler bei uns liegen und die BA ja nicht mehr lange geht.)
- Unterschriebener Auftrag -> Im Wiki  
[https://wiki.ost.ch/display/~stefan.keller@ost.ch/SABA\\_FS21\\_Slot\\_2\\_Aufgabenstellung](https://wiki.ost.ch/display/~stefan.keller@ost.ch/SABA_FS21_Slot_2_Aufgabenstellung) (Unterschrift nicht nötig) -> Aufgabestellung finalisiert

## 06.05.2021

- Austauschplattform für Punktwolken -> Switch Drive
- Daten Lizenz -> Zu den Daten ein License-File schreiben: "dl/by-2-0" (d.h. "Freie Nutzung. Quellenangabe ist Pflicht" gemäss <https://opendata.swiss/de/terms-of-use> )

## 29.04.2021

- Keine Entscheidungen nötig

## 22.04.2021

- Erwähnen/ notieren, dass nur Pointcloud Apps getestet wurden - nicht photogrammetrie-Apps

## 15.04.2021

- Attribut Ausdehnung können wir vorläufig weglassen.

## 08.04.2021

- Entscheide (Technologie und Viewer)
- Architektur (Idee vorstellen)
- Wireframe (Zusammenklappen der Liste und About/Header)
- "React" und andere Produktnamen würde ich von Anfang an "richtig" schreiben.
- Wirklich React vs. reines JS -> React
- Qgis Issue erfassen wegen des mühsamen Drehens von 3D-Daten
- Wireframe (About Page)
- Name: Viewer, Browser, Studio
- Metadaten für Punktwolken bestimmt

## 01.04.2021

Apps: SiteScape, 3d Scanner App, (poly) polycam und Scaniverse

- Potree vs Cesium
- PDAL vs PCL
- Technologie entscheid (Python django vs Flask oder Bottle) (react vs Vue.js "Pointviewer integrieren")  
Web Applikation -> nächste Woche Tabelle mit Gewichtung
- Potree vs Cesium -> Besser dokumentieren
- Im Anhang muss nicht "korrekt" zitiert werden
- PDAL vs PCL - Gegenüberstellung machen (Tabelle)
- Use Case Diagramm -> Klar, dass es zur Webapplikation gehört
- Django vs Flask vs Bottle, React vs Vue (Pointviewer integrieren?)
- Messungen: Wann, wo, wer, mit welchen Geräten?

| Wann                      | Wo               | Was                                                       | Wer                                                        | Geräte                                            |
|---------------------------|------------------|-----------------------------------------------------------|------------------------------------------------------------|---------------------------------------------------|
| 14.04.21 13:00<br>Uhr     | OST<br>Campus RJ | Einführung Leica<br>BLK2GO, Messung<br>Skulptur           | Nadine Sennhauser,<br>Philipp Kramer, Roger<br>Bräm        | Leica BLK2GO                                      |
| 20.04.21<br>13:00Uhr      | OST<br>Campus RJ | Einführung<br>GeoSLAM Zeb Revo<br>RT, Messung<br>Skulptur | Nadine Sennhauser,<br>Denis Nauli, Jürg<br>Pulver (Online) | GeoSLAM Zeb Revo<br>RT, iPad Pro                  |
| 21.04.21 oder<br>28.04.21 | Weinfelden<br>TG | Messung Ruine<br>Neuenburg                                | Nadine Sennhauser,<br>Denis Nauli, Marc<br>Vögele          | iPad Pro, GeoSLAM<br>Zeb Revo RT, Leica<br>BLK2GO |

| Wann                                            | Wo         | Was                                          | Wer                                                                  | Geräte                                                        |
|-------------------------------------------------|------------|----------------------------------------------|----------------------------------------------------------------------|---------------------------------------------------------------|
| 23.04.21 11.00<br>-12.00Uhr +<br>13.00-16.00Uhr | Rapperswil | Messung Villa<br>Aurum,<br>Referenzmessungen | Nadine Sennhauser,<br>Denis Nauli, Marc<br>Vögele oder Roger<br>Bräm | IPad Pro, GeoSLAM<br>Zeb Revo RT, Leica<br>BLK2GO, Tachymeter |

## 25.03.2021

- <https://de.wikipedia.org/wiki/WebVR>
- Pcap is quite standard as a lidar format -> Anschauen
- <https://sitn.ne.ch/lidar/>
- Eisenskulptur durch Sandstein ersetzen? -> Nein, da wir so auch Limits aufzeigen können -> Es müssen Apps verglichen werden (auch kostenpflichtige) -> Bescheid @ Keller welche das sein könnten
- Bei Entscheidungen Vor- und Nachteile auflisten

## 19.03.2021

- Requirements / Usecases
- Titel der Arbeit festgelegt: Geoprocessing mit Handheld-Laser-Scanner erfassten Point Cloud-Daten
- Ende der Soll Customer Journey Map ein Fliesstext (Wie ein Pitch an einen Kunden aus der Zielgruppe).  
Beispiel "Mit unserer Software ghet ihr als Archäologen raus, erledigt das was ihr in 4 Stunden macht in 10 Minuten, geht dann nach Hause und ladet die Point Cloud hoch. Ihr editiert die Pointcloud, bereitet sie schön auf und teilt sie per Link mit Ihren Berufskollegen. Diese können in der Pointcloud dann auch Messungen vornehmen." -> Das für die 3 Zielgruppen Archäologe, Denkmalpfleger/ -Schützer und Mapper.
- Use Cases neu sortieren (Reihenfolge im Diagramm ändern)
- Gewichtung von Requirements anpassen

## 11.03.2021

### Meshes

- Bei der Archäologen weit verbreitet
- Nachbearbeitung der Bruchkanten
- Zentrale Ablage via Punktwolke mit Visionszielen

### Commandline Tool

- Ja Punkte nicht in Commandline löschen.
- Daten hochladen
- Und eventuell Möglichkeit Viewer anzustossen für Bereinigung
- Programme: Compare....

<https://www.opendronemap.org/webodm/>

- Umfang prüfen

- Nur bis zur Punktwolke oder die auch bearbeiten? D3.js
- Einsatz prüfen
- Ist weit verbreitet; Nicht zwingend bei Punktwolken aber im 3D Bereich

Input Kommunikation mit Dejan Šeatović

- Abschätzung der Genauigkeit des iPads
- Die Specs scheinen noch unbekannt <https://developer.apple.com/forums/thread/131013>
- <https://www.st.com/resource/en/datasheet/vl53l1x.pdf>
- <https://www.sciencedirect.com/science/article/pii/B9780444516626500058>

04.03.2021

## Nachtrag

- Philipp Kramer bei Leica Gerät mitnehmen
- Genauigkeitsanalyse der Point Cloud (PC)
- Use Case Messen Strecken/Distanzen/Abstände

Messung

1. Skulptur Eisen/Plastik OST Campus
2. Archäologie Mauer Neuenburg TG
3. Gebäude Villa Aurum

Referenzdaten beschaffen für GPS bei Geometer in Schmerikon Lukas Domaisen - PP-Punkte aus dem Grundbuch

Dejan Šeatović: Bei fragen betreffend Messen und genauigkeiten ()

Requirements

- Inhalt Part 1 ok - Rechtschreibung überarbeiten (z.B. Qgis -> QGIS)
- Unterschied Punktwolke und Mesh (Blender und Godot visualisieren nur Meshes)
- Rohdaten immer archivieren

Server

- Mehr Leistung von Beginn weg.
- Wenn nötig aufstocken per Mail möglich

Rollen 3 Usergruppen - Archäologen, Denkmalpfleger und Mapper

Verschiedene Bedürfniss der Einzelnen festhalten. Als Denkmalpfleger auf einer Grabstätte... -> Teamentscheid mit Absprache Joel: Prsonas und Journeymap zur Dokumentation sind ok "Als Denkmalpfleger will ich ein Denkmalobjekt draussen schnell kartieren und aufgearbeitet im Web visualisieren können."

