

Technische Universität München

Ingenieur fakultät Bau Geo Umwelt

Lehrstuhl für Computergestützte Modellierung und Simulation

Qualitativer und quantitativer Abgleich zwischen 3D-Modellen und Punktwolken im Gerüstbau

Bachelorthesis

für den Bachelor of Science Studiengang Bauingenieurwesen

Autor: Caroline Kerscher

Matrikelnummer:



1. Betreuer: Prof. Dr.-Ing. André Borrmann

2. Betreuer: Alexander Braun

Ausgabedatum: 15. Juli 2020

Abgabedatum: 15. November 2020

Vorwort

Ich möchte an dieser Stelle meinen besonderen Dank an all jene ausdrücken, die zum Gelingen der Arbeit beigetragen haben:

Meinem Betreuer Alexander Braun, für die große Hilfsbereitschaft und die Beratung in allen fachlichen und strukturellen Fragen.

Der Firma Peri AG für die die Bereitstellung der Ressourcen und die finanzielle Unterstützung. Insbesondere dem Team des Digital Transformation Centers in Weißenhorn, das mich jederzeit sehr in allen methodischen, sowie fachlichen Fragen unterstützt hat.

Abstract

Due to numerous trades, subcontractors, and planners, construction projects are tough to manage and monitor. It is necessary to regularly carry out quantitative and qualitative inspections regarding the constantly growing obligation to document and safety regulations. The thesis compares the state-of-the-art approaches for the work with point clouds and 3D building models. It explains the particular process when dealing with point clouds and scaffolding. To find the deviation between the point cloud and the model, the software Autodesk Point Layout and a Python code are tested. With the help of the Python code, which was developed in this thesis, it is possible to investigate different parameters' influence on the result. One most important is the point cloud density. The thesis shows that the 3D model consists of many single areas with a broad spread in their size. This leads to a massive potential for the computation time since a minimal amount of the areas make up a considerable part of the total surface size. Other parameters that influence the quality of the result massively are the threshold for the distance between the model and point cloud and the minimal amount of 3D points per facet.

Zusammenfassung

Es liegt gewissermaßen in der Natur von Bauprojekten, dass deren Abwicklung meist sehr komplex, unübersichtlich und fehleranfällig ist. Dies ist primär auf die enorme Anzahl an Gewerken, Subunternehmern, Planern und den damit verbundenen Teilprozessen zurückzuführen. In Anbetracht der stetig wachsenden Dokumentationspflicht und Sicherheitsvorschriften ist es sinnvoll in regelmäßigen Abständen qualitative und quantitative Kontrollen des Baufortschritts durchzuführen.

Um einen Überblick, über die vielversprechendsten Ansätze zu erhalten werden in dieser Arbeit bestehenden Methoden verglichen und erläutert. Anschließend wird auf Eigenheiten bei der Aufnahme von 3D Punktwolken für Gerüste eingegangen. Diese Punktwolken und die dazugehörigen 3D Modelle werden daraufhin in Autodesk Point Layout und mit einem eigenen Python Code auf Übereinstimmung untersucht. Ein wichtiger Faktor dabei ist der Einfluss der Punktwolkendichte auf die Qualität des Ergebnisses. Die Arbeit stellt fest, dass die Gerüstmodelle eine sehr große Anzahl an detaillierten Teilfläche besitzen, deren Flächeninhalte sehr breit gestreut sind. Dies hat zur Folge, dass eine kleine Anzahl an Flächen einen großen Anteil der Gesamtfläche ausmacht und Berechnungen eine sehr große Kapazität erfordern. Unter Anwendung dieser Erkenntnis kann die Berechnungsgeschwindigkeit enorm erhöht werden. Außerdem wird die Qualität des Ergebnisses maßgeblich von der Toleranzgrenze zwischen Punktwolke und Distanz und von der Mindestanzahl an 3D Punkten je Teilfläche beeinflusst.

Inhaltsverzeichnis

Abbildungsverzeichnis	VII
Abkürzungsverzeichnis	X
1 Einführung	1
1.1 Allgemeine Einführung	1
1.2 Aufbau der Arbeit	4
1.3 Ziel der Arbeit	5
2 State-of-the-art	6
2.1 Fotogrammetrie	6
2.1.1 Erfassung des Objekts	6
2.1.2 Generierung der Punktwolke	8
2.2 Laser basierte Verfahren	11
2.2.1 Erfassung	11
2.3 Vorverarbeitung	15
2.3.1 Vereinheitlichung der Datenstruktur	15
2.3.2 Ausreißer	16
2.3.3 Verdeckungen	17
2.4 Pointcloud-to-BIM	18
2.4.1 Segmentierung	19
2.4.2 Identifizierung der Rand- und Eckpunkte	23
2.4.3 Klassifizierung	23
2.4.4 Überlegung eines auf den Gerüstbau anwendbaren Ansatzes	29
2.5 Point Cloud-vs-BIM	29
2.5.1 Methode 1: Voxelzellen	30
2.5.2 Methode 2: Projektion	31
3 Theoretische Grundlagen	32
3.1 Vorstellung der Firma Peri AG	32
3.2 Allgemeine Eigenschaften von Punktwolken	33
3.3 Einflüsse auf die Qualität der fotogrammetrischen Methodik	33

3.3.1	Ground Sample Distance	33
3.3.2	Kamerasensor.....	34
3.4	Einflüsse auf die Qualität der Laser Methodik.....	34
3.4.1	Innere Parameter	34
3.4.2	Äußere Parameter	35
3.5	Darstellung und Datenformate	37
3.5.1	Datenformat . ifc.....	38
3.5.2	Datenformat .e57	39
3.5.3	Datenformat .las.....	39
3.5.4	Datenformat .pts	40
3.6	Vergleich von Fotogrammetrie und Laser	40
4	Case Study	42
4.1	Zielsetzung	42
4.2	Datenakquise.....	42
4.2.1	Allgemein	43
4.2.2	Auflösung.....	43
4.2.3	Entfernung	43
4.2.4	Oberflächenfarbe des Modells	43
4.2.5	Zugänglichkeit und Verdeckungen.....	43
4.2.6	Registrierung	44
4.2.7	Atmosphäre und Umgebung	44
4.3	Bearbeitung der Punktwolke	45
4.4	Software zum Scan-vs-BIM Abgleich.....	48
4.4.1	Autodesk Point Layout	48
4.4.2	Python.....	55
5	Zusammenfassung und Fazit	66

Anhang

Literaturverzeichnis

Abbildungsverzeichnis

Abbildung 1 Bedeutung der Bauwirtschaft in Deutschland 2017 (Hauptverband der Bauindustrie 2019)	1
Abbildung 2 Digitalisierungsgrad nach Branchen 2014 (Die Welt 2014).....	2
Abbildung 3 Arbeitsproduktivität je Erwerbstätigenstunde (Arbeitsproduktivität je Erwerbstätigenstunde 2018)	3
Abbildung 4 Lebenszyklus eines Gebäudes (Borrmann et al. 2015)	4
Abbildung 5 Typen von Kamerasensoren	7
Abbildung 6 Markierung der korrespondierenden Punkte (Olsson)	9
Abbildung 7 Korrespondierende 3D Punkte werden zueinander in Beziehung gesetzt (Olsson).....	9
Abbildung 8 Prinzip des SfM Algorithmus (Katz 2018)	10
Abbildung 9 Außenseite der Burg, dargestellt durch 3D-Punkte (Olsson)	10
Abbildung 10 Varianten der Registrierung	12
Abbildung 11 Distanzkriterium bei	14
Abbildung 12 Normalenvektor auf Punkt und Flächen (Wang et al.)	14
Abbildung 13 Vereinheitlichung der Datenstruktur und Reduzierung der Datenmenge (Wang et al.).....	16
Abbildung 14 Bereinigte Punktwolke (Malihi et al. 2016)	18
Abbildung 15 Anwendung Farbsegmentierungs Algorithmus bei einem chinesischen Tempel (Zhan et al. 2009).....	19
Abbildung 16 Peri Up Easy mit colorierten Bordbrettern (Softtec 2018)	20
Abbildung 17 Punkte werden den Hauptflächen	21
Abbildung 18 Kanten und versteckte	23
Abbildung 19 Entscheidungsbaum (Yiu 2019).....	24
Abbildung 20 Votum der einzelnen Entscheidungsbäume (Yiu 2019)	24
Abbildung 21 Chronologischer Ablauf der Materialklassifizierung	26
Abbildung 22 Zwischenschritte bei der Extrahierung der vertikalen Rohre aus der Punktwolke	27

Abbildung 23 Extrahiertes Bordbrett	28
Abbildung 24 Gliederung des Modells in Voxel und Raster (Tuttas et al. 2014)	30
Abbildung 25 Projektion der Punktwolke auf das Objekt (Tran und Khoshel-ham 2019)	31
Abbildung 26 Logo der Firma Peri AG	32
Abbildung 27 Übersicht Wellenlängen (Fromme 2014).....	35
Abbildung 28 Schematische Darstellung des Kometenschweif Phänomens (Gordon 2008).....	37
Abbildung 29 Standorte des Laserscanners	44
Abbildung 30 Recap-Einstellung Punktwolkenintensität	45
Abbildung 31 Punktwolke in den Auflösungen 5 mm, 15 mm, 25 mm und 50 mm .	46
Abbildung 32 Einpassen der Punktwolke.....	47
Abbildung 33 Eingepasste Punktwolke mit Modell.....	48
Abbildung 34 Grundeinstellungen bei der Versuchsdurchführung	49
Abbildung 35 Berechnung der Oberflächen der Einzelelemente in Dynamo	50
Abbildung 36 Autodesk Point Layout Release 2021	50
Abbildung 37 Gerüststiel mit Rosette.....	51
Abbildung 38 Histogramm mit Verteilung der Flächeninhalte des Stiels mit Rosette	51
Abbildung 39 3D Modell des gesamten Gerüsts	52
Abbildung 40 Histogramm über die Teilflächeninhalte des gesamten Modells	52
Abbildung 41 Farbscala als Ergebnis der Distanzanalyse	54
Abbildung 42 Liste mir Teilflächen außerhalb der Toleranzgrenze	54
Abbildung 43 Einstellungen für den STL Export in Revit	55
Abbildung 44 Filter für die Reduzierung der STL Flächen in Meshlab	55
Abbildung 45 Einstellungen für die Reduzierung der STL Flächen in Meshlab.....	56
Abbildung 46 Auszug aus einer der verwendeten STL Dateien	56
Abbildung 47 Punktwolkendichte in Relation zur Anzahl der erkannten STL Flächen	60
Abbildung 48 Punktwolkendichte in Relation zur Anzahl der erkannten STL Flächen	61

Abbildung 49 Anzahl an Punkten in Relation zur Punktwolkendichte	61
Abbildung 50 Gesamter Flächeninhalt des STL Meshes in Abhängigkeit zur Punktwolkendichte	62
Abbildung 51 Häufigkeit des Auftretens an Punkten pro Facet bei einer Toleranz von 50 mm	62
Abbildung 52 Häufigkeit des Auftretens an Punkten pro Facet bei einer Toleranz von 100 mm	63
Abbildung 53 Anzahl der erkannten STL Flächen in Bezug zur minimalen Anzahl an Punkten je STL.....	64
Abbildung 54 Gerüstmodell in Abhängigkeit zur Mindestanzahl an Messpunkten je STL.....	64

Abkürzungsverzeichnis

BIP	Bruttoinlandsprodukt
dm ²	Quadratdezimeter
BIM	Building Information Modelling
cm	Zentimeter
m	Meter
mm	Millimeter
RFA	Random Forest Algorithm
IFC	Industry Foundation Classes
UAV	Unmanned Aerial Vehicle
KNN	Künstliche Neuronale Netze
ASTM	American Society for Testing and Materials
ASPRS	American Society for Photogrammetry and Remote Sensing
ASCII	American Standard Code for Information Interchange
MSAC	M-estimator SAmple Concensus
STL	Stereolithografie

1 Einführung

1.1 Allgemeine Einführung

Die Baubranche stellt einen Kernbereich der deutschen Wirtschaft dar und wird oft als Regulierungsorgan der Konjunktur durch den Staat verwendet. Wie in **Abbildung 1** zu erkennen ist beträgt das Wirtschaftsvolumen der gesamten deutschen Baubranche circa 10 % des BIP und sie beschäftigt fast 6 % aller Erwerbstätigen. Somit zählt sie noch vor dem Fahrzeugbau und der Chemischen Industrie zu den Schlüsselbranchen der deutschen Wirtschaft. (Die deutsche Bauindustrie 2018)

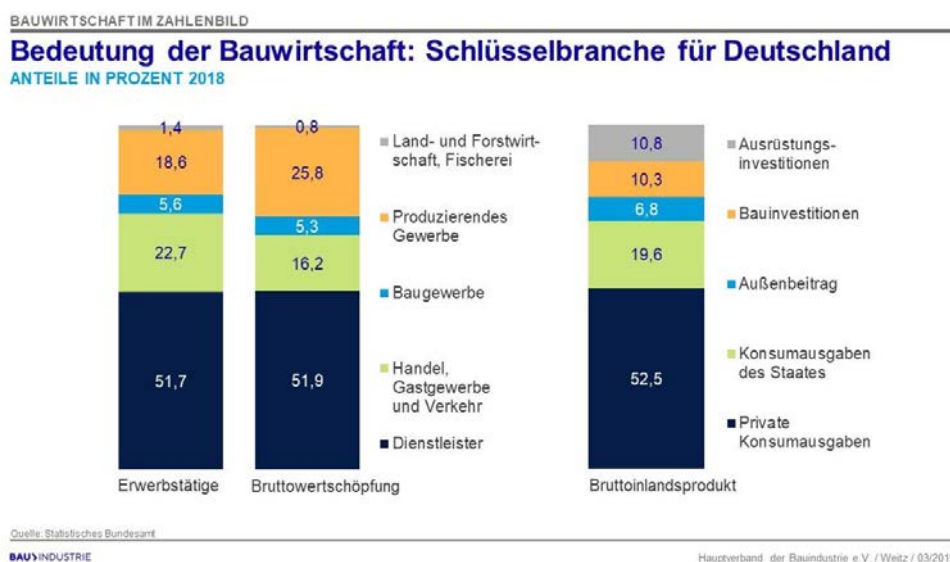
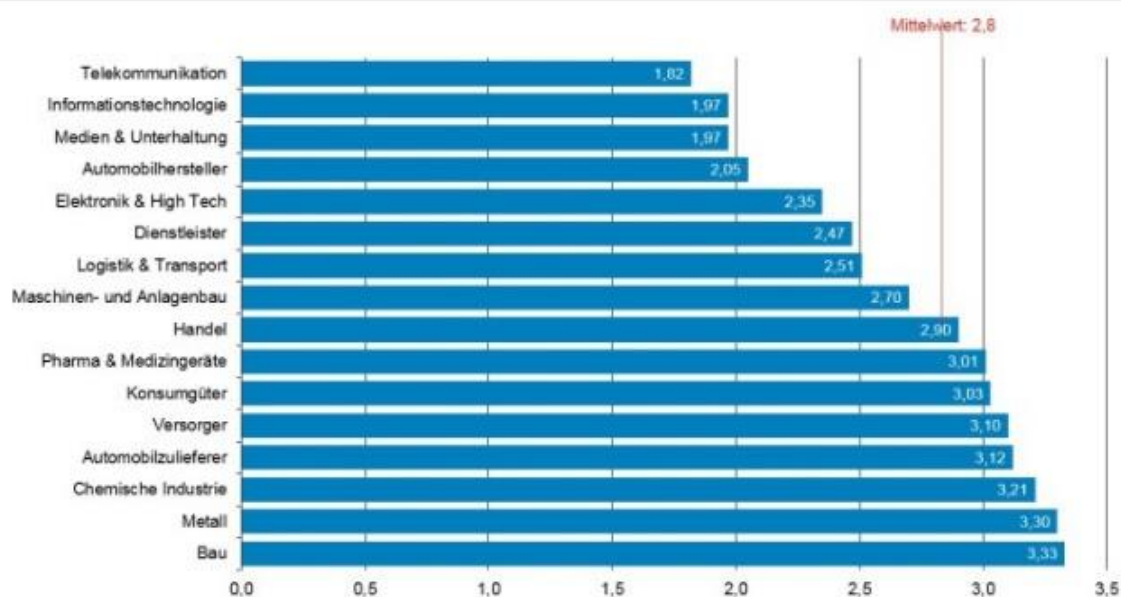


Abbildung 1 Bedeutung der Bauwirtschaft in Deutschland 2017 (Hauptverband der Bauindustrie 2019)