Beschlüsse

- Server bestellen
- iPad und MacBook abholen wenn verfügbar
- MacBook aufsetzen

## 25.02.2021 (Kickoff Meeting)

### Einführung ins Projekt (Wichtige Punkte)

#### Geräte:

- GeoSLAM ZEB HORIZON (Bekommen wir zur Verfügung gestellt, Partner wollen dafür aber Logo in Doku) Unklar wie wir an die Rohdaten kommen
- Leica BLK2GO (Wir brauchen Einführung, damit wir das Gerät richtig bedienen) Ebenfalls unbekannt, wie wir an die Rohdaten kommen
- Ipad Pro - Hat Lidar Sensor im Low-End Bereich eingebaut (→ 10x ungenauer und nur Range von 10 Metern) → Wegen Apple bräuchten wir noch einen Mac (Mac am Institut vorhanden)

Infrastruktur Beschaffung: Ipad Pro durch Ost beschafft, überlegen was brauchen wir noch?

Bestehende Lösungen: Darstellung via Kacheln gibt es schon (nur dann flüssig), entwine vor dem Kacheln verarbeiten (nicht klare Verwendung) Aufbearbeitung PDAL Pottree, Daten ansehen und video erste sachen: <https://www.youtube.com/watch?v=pe04m-gHM-M> (Warum hat Klas Karlsson nicht schon in opendronemap die Daten "gecleart" und hat noch ein weiteres Tool verwendet?), Papers suchen, lesen!, Ablauf optimieren was kann man weglassen

Geräte Testen: Mit Roger Bräm BLM Termin abmachen für Leica (mit Einführung). Mit Jürg Pulfer Termin für GeoSLAM Outdoor und mit GPS, Range 1 bis 25 Meter nicht 3D Print vorlage, Evaluation: Was ist die Genauigkeit, mittlere Genauigkeit von einem Punkt im Raum in 97.5% der Fälle, was ist die Auflösung → Wie genau sind die Daten, wie viel cm in 97.5% der Fälle ist es daneben. Wir sollten wenn möglich auch Papers und Bücher referenzieren - zeigen, dass wir nicht die ersten sind, die sowas machen.

Verwendungszwecke der von uns zu liefernder Webapplikation: Webapplikation Archeologen - im schlimmsten Fall zeichnen die von Hand die Ausgrabung ab - im besten Fall mit 3-Bein alles abscannen → Idee mit diesem "Teekrug"-Sensor langsam durch Baugrube gehen und zu Hause kann man es auf die Webapplikation hochladen und Resultat allen zur Verfügung stellen + von 3D nach 2D Ansicht der Punktwolke. Wichtigkeit der Arbeit vielleicht auch hier ersichtlich: <https://pixel8earth.medium.com/should-the-3d-map-of-the-globe-be-a-public-good-c50e3e667b0e>

Ausblick (nicht mehr in unserem Scope): Mehrfach scannen und dann das zusammenrechnen, damit das Ergebnis immer genauer wird → Übereinanderlegen geht zwar, aber nicht ein verbesserter Datensatz durch Algorithmus.

#### Beschlüsse

- Ipad wird beschafft

## **6.2 Konstruktionsplan**

# Konstruktionsplan

---

## Prototyp

### Frontend

- Overview Page mit Platzhaltern
- Einbindung von Potree für die 3D-View
- Leere Platzhalter für restliche Funktionen

### Backend

- File Struktur aufsetzen (initialisieren mit default Daten)
- DB aufsetzen (initialisieren mit default Daten)
- Pointcloud von Filestruktur -> Pipeline -> Backend -> Frontend
- Metadaten von DB -> Backend -> Frontend

### Sonstiges

- API Documentation (Swagger)
- Sonarqube aufsetzen
- Default Data definieren

## Alpha

### Frontend

- Upload-Funktion (PointCloud und Metadaten)
- Edit-PointCloud-Metadaten Knopf und Löschen der Pointcloud und Metadaten
- Punktwolken Liste (mit Filter - zum Auswählen von Punktwolken)
- Login

### Backend

- Upload-Funktion (PointCloud in LAZ speichern, Metadaten in DB speichern)
- Punktwolke löschen Funktion (LAZ + Metadaten)
- Metadaten bearbeiten (Neue Daten in DB)
- Liste von Pointclouds (nur Metadaten) -> Frontend
- Login

## Beta

### Frontend

- Overview Page Design
- Download Button mit Funktionalität
- Edit PointCloud in CloudCompare
- 2D-Ansicht

## Backend

- PointCloud LAZ -> Frontend für Download
- PointCloud Controller: Pointcloud LAZ nach dem Editieren (in CloudCompare editieren) speichern
- 2D Controller: 2D inklusive Prozessierung 3D Punktwolke zu 2D-ableitbarer Ansicht -> Frontend

## Final

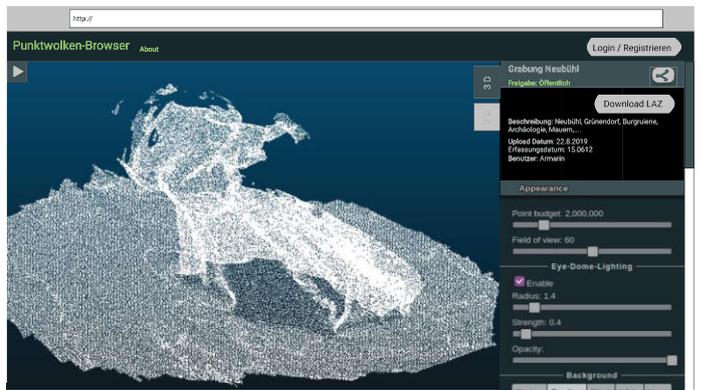
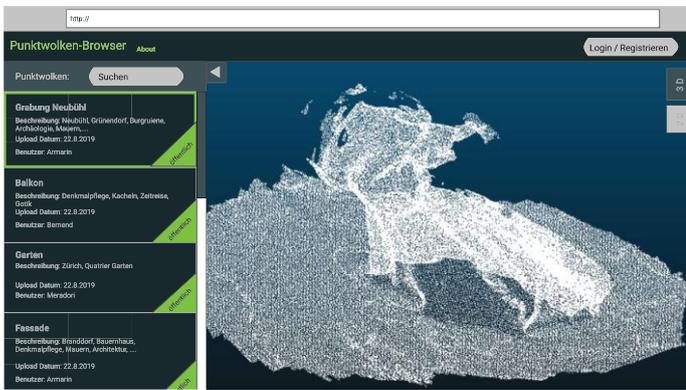
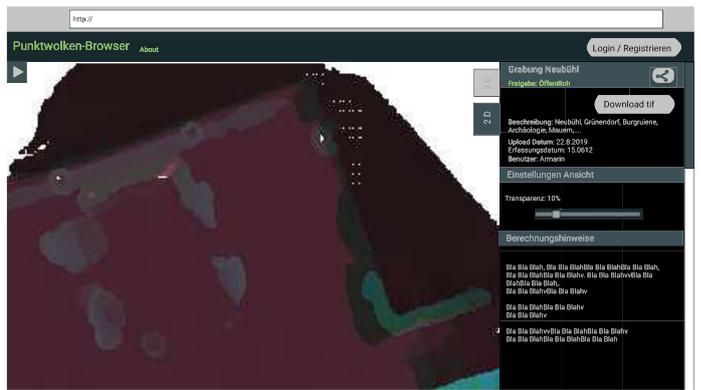
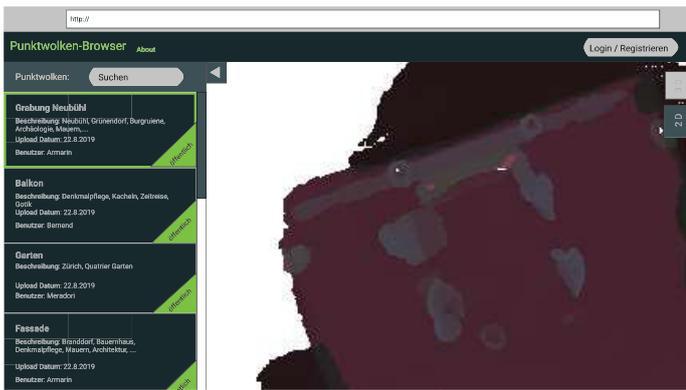
### Frontend