Dabei ist es fast erschreckend, dass die Bauindustrie wie in **Abbildung 2** dargestellt das Schlusslicht im Bereich Digitalisierung bildet. (Die Welt 2014) Denn andere Branchen mit viel geringerem Anteil am Bruttoinlandsprodukt sind bedeutend besser entwickelt. (Hauptverband der Bauindustrie 2019)

Digitalisierungsgrad nach Branchen, 2014



Bewertungsskala: 1 = größtenteils, 2 = teilweise, 3 = wenig, 4 = ansatzweise digitalisiert

Abbildung 2 Digitalisierungsgrad nach Branchen 2014 (Die Welt 2014)

Erst einmal scheint dieser Rückstand, besonders wegen der überwiegend lokalen Beschränktheit des Baumarkts, nicht besonders gravierend. Denn in der Regel konkurrieren vor allem deutsche Unternehmen untereinander um inländische Bauaufträge. Doch durch viele Wachstumsimpulse besteht eine große Nachfrage an zusätzlichem Bauvolumen. Dazu zählen der anhaltende Trend zur Urbanisierung, Migration, energetische Sanierungen, demografischer Wandel und der Bundesverkehrswegeplan 2030. Letzterer trägt unter anderem zur langfristigen Wettbewerbsfähigkeit Deutschlands bei. Demgegenüber steht eine fast gleichbleibende Zahl an Handwerkern und Facharbeitern mit einer ebenfalls seit 25 Jahren stagnierenden Arbeitsproduktivität je Erwerbstätigenstunde. (siehe **Abbildung 3**) (BAUMANN et al. 2016; Die Welt 2014)

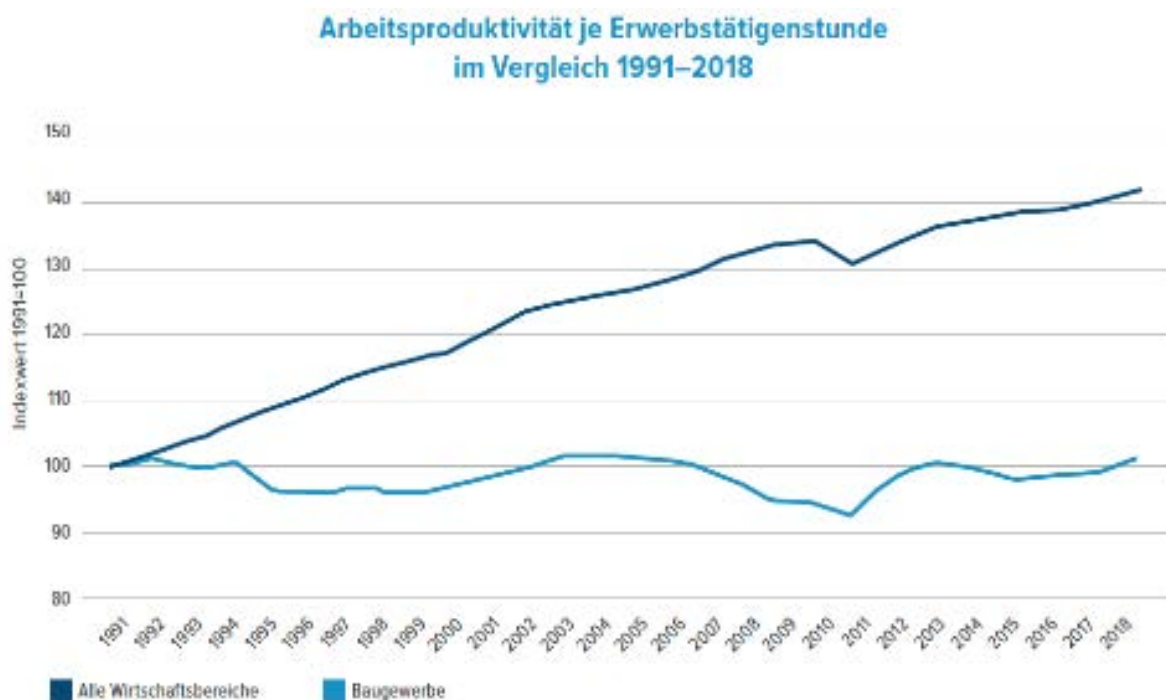


Abbildung 3 Arbeitsproduktivität je Erwerbstätigenstunde (Arbeitsproduktivität je Erwerbstätigenstunde 2018)

In Anbetracht der momentan hohen Nachfrage nach Bauleistungen und der beschränkten Kapazität bietet die Digitalisierung der Bauunternehmen eine gute Möglichkeit, um die eigene Effizienz zu steigern. Das größte Einsparpotential liegt im Bereich der Planung und Bauüberwachung. Denn ca. 10 % der Kosten im Baugewerbe sind auf Fehler zurückzuführen. Somit würde sich durch eine umfassende Informationsverarbeitung die Möglichkeit eröffnen den gesamten Bauprozess zu verkürzen und die Margen für die Unternehmen zu erhöhen. (KOCIJAN 2018)

Im Jahr 2010 waren 32 % des Energieverbrauchs und 19 % aller Treibhausgasemissionen auf Gebäude zurückzuführen. Dies ist vor allem durch die aufwendigen Baumaßnahmen, Materialien und den Betrieb bedingt. (University of Cambridge) Dabei wäre es in vielen Fällen leicht möglich die Gebäude mit geringem Aufwand energetisch zu sanieren oder einen Abriss durch eine Umnutzung zu vermeiden. Doch in vielen Fällen scheitert diese Herangehensweise an fehlenden Informationen und Plänen. Hierbei könnte die Betrachtung der gesamten Lebenszyklusphasen durch die BIM Methodik hilfreich sein, denn momentan ist der überwiegende Teil des gesellschaftlichen und politischen Drucks noch auf die Automobil- und Energiebranche gerichtet. Doch falls dieser Trend eines Tages umschlägt wird auch die Baubranche vermehrt mit Sanktionen durch den Staat und Gesellschaft zu rechnen haben.

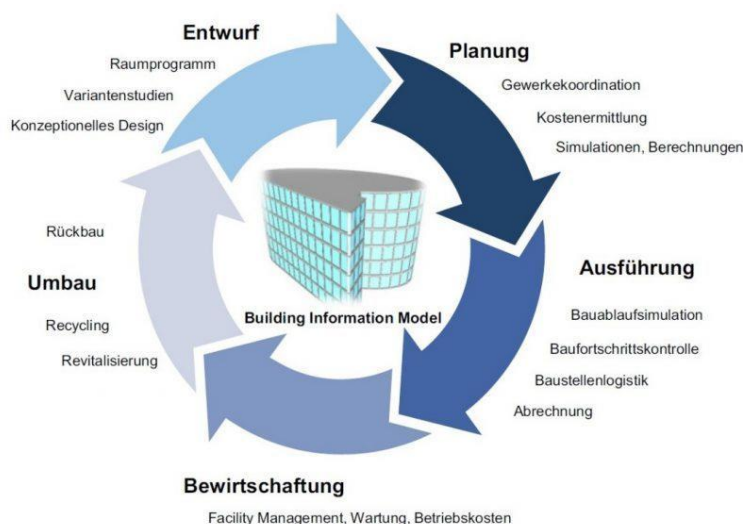


Abbildung 4 Lebenszyklus eines Gebäudes (Borrmann et al. 2015)

In dieser Arbeit wird die Optimierung der Planung am Beispiel des Gerüstbau Handwerks untersucht. Der folgende Abschnitt informiert über den Aufbau der Arbeit.

1.2 Aufbau der Arbeit

Der State-of-the-Art Teil der Arbeit beschreibt die aktuellsten Algorithmen und Vorgehensweisen für die qualitative beziehungsweise quantitative Baufortschrittskontrolle. Des Weiteren wird auf die Extrahierung eines BIM Modells aus einer Punktwolke eingegangen. Die zwei dominierenden Verfahren zur Datenerfassung von Punktwolken sind Laserscanning und Fotogrammetrie, welche analog in Kapitel 2.2 und 2.3 beschrieben werden.

Abschnitt 4 befasst sich mit der Vorverarbeitung der Punktwolken, wobei er insbesondere auf die Reduzierung und Systematisierung der Punktwolkendaten eingeht. Ab diesem Punkt ist das Vorgehen unabhängig vom Aufnahmeverfahren. Im Kapitel 2.5 wird genauer auf die aktuellen Verfahren eingegangen, um aus einer Punktwolke ein dreidimensionales BIM Modell zu extrahieren. Kapitel 2.6 beschreibt den Vorgang ein bestehendes 4D-BIM Modell mit einem realen Aufbau zu vergleichen.

In Kapitel 3 soll zunächst die Firma PERI mit ihrer Rolle als einer der wesentlichen Innovationstreiber in der Bauzulieferbranche vorgestellt werden. Im Speziellen wird auf die Funktion des eigens eingerichteten Digital Competence Centers eingegangen, dass sich überwiegend mit Fragestellungen zur Anwendung und Weiterentwicklungen im Bereich Building Information Modeling beschäftigt. Anschließend werden allgemein

die Eigenschaften von Punktwolken dargestellt. Des Weiteren werden die Randbedingungen erläutert, die bei der Datenerfassung mit Fotogrammetrie beziehungsweise Laser beachtet werden müssen. Anschließend wird die Datenakquise mit Fotogrammetrie und Laser gegenübergestellt und die jeweiligen Vor- und Nachteile erarbeitet.

Im Rahmen des Kapitels 4 wird die Case Study beschrieben. Dafür wird zuerst die Zielsetzung erklärt und anschließend die Umsetzung mit Datenakquise und Test der verschiedenen Programme. Dabei wird genauer auf die Vor- und Nachteile von Autodesk Point Layout eingegangen. Anschließend die Funktionsweise des für diese Arbeit entwickelte Python Code erläutert und statistisch ausgewertet.

1.3 Ziel der Arbeit

Die vorliegende Arbeit hat das Ziel einen Überblick über die aktuell bevorzugten Ansätze zur Baufortschrittskontrolle und BIM Modellextraktion aus einer Punktwolke zu geben. Im Spezielleren wird diese Aufgabenstellung für die Teildisziplin des Gerüstbaus betrachtet, wobei es durch den hohen Detailgrad der Konstruktion besondere Herausforderungen gibt. Ziel dabei ist eine Methode zum qualitativen und quantitativen Abgleich zwischen Punktwolke und Gerüstmodell zu finden. Die Methode soll dabei vor allem auf die Anforderungen aus der Planung und den Baustellengegebenheiten eingehen. Beispiele dafür sind Toleranzgrenzen und Beschaffenheit der Punktwolke. Dafür werden Versuche mit bereits bestehenden Software Programmen evaluiert und anschließend ein eigenes Programm in Python entwickelt.

2 State-of-the-art

2.1 Fotogrammetrie

Allgemein ist die Zielsetzung der fotogrammetrischen Bearbeitung die räumliche Lage von zwei oder mehreren Bildern zueinander wiederherzustellen. Maßgeblich dafür ist die Zentralprojektion unter Beachtung der Komplanaritätsbedingungen. (Wikipedia-Autoren 2019) Im Anwendungsgebiet der Baubranche kommt vor allem die Nahbereichsfotogrammetrie zum Einsatz, wobei aus singulären 2D-Aufnahmen ein 3D-Objekt rekonstruiert wird. (Autodesk 2019) Im Anschluss daran werden geometrische Maße aus dem Modell abgeleitet. Der folgende Abschnitt befasst sich mit der Erfassung der Bilddaten und der anschließenden Generierung der Punktwolke.

2.1.1 Erfassung des Objekts

Bei der fotogrammetrischen Methode werden durch einen Kamerasensor 2D-Bilder aufgenommen, die durch geschickte Transformation ein Modell mit 3D Informationen ergeben. Um dies zu erreichen werden von jeder statischen Szene mindestens drei Aufnahmen aus unterschiedlichen Blickwinkeln benötigt, um ein robustes Ergebnis zu erhalten. Daraus ergibt sich die stereotypische Ansicht auf das Modell und die Möglichkeit zur Ableitung von 3D Informationen in den sich überlappenden Bildbereichen. (Faltýnová et al. 2016)

Generell gibt es verschiedene Geräte zur Aufnahme, über die die folgende Grafik einen Überblick gibt:

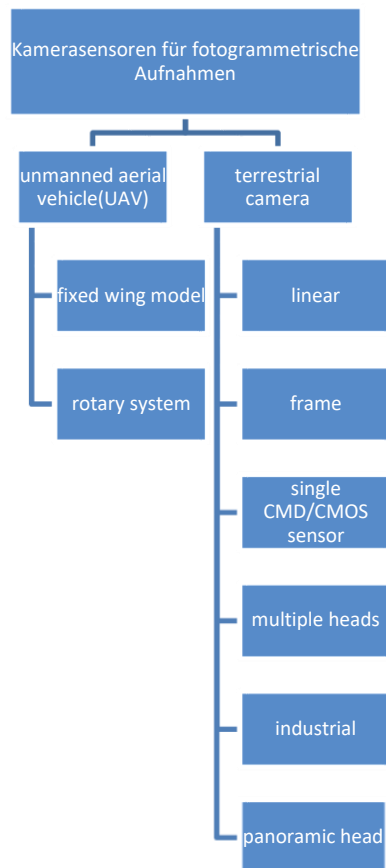


Abbildung 5 Typen von Kamerasensoren

UAV eignen sich vor allem bei großen Gebäuden, bei denen auch das Dach erfasst werden muss. Für diese Art der Aufnahme benötigt man neben dem Kamerasensor eine Drohne und eine Bodenkontrollstation. Die Bodenkontrollstation dient dazu die Drohne aus der Ferne zu steuern, wofür beispielsweise ein Laptop verwendet werden kann (Lliaifar 2019).

Datenformat

Das Datenformat hängt maßgeblich vom verwendeten Sensor ab. Die gängigsten Typen sind dabei die 3D-Punktwolke, 3D Mesh und das Voxelgitter. Voxelgitter haben den Vorteil, dass sie wenig Rechenkapazität beanspruchen und sich gut für die Prozessierung mit Künstlichen Neuronalen Netzen (KNN) eignen. Andererseits haben sie aber einen geringen Detailgrad. Bei KNN handelt es sich um eine dem Gehirn nachempfundene Struktur, die sich für Künstliche Intelligenz und Maschinelles Lernen einsetzen lässt. Nach der Konstruktion eines neuronalen Netzes folgt eine Trainingsphase, die auf der Variation der Gewichtungen der einzelnen Knoten basiert. Dabei

lernt das KNN und verändert die Verbindungen, Platzierung der Knoten oder passt die Schwellenwerte an. (Luber und Litzel 2018)

Meshes haben im Gegensatz zu den Punktwolken eine geschlossene Oberfläche und bilden deren Struktur ab. So ist es möglich die Normalen der Oberfläche zu bestimmen und auch die Umgebung fotorealistisch darzustellen. (Runceanu und Haala 2018) Da mit den Aufnahmen jedoch weitere Berechnungen und Ableitungen durchgeführt werden müssen würde sich in unserem Fall nur die 3D-Punktwolke eignen.

2.1.2 Generierung der Punktwolke

Nach der Erfassung der Daten besteht die Generierung von Punktwolken aus drei Schritten: Orientierung der Bilder, Zusammenfügen der Bilder und Co-Registrierung. (Tuttas et al. 2014) Die gängigsten Ansätze dafür sind Multi Stereo Matching und Structure from Motion (SfM). Letzterer zählt zur sogenannten Mehrbildfotogrammetrie und wird im Folgenden am Beispiel der Burg Örebro Castle schrittweise erläutert.