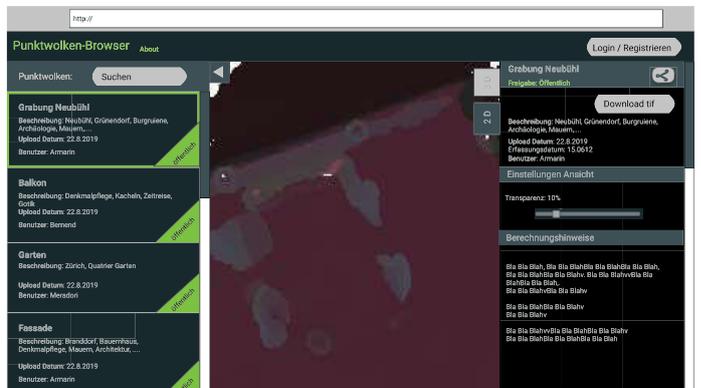
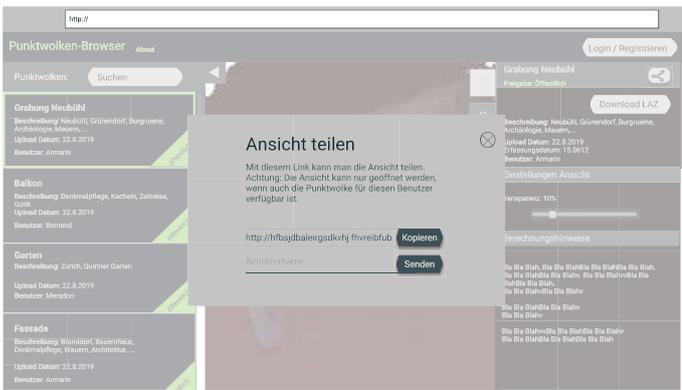
- Error-Handling
- My Profile Ansicht
- Button Design teilen (via URL)
- Verbesserungen Usability Test Beta

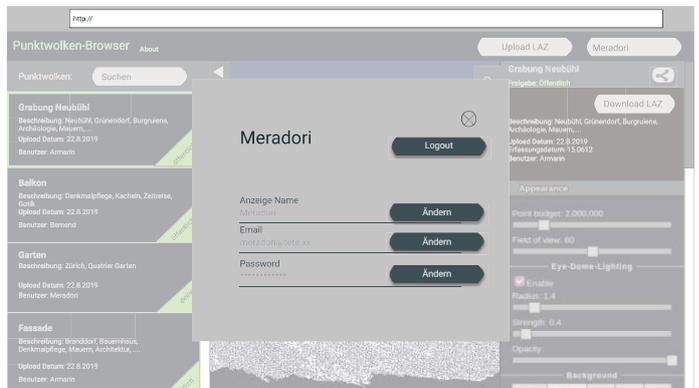
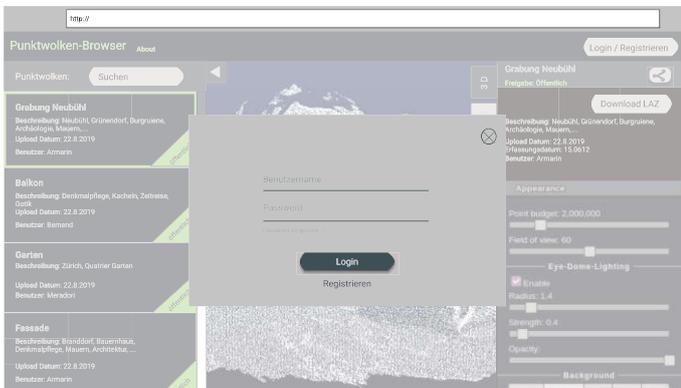
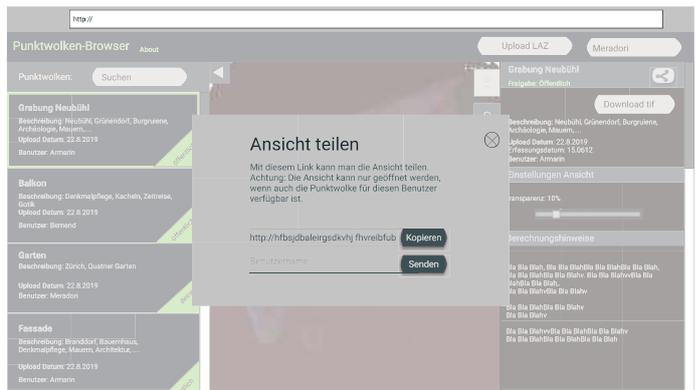
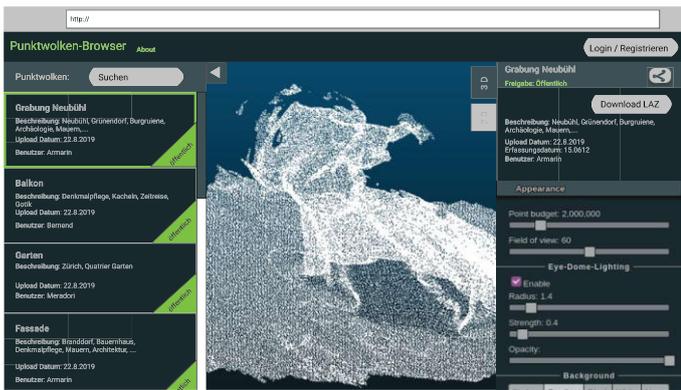
### Backend

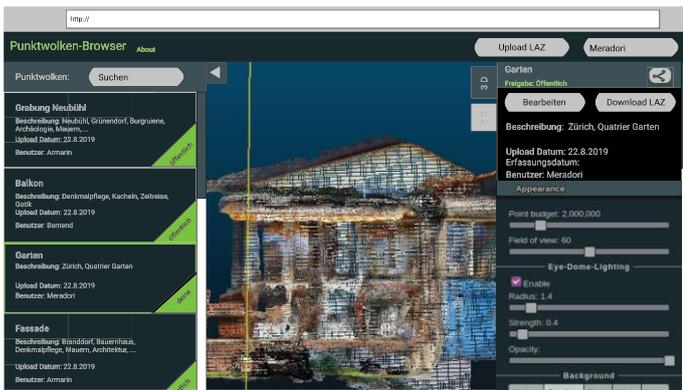
- Login Erweiterung (Erweiterung Password neusetzen und Mail ändern)
- Performance Optimierung (Vorprozessieren häufig verwendeter Daten)