1. Am Beginn des Prozesses werden sehr viele 2D-Aufnahmen vom Objekt gemacht, wobei sich die benachbarten Fotos bestenfalls um 60% überlappen sollten. (Olsson) Des Weiteren sollte zwischen den verschiedenen Kamerastandpunkten eine gewisse Distanz liegen, um die Genauigkeit des späteren Modells nicht zu gefährden. Um dies zu gewährleisten und die genauen Standpunkte für die spätere Berechnung vorzumerken werden die Positionen durch eine Totalstation oder ein GNSS aufgezeichnet. (Faltýnová et al. 2016) Die Fotos sollten so aufgenommen werden, dass zuerst eine Übersichtsaufnahme vom gesamten Objekt angefertigt wird und dann sukzessive alle Verdeckungen durch Nahaufnahmen eliminiert werden. Die Nahaufnahmen können auch gerne durch Heranzoomen des Objekts erstellt werden. Um den späteren Rechenaufwand so gering wie möglich zu halten, ist es von Vorteil, wenn bei der Kamera nicht zwischen Hoch- und Querformat gewechselt wird. Generell wirkt es sich günstig auf das Ergebnis aus, wenn während der Aufnahme nur mäßige Sonneneinstrahlung vorherrscht.
2. Im Rahmen der Bildtriangulation werden in den Fotos diejenigen korrespondierenden Punkte identifiziert, die von mehreren Positionen aus in den sich überlappenden Bildbereichen erkennbar sind. Die Auswahl der Punkte kann entweder händisch oder mit einem Algorithmus zur Mustererkennung erfolgen. Das

dient als Grundlage um einzelne Bilder miteinander verknüpfen zu können. Beispiele hierfür sind Ecken oder Kanten. Der Teilschritt ist in **Abbildung 6** dargestellt. (Olsson)



Abbildung 6 Markierung der korrespondierenden Punkte (Olsson)

3. Im dritten Schritt werden die korrespondierenden 3D-Punkte der benachbarten Bilder identifiziert und eine Beziehung hergestellt. Dieser Prozess ist in **Abbildung 7** durch die grünen Linien dargestellt. (Olsson)

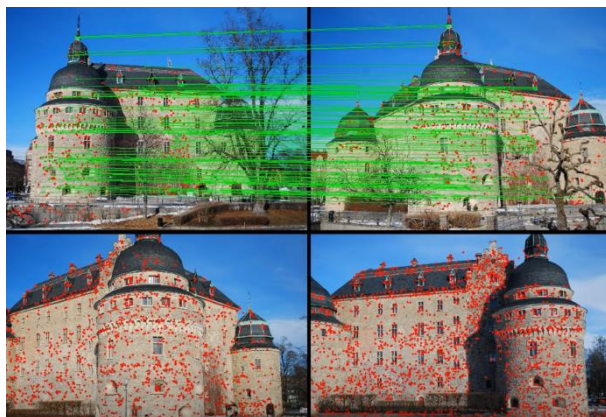
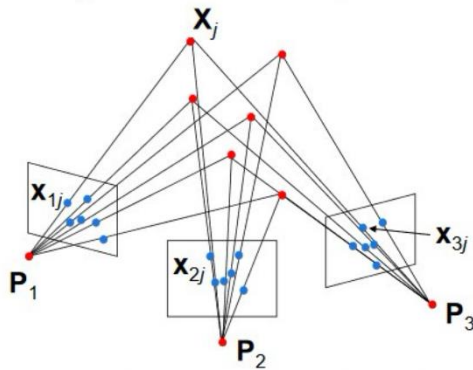


Abbildung 7 Korrespondierende 3D Punkte werden zueinander in Beziehung gesetzt (Olsson)

4. Anschließend werden die ursprünglichen Kamerapositionen und 3D-Punkte so ermittelt, dass sich die korrespondierenden Blickwinkel überschneiden. (Olsson) Der zugrundeliegende Algorithmus wird als Bündelblockalgorithmus bezeichnet, da von jedem Bildpunkt (X_j) aus, ein Strahl zum Projektionszentrum (Kamera) gezogen, und dort „gebündelt“ wird. So kann ein Zusammenhang zwischen Objekt- und Bildkoordinaten hergestellt werden. (Folke 2001) Da es sich wie in **Abbildung 8** dargestellt, um verschiedene Kamerastandorte (P_1 ,



P_2 , P_3) handelt, spricht man vom sogenannten Streifenlichtscanning (vom Englischen Structure from Motion). Um am Ende das gesamte Modell zusammenzufügen, werden die Bilder extern orientiert und die Überlappungen entfernt. (Braun et al.

2015)

Abbildung 8 Prinzip des SfM Algorithmus (Katz 2018)

5. In dem beschriebenen Beispielfall entstand ein 3D-Modell der Burg aus insgesamt 671 Aufnahmen. Daraus wurden circa 5.000.000 3D-Punkte ermittelt, mit denen im Anschluss die Oberfläche nachmodelliert wurde. (Olsson) Der gelbe Polygonzug im Umkreis der Burg in **Abbildung 9** markiert die einzelnen Standorte der Kamera.

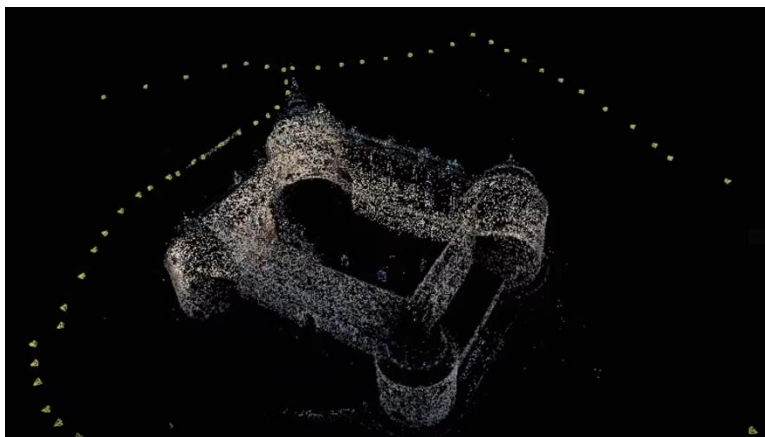


Abbildung 9 Außenseite der Burg, dargestellt durch 3D-Punkte (Olsson)

2.2 Laser basierte Verfahren

Laser Scanner unterscheiden sich grundsätzlich nach Scan Radius, Scan Geschwindigkeit, Genauigkeit und Handhabbarkeit. Meistens arbeiten sie nach einem ähnlichen Verfahren: Die Station emittiert in hoher Geschwindigkeit pulsierende oder kontinuierliche Lichtstrahlen in Richtung der Fläche, die aufgenommen werden soll. Indem der Lichtstrahl in horizontaler sowie vertikaler Richtung geführt wird entsteht ein symmetrisches Bild von der gesamten Umgebung. Wenn der Strahl dabei auf ein Objekt im Umfeld trifft wird er zurück zur Laserstation reflektiert. Falls das ankommende Signal stark genug ist wird der Punkt erkannt und mit Hilfe eines Timers oder Phasenvergleichsverfahren die Distanz zwischen Objekt und Gerät ermittelt. Für jede einzelne Distanzmessung wird der horizontale und vertikale Winkel des rotierenden Lasers ermittelt um die X-, Y- und Z-Koordinaten der Punkte festzustellen. (Renovation of Heritage Assets using BIM: a Case Study of the Durham Cathedral 2015) Als Resultat entsteht innerhalb weniger Minuten ein Datensatz mit Millionen von sehr dichten, netzförmig angeordneten 3D Punkten. Im alltäglichen Sprachgebrauch wird diese Datenstruktur als Punktwolke bezeichnet. (SHEN Z. et al. 2013)

2.2.1 Erfassung

Laseraufstellpunkte

Die Standorte sollten so gewählt werden, dass ein möglichst großer Teil des Modells erfasst wird, ohne zu viele Redundanzen zu erzeugen. Dabei muss vor allem auf die Winkel geachtet werden um auch Okklusionen zu erfassen. Für eine reibungslose Georeferenzierung sollte an den Schnittstellen zwischen zwei Standorten eine Überlappung von circa 20 % angestrebt werden. Das optimale Ergebnis wird durch einen geschlossenen Polygonzug der Aufstellpunkte erlangt. Dabei sollte jeder neue Standpunkt in direkter Sichtlinie des vorhergehenden liegen. Bei der Wahl der Standfläche sollte unbedingt auf eine ebene Fläche ohne Vibration oder Bewegungen geachtet werden, da sich sonst erhebliche Messungenauigkeiten ergeben. Dies kann vor allem bei laufendem Baustellenbetrieb mit Erdarbeiten und schwerem Gerät eine Herausforderung darstellen.

Registrierung

Unter Registrierung versteht man die Transformation der Koordinatensysteme der einzelnen Standpunkte in ein übergeordnetes Koordinatensystem. Bei dem Abgleich von Scan und Modell muss auch das Modell in das übergeordnete Koordinatensystem eingepasst werden.

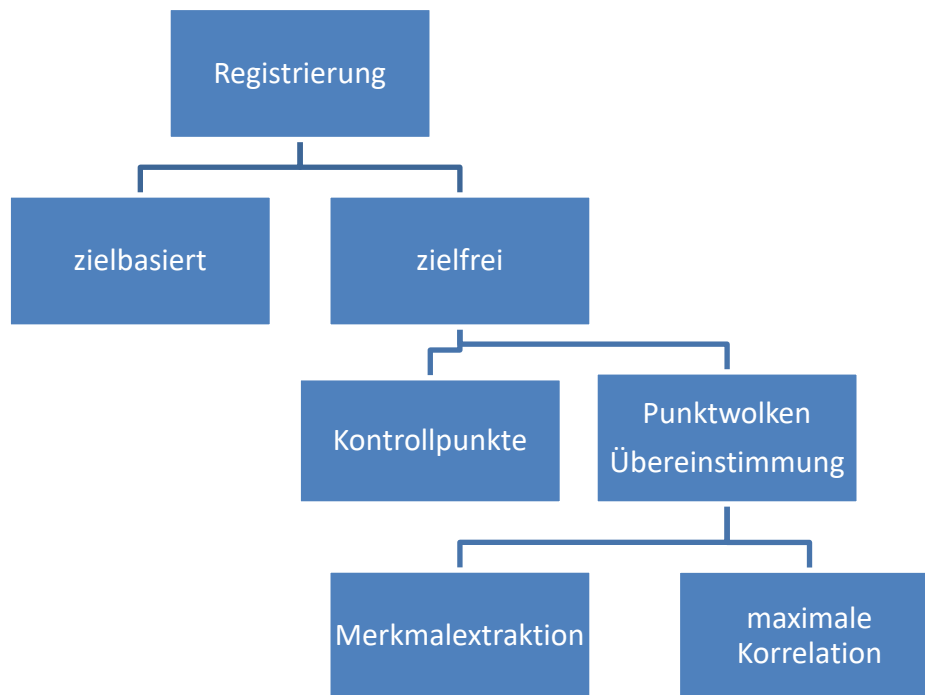


Abbildung 10 Varianten der Registrierung

Dies ist notwendig, da jedem Aufstellort des Scanners ein separates Koordinatensystem zugewiesen wird. Voraussetzung für die Zusammenstellung ist, dass die einzelnen Koordinatensysteme konsistent sind. Im Anschluss an die Definition der einzelnen Koordinatensysteme wird jedem gescannten Punkt ein Wert auf den drei Achsen zugewiesen. Bei der Registrierung handelt es sich um einen kritischen und fehleranfälligen Schritt, von dem die Genauigkeit des späteren Modells abhängt. Da ein Modell fast nie durch nur einen Blickwinkel erfasst werden kann, ist die Registrierung Bestandteil von nahezu jedem Scan.

Um den Rechenaufwand zu optimieren werden die Scans und Modelle erst grob registriert und dann die einzelnen Objekte fein nachjustiert. Dieser Prozess besteht im Allgemeinen aus zwei Schritten: Zuerst werden der Scan und das Modell grob angeordnet und anschließend die einzelnen Bauteile fein angepasst. Dabei unterscheidet man wie in **Abbildung 10** dargestellt generell in eine zielbasierte und zielfreie Registrierung. Die Eigenheiten der verschiedenen Methoden werden im Folgenden erläutert.

Grobe Registrierung

Um ein gutes Ergebnis zu erhalten sollte die grobe Registrierung jedoch nicht zu sehr vom endgültigen Ergebnis abweichen. Um dies zu erreichen gibt es verschiedene Methoden wobei im Folgenden das grobe Vorgehen bei der Registrierung von zwei Scans und anschließend die von einem Scan und Modell beschrieben wird.

Scan-Scan:

Eine der drei Methoden für die grobe Scan-Scan Registrierung ist die manuelle Selektion von Punkten, welche zu den nicht zielbasierten Verfahren zählt. Dafür müssen drei, meist visuell hervorstechende Punkte wie Ecken, ausgewählt werden. Die gewählten Punkte sollten dabei bevorzugt einen maximalen Abstand und unterschiedliche Höhen zueinander haben. Aufgrund der enormen Größe der Punktwolken gestaltet sich die Auswahl oft als schwierig und sehr fehleranfällig.

Bei der zielbasierten Registrierung werden im Aufnahmebereich mindestens drei Zielobjekte platziert, die in jedem Scan sichtbar sind. Die Zielobjekte sollten dabei leicht zu erkennen und zusammenführbar sein. Anschließend werden die Koordinaten der Zielzentren errechnet und die Scans zusammengeführt. Hierfür gibt es manuelle sowie automatische Ansätze.

Ziel der Objektbasierten Registrierung ist es 3D Objekte in beiden Scans zu identifizieren, die automatisch und zuverlässig überlappt werden können. Vorteilhaft bei dieser Methode ist, dass keine Zielpunkte installiert werden müssen. Diese Vorgehensweise eignet sich jedoch nur bei Objekten mit viel Struktur ohne sich ständig wiederholende Muster. Desweiteren sollten die Punktwolken keine zu großes Rauschen aufweisen.

Scan-Modell:

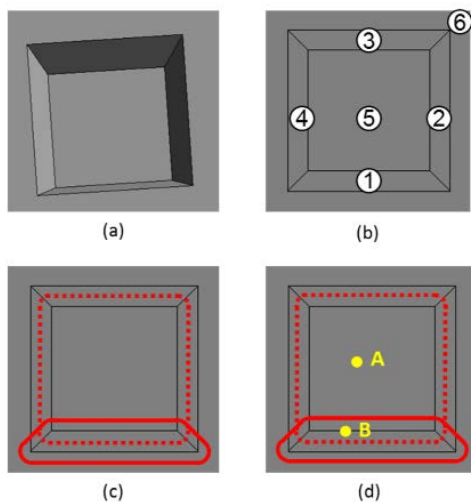
Bei der Registrierung von Scan und Modell ist die zielbasierte Registrierung eher unpraktisch und die manuelle Auswahl von Punkten und die objektbasierte Methode vorzuziehen. Bei der manuellen Auswahl von Punkten werden übereinstimmende Punkte im Modell und in der Punktwolke gesucht. Wie bei der Registrierung von zwei Scans ist dieses Vorgehen jedoch oft nicht akkurat genug. Diese Tatsachen lassen darauf schließen, dass der objektbasierte Ansatz zu bevorzugen ist. Dafür werden sowohl das Modell als auch die Punktwolke in Teilflächen mit homogener Form zerlegt. Diese werden im Anschluss wieder zusammengefügt und die Form von Scan und Modell auf

Ähnlichkeit getestet. Die Flächen haben gegenüber den Punkten den Vorteil, dass sie auch bei Verdeckungen zumindest teilweise sichtbar sind. (Bosche 2011)

Feine Registrierung

Ziel der feinen Registrierung ist es alle Punkte der Punktwolke einer Fläche des Modells zuzuordnen. Für eine eindeutige Zuordnung gibt es zwei Kriterien

- 1) Abstand zur Fläche
- 2) Richtung der Normalen



Um den zulässigen Abstand zu einer Fläche festzulegen wird wie in

Abbildung 11 c) dargestellt um jede einzelne Fläche ein Umkreis definiert. Wenn ein Punkt innerhalb dieses Umkreises liegt (A) erfüllt er das erste Kriterium. Er kann aber auch wie der Punkt B in

Abbildung 11 d) zu sehen innerhalb von zwei Umkreisen gleichzeitig liegen. In diesem Fall wird dann das zweite Kriterium benötigt.

Abbildung 11 Distanzkriterium bei

der feinen Registrierung (Wang et al.)

Kriterium Nummer zwei ist der Unterschied zwischen dem Normalenvektor des Punktes und den Normalenvektoren der Flächen. Der Normalenvektor des Punktes wird wie in **Abbildung 12** dargestellt durch die Methode der kleinsten Quadrate konstruiert. Dafür wird aus dem Punkt und seinen acht angrenzenden Nachbarpunkten eine Ebene geschaffen auf die der Normalenvektor konstruiert wird. Anschließend wird der Winkel zwischen dem Normalenvektor des Punktes und den Normalenvektoren der Flächen berechnet. Falls der Winkel ein gewisses Maß nicht überschreitet wird er zu der jeweiligen Fläche gezählt.