## **7 UI-Entwurf**



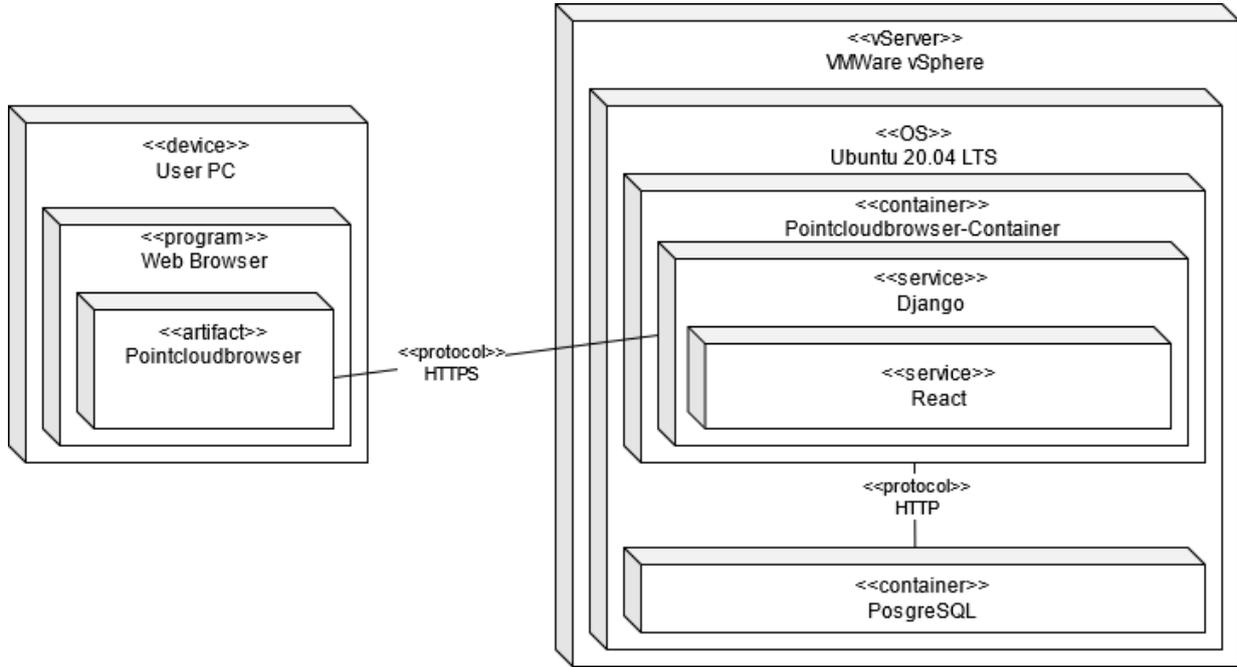




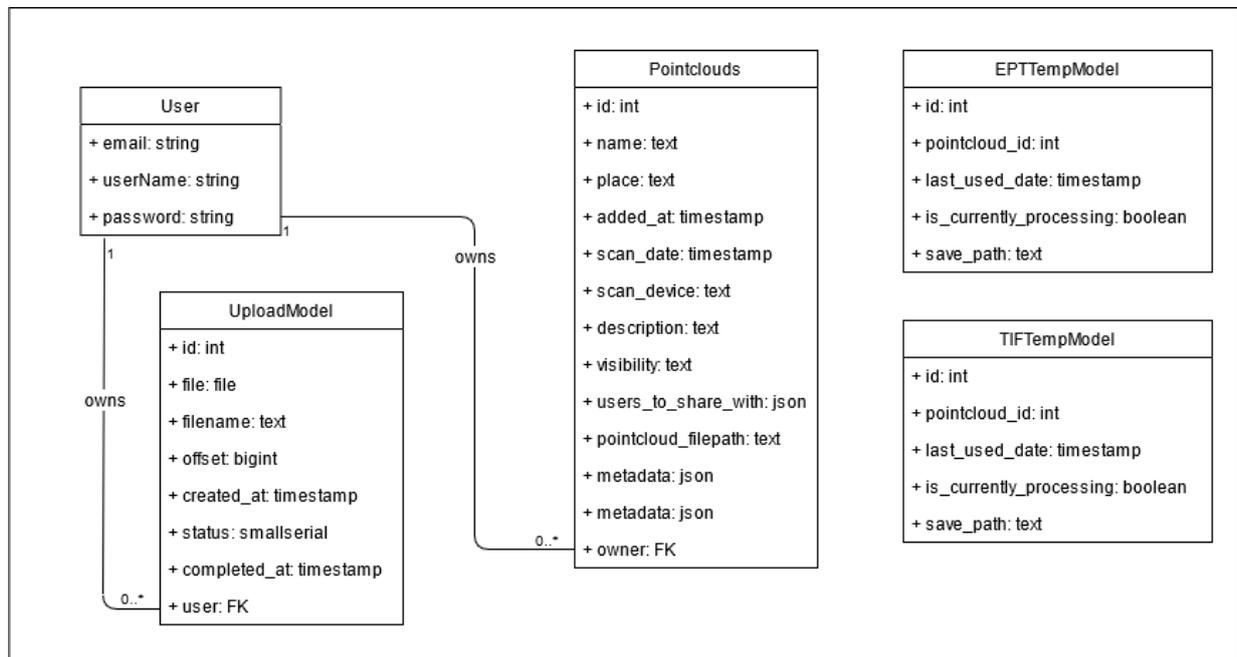


## **8 Architektur**

# Deployment



# Datenspeicherung



## **9 Pointcloudbrowser Manual**

# Punktwolken-Browser

---

## Installation

Um den Punktwolken-Browser zu starten muss Docker und Docker-Compose installiert werden.

Install Docker <https://docs.docker.com/engine/install/ubuntu/>

```
1. $ sudo apt update
```

```
2. $ sudo apt-get install \
 apt-transport-https \
 ca-certificates \
 curl \
 gnupg-agent \
 software-properties-common
```

```
3. $ curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo apt-key add
 -
```

```
4. $ sudo add-apt-repository \
 "deb [arch=amd64] https://download.docker.com/linux/ubuntu \
 $(lsb_release -cs) \
 stable"
```

```
5. $ sudo apt-get install docker-ce docker-ce-cli containerd.io
```

Install Docker-Compose <https://docs.docker.com/compose/install/>

```
1. $ sudo curl -L
 "https://github.com/docker/compose/releases/download/1.27.4/docker-
 compose-$(uname -s)-$(uname -m)" -o /usr/local/bin/docker-compose
```

```
2. $ sudo chmod +x /usr/local/bin/docker-compose
```