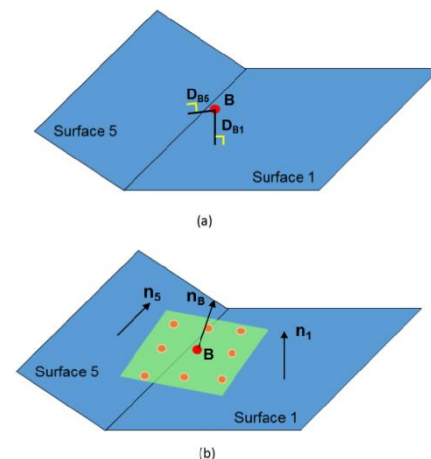


Abbildung 12 Normalenvektor auf Punkt und Flächen (Wang et al.)

Mit Hilfe dieses Doppelkriteriums kann ein genaueres Ergebnis bei der feinen Registrierung erzielt werden. (Wang et al.)

2.3 Vorverarbeitung

Ziel der Vorverarbeitung der Punktwolke ist einerseits die Standardisierung der Datenstruktur und andererseits die Verfeinerung durch Beseitigung von Ausreißern durch Störquellen bei der Aufnahme.

2.3.1 Vereinheitlichung der Datenstruktur

Bei einer Punktwolke handelt es sich um eine große Anzahl an Punkten in einem Koordinatensystem, die die äußere Oberfläche eines Objekts annähern. Dieses Konsortium enthält sehr detaillierte räumliche Informationen über das reale Objekt. Die einzelnen Messpunkte enthalten ausschließlich die Information über ihre Position im kartesischen x-y-z Koordinatensystem, weshalb sie sich nur bedingt für die weitere Verarbeitung eignen und einen hohen Rechenaufwand erfordern. Zur Modellierung und weiteren Ableitung bedarf es einer simpleren Datenstruktur. Eine Möglichkeit dafür ist die Umwandlung in ein prismatisches oder polygonales Modell. Dabei wird jedoch der Detaillierungsgrad drastisch reduziert. (Faltýnová et al. 2016)

Eine Alternative dazu ist die Implementierung eines, wie in Abbildung 15 a) dargestellten, Voxel Gitters. Dabei wird über die gesamte Punktwolke ein einheitliches 3D-Gitter gelegt. Jede Voxeleinheit enthält je nach Dimensionierung eine bestimmte Anzahl an Scan Punkten, welche gelöscht und durch einen einzigen Punkt im Schwerpunkt ersetzt werden. Dieser Vorgang ist durch die roten und blauen Punkte in **Abbildung 13 b) und c)** dargestellt. Die Farbwerte innerhalb einer Voxeleinheit werden dabei interpoliert. Je größer also die Abmessungen eines Voxels, desto mehr Punkte können eliminiert werden. Auf der Kehrseite sinkt jedoch die Genauigkeit des Modells. Als Resultat erhält man eine stark verringerte Datenmenge mit diskreten Punkten. (Xu et al. 2015) (Wang et al. 2015)

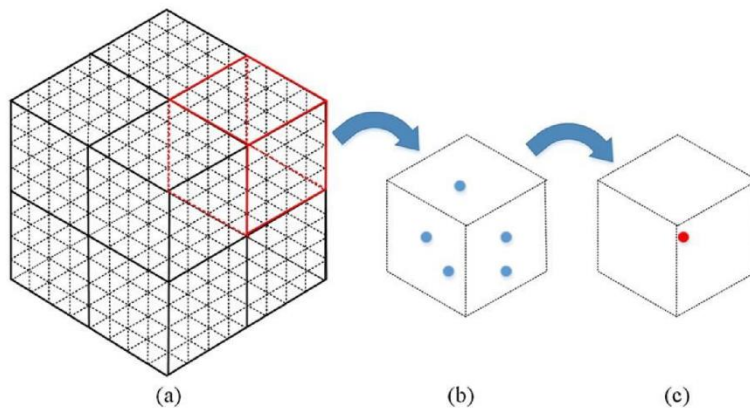


Abbildung 13 Vereinheitlichung der Datenstruktur und Reduzierung der Datenmenge (Wang et al.)

Für eine effizientere Speicherung der Daten werden die Daten in eine Baumstruktur gebracht. (Pațrațucean et al. 2015)

2.3.2 Ausreißer

Da die Aufnahmebedingungen, vor allem im Freien, oft nicht optimal sind, enthalten die Scans oft ein Rauschen in den Daten. Um Fehler und Ungenauigkeiten zu vermeiden muss dieses Rauschen reduziert werden. Unter einem Rauschen sind vor allem einzelne isolierte Punkte zu verstehen, die identifiziert und entfernt werden müssen. (Wang et al. 2015) Dafür können verschiedene Verfahren angewendet werden, wobei im Folgenden der Gauß Algorithmus und der Tensoren Abstimmungsalgorithmus vorgestellt werden.

Um die verstreuten Ausreißer zu finden wird der Gauß Algorithmus angewandt. Dazu wird von jedem Punkt aus, die Distanz zu all seinen Nachbarpunkten gemessen und ein globales Mittel errechnet. Die resultierenden Distanzen sind standard-verteilt nach Gauß. Im Anschluss wählt man ein Intervall um den Mittelwert herum um die Grenze für die Ausreißer zu definieren. Alle Punkte die außerhalb des Intervalls liegen werden rausgefiltert.

Bei Tensoren Abstimmungsverfahren handelt es sich um einen nicht-parametrischen Algorithmus. Diese Algorithmusart ist eine Untergruppe der sogenannten Klassierungsverfahren und eignet sich besonders bei großen Datenmengen, über die man keine Vorannahmen treffen kann. (Brownlee 2016) Hierbei besteht der Vorgang aus zwei Teilschritten: die Tensoren Kodierung und die Tensoren Bewertung. Das Ziel des

Tensor Abstimmungsverfahren ist die Struktur des Modells in der Punktwolke zu identifizieren. Dabei werden für jeden 3D Punkt Tensoren kodiert und in der Bewertungsphase wird diese Information an die Nachbarpunkte weitergeleitet. Dies geschieht innerhalb eines vordefinierten Vektorfeldes. Jeder Gitterpunkt sammelt die Wahlergebnisse seiner Nachbarpunkte und bildet daraus einen neuen Tensor. Im Anschluss entfernt man die Ausreißer, indem die Punkte mit den niedrigsten Tensorwerten gelöscht werden. (Wang et al. 2015)

2.3.3 Verdeckungen

Ein weiterer Faktor, der zu Ungenauigkeiten führt, sind Verdeckungen der Oberfläche durch Hilfskonstruktionen wie Gerüste. Ein Beispiel einer bereinigten Punktwolke ist in

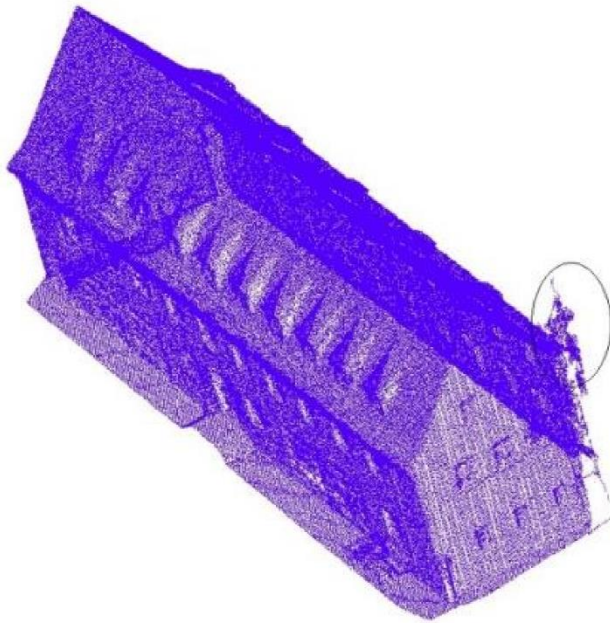


Abbildung 14 zu betrachten, wobei das verdeckende Gerüst durch den Kreis gekennzeichnet ist. Problematisch dabei ist, dass die Abmessungen hinter der Verdeckung nur abgeschätzt werden können. Abhängig vom Detailgrad des Modells werden deshalb die Punkte den Hauptflächen des Objektes zugeordnet. Die sich daraus ergebende spezifische Flächenform erhält man durch eine Berechnung mit dem RANSAC Algorithmus. Ob eine neue Fläche erzeugt wird oder nicht wird anhand der Menge der darauf projizierten Punkte entschieden. (Malihi et al. 2016)