## Start Pointcloudbrowser

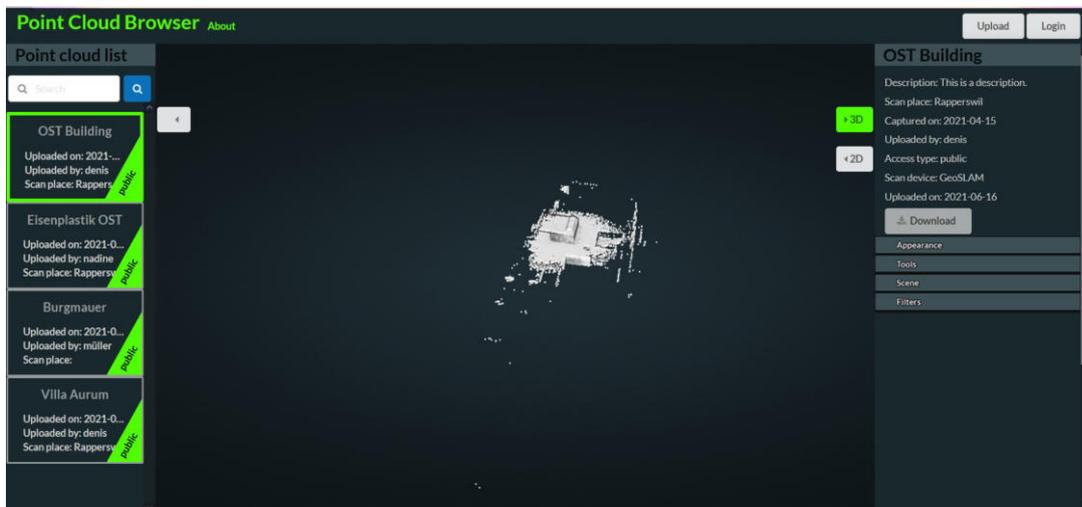
```
1. $ git clone https://gitlab.ost.ch/nadine.sennhauser/pointcloudbrowser.git
```

```
2. $ cd ./pointcloudbrowser
```

```
3. $ docker-compose up --build
```

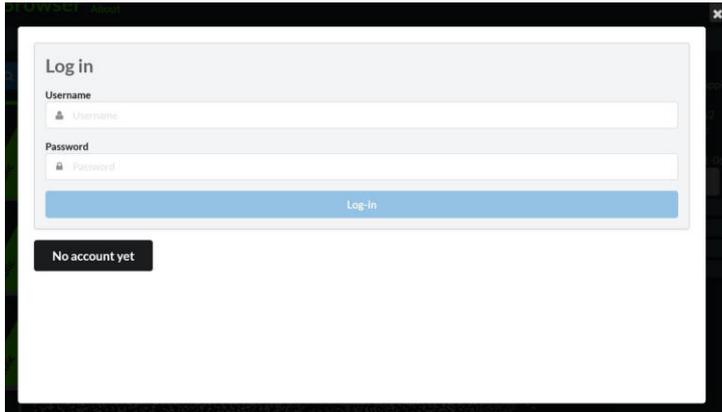
## User Manual

Point Cloud Browser is a web-based point cloud visualization and management software that allows you to not only visually configure your point clouds according to your needs, but also share them with other users. So that you don't have to keep the point clouds twice, they can be downloaded at any time. Thus, the Point Cloud Browser also serves as a management program. As a guest, all public point clouds can be viewed, while as a registered user, team point clouds and private ones can also be displayed. If a 2D view is needed, it can be calculated, visualized and downloaded directly with the tool.

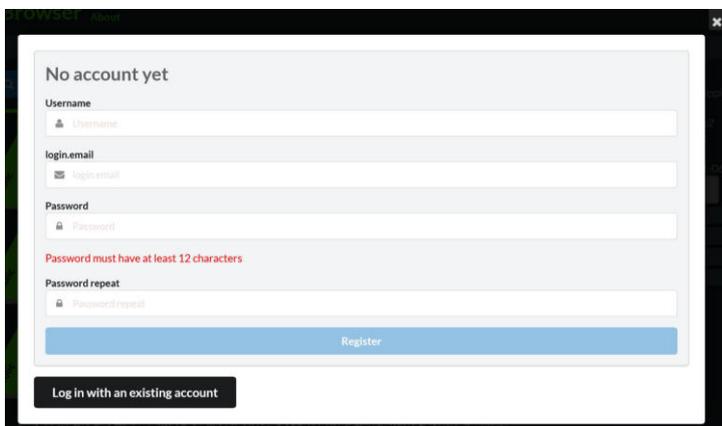


## Login and register

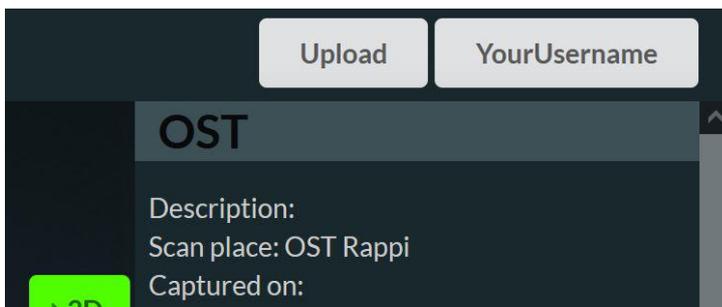
The login button is in the top right corner of the screen. By clicking on it, a window will appear where you can log in.



If you do not have an account yet, you can create one by clicking on "No account yet". The registration form will then open.

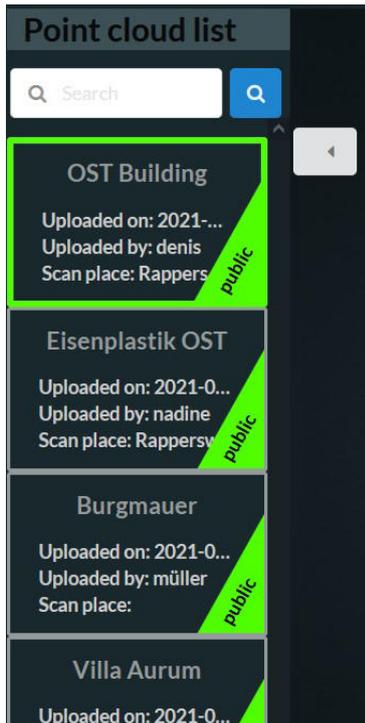


If you want to logout, click on your name in the top right corner.



Visualize point cloud

On the left side there is the selection list of point clouds.



If you are not logged in, you will only see a selection of public point clouds. By clicking on an entry, the point cloud is displayed in the center of the screen.

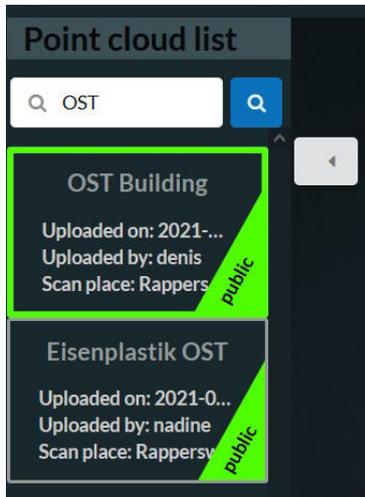
If you are logged in, you will also see your private and shared point clouds.



For more space for your point cloud in the center of the screen, click on the arrow button near the list close it. You can later unfold the list if you wish to change the point cloud.

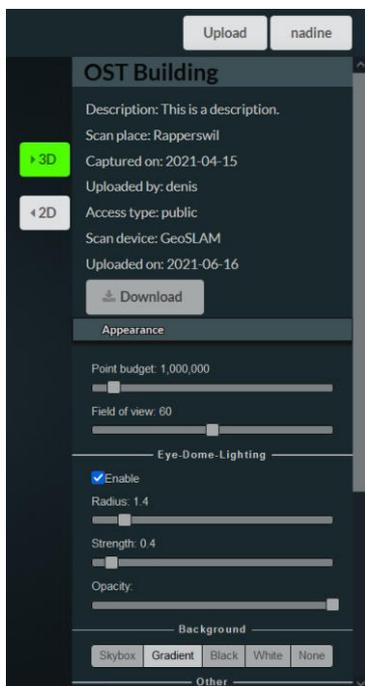
### Find the right point cloud

In the left list side panel you can search for keywords to find the right point cloud.



### Visualize in 3D

Once you have selected a point cloud, you can click on the 3D button on the right side of the screen to expand the settings.

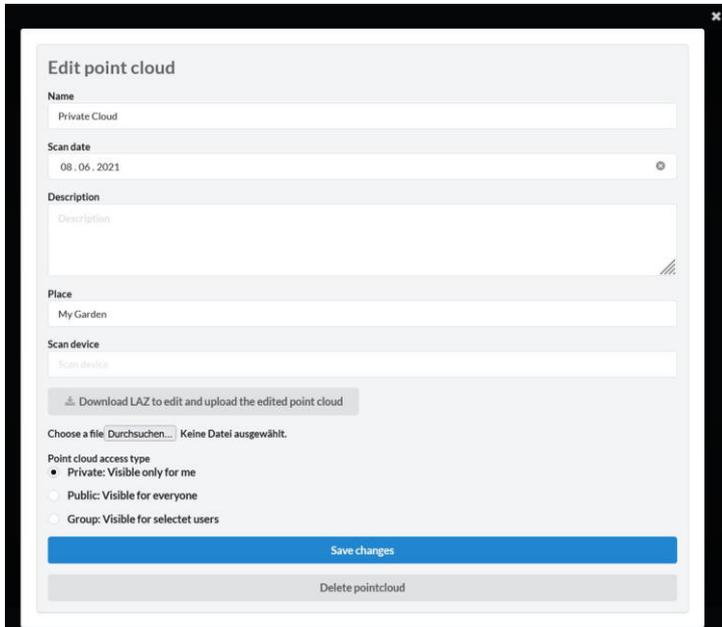


In the 3d options you can change the visualization, perform **measurements** or **download** the point cloud in the LAZ format.

If you are logged in, you will also find an **edit** button on your point clouds.

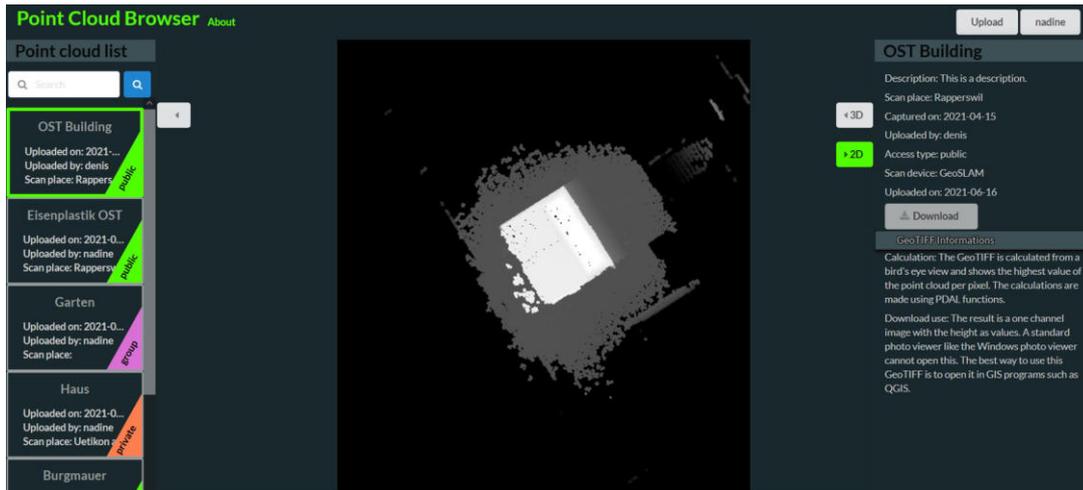


If you click that edit button, you can edit all metadata or even upload a newer version of your point cloud. If you no longer need the point cloud, you can also **delete** it.



### Visualize in 2D

Once you have selected a point cloud, you can click on the 2D button on the right side of the screen to expand the settings.



There you can download the 2D GeoTIFF file.

## **10 Pointcloudpipeline Manual**

# Pointcloudpipeline

---

A simple commandline app for managing pointclouds in LAZ. There are functions to get a laz out of e57, ply or las as well as a function to convert laz back to e57, ply or las and these will show you all the metadata on your commandline. You also can convert pointclouds into a viewable Entwine format or into a 2D raster.

If you want to use the functionality as import in a project and not as a commandline app you can use the functions over the import statement.

## Installation

---

### Preparation

1. Install Python and pipenv

```
$ sudo apt install python3
$ sudo apt install python3-pip
$ sudo apt install pipenv
```

2. Install git

```
$ sudo apt install git
```

3. Install PDAL for converting point clouds

```
$ sudo apt install libpdal-dev
$ sudo apt install cmake
$ sudo apt install pdal
$ pdal --version (should be min 2.1 or higher, but not 2.0.1)
```

4. Install untwine <https://github.com/hobu/untwine.git> anywhere you want. Later you need only the path to the entwine build data. (Example: /home/user/Documents/original/untwine/build/entwine)

```
$ git clone https://github.com/hobu/untwine.git
$ cd untwine
$ mkdir build
$ cd build
$ cmake ..
$ make
```

### Manual installation Pointbrowserpipeline

```
$ git clone https://gitlab.ost.ch/nadine.sennhauser/pointcloudpipeline
$ cd pointcloudpipeline
$ pipenv shell
$ pipenv install
$ python setup.py install
```

## Using Pip

```
$ pip install pointcloudpipeline
```

## Usage

---

### As a commandline tool

Functions:

- convertfromlaz: Convert a LAZ-pointcloud to desired type.
  - Options:
    - --lazFile TEXT: LAZ-file to convert [required]
    - --directoryTo TEXT: Directory to save converted file
    - --convertTo [las|laz|e57|ply]: Desired filetype
    - --help: Show help
- convertlazto2d: Convert LAZ to a top rasterview in 2D.
  - Options:
    - --lazFile TEXT: LazFile to convert [required]
    - --directoryTo TEXT: Directory to save
    - --help: Show help
- convertlaztoeptlaz: Convert a pointcloud from LAZ to EPT-LAZ.
  - Options:
    - --lazFile TEXT: LazFile to convert [required]
    - --directoryTo TEXT: Directory to save
    - --untwinePath TEXT: Path to untwine
    - --help: Show help.
- converttolaz: Convert a pointcloud from ['las', 'laz', 'e57', 'ply'] to LAZ and save it.
  - Options:
    - --file TEXT: File to convert to LAZ [required]
    - --directoryTo TEXT: Directory to save LAZ
    - --help: Show help.

## Options:

- --version: Show the version and exit.
- --help: Show this message and exit.

## Example:

```
$ pointcloudpipeline convertlazto2d --lazFile
/home/hacker/Documents/original/pointcloudpipeline/test_data/from/test.laz --
directoryTo /home/hacker/Documents/original/pointcloudpipeline/test_data/to
```

```
{
 "metadata":
 {
 "readers.las":
 {
 "comp_spatialreference": "",
 "compressed": true,
 "count": 1843780,
 "creation_doy": 119,
 "creation_year": 2021,
 "dataformat_id": 2,
 "dataoffset": 327,
 "filesource_id": 0,
 "global_encoding": 0,
 "global_encoding_base64": "AAA=",
 "header_size": 227,
 "major_version": 1,
 "maxx": 33.4728,
 "maxy": 39.8786,
 "maxz": 9.5308,
 "minor_version": 2,
 "minx": -72.9241,
 "miny": -19.287,
 "minz": -1.1721,
 "offset_x": 0,
 "offset_y": 0,
 "offset_z": 0,
 "point_length": 26,
 "project_id": "00000000-0000-0000-0000-000000000000",
 "scale_x": 0.0001,
 "scale_y": 0.0001,
 "scale_z": 0.0001,
 "software_id": "GeoSLAM",
 "spatialreference": "",
 "srs":
 {
 "compoundwkt": "",
 "horizontal": "",
 "isgeocentric": false,
 "isgeographic": false,
 "prettycompoundwkt": "",
 "prettywkt": "",

```

```
 "proj4": "",
 "units":
 {
 "horizontal": "unknown",
 "vertical": ""
 },
 "vertical": "",
 "wkt": ""
},
"system_id": "libLAS",
"vlr_0":
{
 "data":
"AgAAAAIAAgAAAAAUMMAAP////////////////////////////////////8CAAYAFACAAgABgACAA==",
 "description": "http://laszip.org",
 "record_id": 22204,
 "user_id": "laszip encoded"
}
},
"writers.gdal":
{
 "filename":
 [
 "/home/hacker/Documents/original/pointcloudpipeline/test_data/to/test.tif"
]
}
}
}
```

## As import in other projects

```
from pipeline import pipeline_worker
```

Functions:

- `convert_laz_to_ept_laz(laz_file, directory_to, untwine_path)`
- `convert_to_laz(file, directory_to)`
- `convert_from_laz(laz_file, directory_to, type_out)`
- `convert_laz_to_2d(laz_file, directory_to)`

Example:

```
pipelineworker.convert_laz_to_2d(/home/Documents/original/pointcloudpipeline/test_
data/test.laz /home/hacker/Documents/original/pointcloudpipeline/test_data/to)
```

## **11 Glossar**

# Glossar

**BSD-2** BSD-2 besagt, dass die Software unter der Angabe des Urheberrechtes frei verwendet werden und unter der gleichen Lizenz weiterverbreitet werden kann.. 6

**Chunks** Chunks sind nicht überlappende Datenstrom-Teile.. 4

**dockerisiert** Der Begriff dockerisiert beschreibt, dass der entsprechende Softwareteil in einem Dockercontainer liegt.. 4

**fotogrammetrisch** Ein Verfahren bei welchem Punktwolken über Bilder zu berechnen werden.. 1

**Genauigkeit** Die Genauigkeit ist in der Messtechnik die Kombination von Richtigkeit und Präzision.. 1

**Geoprocessing** Verarbeiten und Verwalten von Geodaten ohne neues Wissen zu generieren. Zum Beispiel: Koordinatenkonvertierung.. 1

**GNSS** GNSS ist eine Technologie zur satellitengestützten Positionsermittlung. 70

**Handlaserscanner** Portable Laserscanner um Punktwolken zu erfassen.. 1

**innere Orientierung** Die innere Orientierung beschreibt die Lage des Projektionszentrums der Kamera relativ zur Bildebene.. 18

**Kacheln** Das DOM (Digitales Oberflächen Modell) ist eine Abbildung der Oberfläche.. 60

**Kalman-Filter** Das Kalman-Filter dient dazu, nicht direkt messbare Systemgrößen zu schätzen. 18

**LiDAR** Abkürzung für Englisch "Light detection and ranging" auch "Light imaging, detection and ranging". Technologie um zum Beispiel Punktwolken zu Messen.. 1

**Normalen** Die Normalen bilden die Senkrechten auf eine Oberfläche.. 16

**OpenStreetMap** OpenStreetMap (OSM) ist ein freies Projekt, das frei nutzbare Geodaten sammelt, strukturiert und für die Nutzung durch jedermann in einer Datenbank vorhält (Open Data).. 3

**Orthofotos** Ein Orthofoto ist eine verzerrungsfreie und massstabsgetreue Abbildung der Erdoberfläche. 71

**Punktwolken** Sammlungen von Punkten mit 3D-Koordinaten und ggf. weiteren Attributen.. 1

**SLAM** Abkürzung für Englisch "Simultaneous Localization and Mapping". Dabei wird während dem Messen die Position des Aufnahmegeräts ermittelt.. 1

**Stop-and-Go** Stop-and-Go ist ein Staiv-basiertes Verfahren, um Punktwolken zu erfassen.. 5

**Vektor** Der Vektor ist eine Grösse, die die Richtung und Länge beschreibt. . 17

**Voxels** Der Voxel entspricht einem Pixel im 3D-Raum.. 24

## 12 Quellenverzeichnis

[1] Punktwolke, <https://de.wikipedia.org/wiki/Punktwolke>, letzter Zugriff am 07.03.2021

[2] Dieter Fritsch, "Punktwolken, und was dann ?", Vermessung & Geoinformation, 1/2015, S. 54 – 66, 12 Abb

[3] What are point clouds? 5 easy facts that explain point clouds, <https://info.vercator.com/blog/what-are-point-clouds-5-easy-facts-that-explain-point-clouds>, letzter Zugriff am 08.03.2021

[4] Introduction to LiDAR, <https://pdal.io/workshop/lidar-introduction.html>, letzter Zugriff 05.03.2021

- [5] Airborne.Laserscanning, [https://de.wikipedia.org/wiki/Airborne\\_Laserscanning](https://de.wikipedia.org/wiki/Airborne_Laserscanning), letzter Zugriff 11.06.2021
- [6] The C4 model for visualising software architecture, <https://c4model.com/>, letzter Zugriff am 15.06.2021
- [7] Mesh Bearbeitung mit MeshLab, <https://www.scanner.imagefact.de/de/mesh.html>, letzter Zugriff am 16.06.2021
- [8] Fotogrammetrie, <https://de.wikipedia.org/wiki/Photogrammetrie>, letzter Zugriff am 10.03.2021
- [9] What is SLAM and how does it work?, <https://geoslam.com/what-is-slam/>, letzter Zugriff am 02.03.2021
- [10] PointShare+ Useful Features, <https://pointshareplus.com/features>, letzter Zugriff am 15.06.2021
- [11] Structured and Unstructured data, <https://www.laserscanning-europe.com/en/laserscanning-forum/english-laser-scanning-forum/laserscanning-software/structured-unstructured-data>, letzter Zugriff am 26.03.2021
- [12] Point Cloud Formats . . . and What They Mean, [https://www.kaart.com/scanning\\_tips/point-cloud-formats/](https://www.kaart.com/scanning_tips/point-cloud-formats/), letzter Zugriff am 26.03.2021
- [13] Point\_set\_registration, [https://en.m.wikipedia.org/wiki/Point\\_set\\_registration](https://en.m.wikipedia.org/wiki/Point_set_registration), letzter Zugriff am 26.03.2021
- [14] Can anyone explain the difference between organized and unorganized point cloud?, <https://stackoverflow.com/questions/18994680/can-anyone-explain-the-difference-between-organized-and-unorganized-point-cloud>, letzter Zugriff am 26.03.2021
- [15] Massive point cloud data management: Design, implementation and execution of a point cloud benchmark, [https://www.researchgate.net/publication/272373258\\_Massive\\_point\\_cloud\\_data\\_management\\_Design\\_implementation\\_and\\_execution\\_of\\_a\\_point\\_cloud\\_benchmark/link/5adede7b458515c60f6181bb/download](https://www.researchgate.net/publication/272373258_Massive_point_cloud_data_management_Design_implementation_and_execution_of_a_point_cloud_benchmark/link/5adede7b458515c60f6181bb/download), letzter Zugriff am 27.03.2021
- [16] Entwine Point Tile, <https://entwine.io/entwine-point-tile.html>, letzter Zugriff am 27.03.2021
- [17] pgPointCloud - A PostgreSQL extension for storing point cloud (LIDAR) data., <https://pgpointcloud.github.io/pointcloud/index.html>, letzter Zugriff am 27.03.2021
- [18] Vue vs React in 2021: Which Framework to Choose and When, <https://www>.

monterail.com/blog/vue-vs-react-2021, letzter Zugriff am 03.05.2021

[19] Bild, [https://blog.prusaprinters.org/wp-content/uploads/2018/03/epipolar\\_geometry.jpg](https://blog.prusaprinters.org/wp-content/uploads/2018/03/epipolar_geometry.jpg), letzter Zugriff am 16.06.2021

[20] How to integrate Potree into react or VueJS ?, <https://github.com/potree/potree/issues/939>, letzter Zugriff 13.05.2021

[21] Django TokenAuthentication missing the 'Authorization' http header, <https://stackoverflow.com/questions/15113248/django-tokenauthentication-missing-the-authorization-header>, letzter Zugriff 16.06.2021

[22] Foto, <https://www.scanner2go-shop.com/de/leica-blk2go-laserscanner.html>, letzter Zugriff 16.06.2021

[23] Difference of Normals Based Segmentation, [https://pcl.readthedocs.io/projects/tutorials/en/latest/don\\_segmentation.html](https://pcl.readthedocs.io/projects/tutorials/en/latest/don_segmentation.html), letzter Zugriff 16.06.2021

## 13 Codeverzeichnis

# Listings

|      |                                         |    |
|------|-----------------------------------------|----|
| II.1 | Test mit zu kleiner Auflösung . . . . . | 88 |
| II.2 | Test mit zu grosser Auflösung . . . . . | 88 |
| II.3 | Umwandlung GeoTIFF zu JEPEG . . . . .   | 89 |
| II.4 | Caption hinzufügen . . . . .            | 90 |

## 14 Abbildungsverzeichnis

|      |                                                                    |    |
|------|--------------------------------------------------------------------|----|
| 1    | Punktwolke der Villa Aurum gemessen mit dem GeoSLAM ZEB Revo-RT    | 4  |
| 2    | Overview Diagramm . . . . .                                        | 5  |
| 3    | Benutzeroberfläche Point Cloud Browser . . . . .                   | 6  |
| I.1  | Beispiel Fotogrammetrie [19] . . . . .                             | 19 |
| I.2  | Visualisierung der Octree Struktur . . . . .                       | 24 |
| I.3  | Eisenplastik OST im Potree visualisiert . . . . .                  | 30 |
| I.4  | Eisenplastik OST im Cesium visualisiert . . . . .                  | 30 |
| I.5  | Mauer der Ruine Neuenburg in Weinfelden . . . . .                  | 41 |
| I.6  | Messung der Eisenplastik OST mit dem GeoSLAM ZEB-Revo RT . . . . . | 41 |
| I.7  | Villa Aurum in Rapperswil mit den Referenzpunkten . . . . .        | 42 |
| I.8  | Referenzmessung . . . . .                                          | 43 |
| I.9  | BLK2GO[22] . . . . .                                               | 44 |
| I.10 | GeoSLAM ZEB-Revo RT . . . . .                                      | 46 |
| I.11 | Messung mit iPad Pro . . . . .                                     | 47 |
| I.12 | Punktwolke aus Structure from Motion . . . . .                     | 52 |
| I.13 | Punktwolke von Leica BLK2GO . . . . .                              | 52 |
| I.14 | Punktwolke von GeoSLAM ZEB-Revo RT . . . . .                       | 53 |
| I.15 | Scaniverse . . . . .                                               | 53 |

|                                                                            |     |
|----------------------------------------------------------------------------|-----|
| I.16 Polycam . . . . .                                                     | 53  |
| I.17 3d Scanner App . . . . .                                              | 54  |
| I.18 SiteScape . . . . .                                                   | 54  |
| I.19 Aufbau Überblick . . . . .                                            | 57  |
| I.20 Upload . . . . .                                                      | 58  |
| I.21 Download . . . . .                                                    | 58  |
| I.22 Berechnung der 3D-Ansicht . . . . .                                   | 59  |
| I.23 Berechnung der 2D-Ansicht . . . . .                                   | 61  |
| II.1 Hauptpersona Archäologe . . . . .                                     | 69  |
| II.2 Visionspersona Opensource Mapper . . . . .                            | 69  |
| II.3 UseCase-Diagramm . . . . .                                            | 74  |
| II.4 C4 Modell Level 1 . . . . .                                           | 82  |
| II.5 C4 Modell Level 2 . . . . .                                           | 83  |
| II.6 C4 Modell Level 3 Frontend . . . . .                                  | 84  |
| II.7 C4 Modell Level 3 Backend . . . . .                                   | 84  |
| II.8 C4 Modell Level 3 Pointcloudpipeline . . . . .                        | 85  |
| II.9 PDAL Fehlversuch der Funktion Cloth Simulation Filter (CSF) . . . . . | 86  |
| II.10 PCL Fehlversuch der Funktion range image . . . . .                   | 87  |
| II.11 difference of normals based segmentation Vorgang[23] . . . . .       | 87  |
| II.12 Test mit zu kleiner Auflösung . . . . .                              | 88  |
| II.13 Test mit zu grosser Auflösung und zu vielen Strukturen . . . . .     | 89  |
| II.14 Verwendete Auflösung . . . . .                                       | 90  |
| II.15 Wireframe Design . . . . .                                           | 92  |
| II.16 Design am Ende . . . . .                                             | 92  |
| II.17 Stakeholder Map . . . . .                                            | 97  |
| II.18 Risiken Inception Phase . . . . .                                    | 99  |
| II.19 Risiken Final Phase . . . . .                                        | 99  |
| II.20 Zeitaufwendung pro Phase . . . . .                                   | 102 |
| II.21 Zeitaufwendung pro Teammitglied . . . . .                            | 102 |
| II.22 Zeitaufwendung pro Aufwandstyp . . . . .                             | 103 |
| II.23 SonarQube-Übersicht . . . . .                                        | 103 |
| II.24 Schnittstellendokumentation mit Swagger . . . . .                    | 106 |

## 15 Tabellenverzeichnis

|      |                                                               |    |
|------|---------------------------------------------------------------|----|
| I.1  | Evaluation Apps . . . . .                                     | 26 |
| I.2  | Punktwolken vs Meshes . . . . .                               | 27 |
| I.3  | Vergleichstabelle der Punktwolken Viewer . . . . .            | 29 |
| I.4  | Vergleichstabelle von Potree vs Cesium . . . . .              | 29 |
| I.5  | Vergleichstabelle der Datenverwaltungsmöglichkeiten . . . . . | 31 |
| I.6  | Vergleichstabelle der Punktwolkenverarbeitung . . . . .       | 33 |
| I.7  | Vergleichstabelle der PDAL PCL . . . . .                      | 34 |
| I.8  | Vergleichstabelle der Konvertierungstools . . . . .           | 35 |
| I.9  | Vergleichstabelle der Backendframeworks . . . . .             | 36 |
| I.10 | Vergleichstabelle der Backendframeworks . . . . .             | 36 |
| I.11 | Vergleichstabelle der Frontendframeworks . . . . .            | 38 |
| I.12 | Vergleichstabelle der Kommunikation . . . . .                 | 39 |
| I.13 | Gegenüberstellung der Messgeräte . . . . .                    | 49 |
| I.14 | Vergleichstabelle der Messgeräte . . . . .                    | 50 |
| I.15 | Vergleichstabelle der Messgeräte . . . . .                    | 51 |
| II.1 | Requirements Delivery 1 . . . . .                             | 66 |
| II.2 | Requirements Delivery 2 . . . . .                             | 67 |
| II.3 | Requirements Delivery 3 . . . . .                             | 68 |
| II.4 | Aktoren-Beschreibung . . . . .                                | 73 |
| II.5 | Projektplan Iterationen . . . . .                             | 98 |