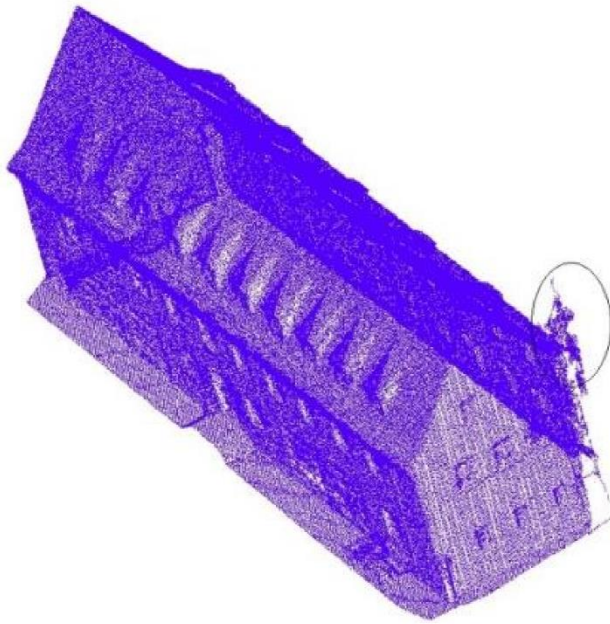


Abbildung 14 Bereinigte Punktwolke (Malihi et al. 2016)

2.4 Pointcloud-to-BIM

Laut der offiziellen Definition handelt es sich bei einem Modell um eine vereinfachte Abbildung der Wirklichkeit. So kann auch ein as-built BIM Modell das Bauwerk nie vollständig korrekt abbilden, sondern richtet sich immer nach dem vom späteren Nutzer geforderten Informationen. Generell kann man die daraus ableitbaren Informationen in die Bereiche Geometrie, Semantik und Attribute einteilen. (Pațrațucean et al. 2015) Daraus kann man folgern, dass auch das BIM Modell eines Gebäudes keine 100 prozentige Abbildung darstellt. Aus diesen Gründen muss vor der Aufnahme des Gebäudes festgelegt werden, welchen Detailgrad und Perspektiven dargestellt werden sollen.

Aktuell werden in der Praxis vor allem semi-automatische Ansätze angewandt, um Bauteile aus einem Punktwolke zu extrahieren. Beispiele dafür, die sich auch zum Modellieren von Rohren im Gerüstbau eignen würden werden im Folgenden aufgelistet. Mit Leica CloudWorx müssen einzelne Rohre ausgewählt werden, woraufhin ein Algorithmus die Mittelachse dazu berechnet und das richtige Rohr an der entsprechenden Stelle platziert. Mit Autodesk Plant 3D müssen innerhalb der Punktwolke eines Rohres

zwei Punkte ausgewählt. Anschließend wird das Rohr nachmodelliert und an die richtige Position gebracht. (Wang et al. 2015) Eine Überlegung, wie diese Ansätze in der Erkennung von Gerüsten angewendet werden könnten ist in 2.4.4 zu finden.

Der Scan-to-BIM Ablauf kann grob in drei, aufeinanderfolgenden Arbeitsschritte eingeteilt werden. Zuerst wird die Punktwolke segmentiert, wobei die einzelnen Punkte, beziehungsweise Voxel den jeweiligen Bauteilen zugeordnet werden. Anschließend werden die Schnittstellen dazwischen zu den einzelnen Komponenten identifiziert und die Bauteile klassifiziert.

2.4.1 Segmentierung

Unter Segmentierung versteht man den Prozess bei dem die unsortierten Punkte der Wolke, anhand von ähnlichen Eigenschaften, zu Einheiten zusammengefasst werden. Dies kann manuell oder automatisch erfolgen. Die algorithmusbasierten Ansätze kann man generell in direkte und indirekte Herangehensweise unterscheiden. Beim direkten Ansatz werden die Punkte anhand von geometrischen Ähnlichkeiten sukzessive gruppiert. Der indirekte Ansatz wendet eine Hough Transformation an und schätzt die geometrischen Parameter ab. Im Folgenden werden die verschiedenen Methoden der Segmentierung beschrieben. (Zhan et al. 2009)

Methode 1: Farbbasierte Segmentierung

Diese Methode arbeitet mit der Ähnlichkeit der Farben und räumlicher Nähe. Der Algorithmus beginnt beim Saatpunkt und breitet sich räumlich aus. Dabei sucht er den jeweils nächsten Nachbarn und fusioniert Bereiche, die anhand von Farbe und räumlicher Entfernung definiert werden. Ein Beispiel dafür ist die aggregierte grüne Fläche unter dem Torbogen in **Abbildung 15**.



Abbildung 15 Anwendung Farbsegmentierungs Algorithmus bei einem chinesischen Tempel (Zhan et al. 2009)

Im Gerüstbau könnte dieses Verfahren für eine erste grobe globale Einordnung verwendet werden, da sich die farblichen Bestandteile immer in einem charakteristischen Muster befinden. Als Merkmal würden sich dafür die in der **Abbildung 16** dargestellten gelben Bordbretter eignen. So ist es möglich die Gerüstkomponenten von der Wand zu differenzieren und die Orientierung im Raum als verschlüsselte Bedingung für den lokalen Algorithmus zu verwenden.



Abbildung 16 Peri Up Easy mit colorierten Bordbrettern (Softtec 2018)

Methode 2: Formen Erkennung

Der Algorithmus basiert auf der Erkennung von Formen, wobei er in dem zitierten Ansatz vor allem bei planaren Flächen angewandt wird. Dabei wird auf die Ebenen jeweils der Normalenvektor konstruiert. Als Startpunkt dient ein Saatpunkt P , von dem aus sich der Algorithmus räumlich ausbreitet und die Punkte nach ihrem Krümmungswert beurteilt. Ausgangspunkt ist dabei der Ort mit dem geringsten Krümmungswert. Anschließend wird auf den Saatpunkt und seine Nachbarpunkte der Normalenvektor konstruiert, um deren Winkel zueinander zu bestimmen. Falls der Winkel geringer als ein Residuum θ_{grenz} ist wird der aktuelle Saatpunkt zur Fläche hinzugefügt. Anschließend wird der Krümmungswert der Nachbarpunkte mit dem residualen Wert C_{grenz} abgeglichen. Wenn er kleiner ist werden die Nachbarpunkte in die Liste der Saatpunkte aufgenommen und der Algorithmus fortgesetzt, bis alle Punkte gelabelt sind. Als Resultat erhält man ein Punktecluster, wobei alle Punkte des gleichen Clusters Teil der gleichen ebenen Fläche sind. (FROM POINT CLOUD TO BIM: A SURVEY OF EXISTING APPROACHES 2013) (Wang et al. 2015)

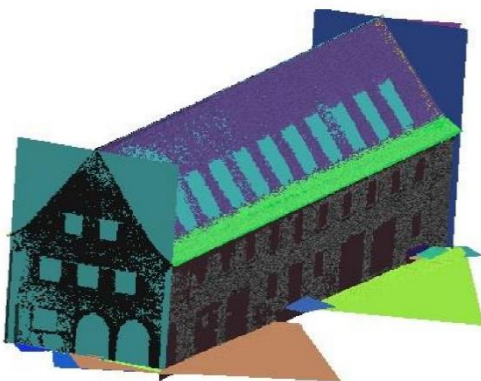
Methode 3: Distanz zwischen planaren Flächen

Die Methode basiert auf der Berechnung der Distanz zwischen ebenen Flächen. So wird das Cluster für die Saatpunkte bestimmt, um den räumlich wachsenden Algorithmus zu starten. Nach einer Analyse der Komponentenverbindungen werden alle Saatpunkte mit ähnlicher Struktur zusammengefasst (FROM POINT CLOUD TO BIM: A SURVEY OF EXISTING APPROACHES 2013). Anschließend werden die Punkte der Aufnahme den maßgebenden Flächen des Objekts zugeordnet. Jene können jedoch nur approximiert werden. (Malihi et al. 2016) Ein Beispiel dafür ist die approximierte Giebelseite in **Abbildung 17**. Dabei wurden Vorsprünge von Fenstern oder dem Dach außer Betracht gelassen.

Abbildung 17 Punkte werden den Hauptflächen des Objekts zugeordnet (Malihi et al. 2016)

Methode 4: Differenzierung von Fassade und Gerüst

Die zuvor beschriebenen Methoden 1 bis 3 der Punktwolken Segmentierung eignen sich vor allem für Häuser oder Hallen. Da es sich bei Gerüsten um sehr feingliedrige Strukturen handelt, die im Fall von Fassadengerüsten immer mit einem Gebäude einherkommen, müssen weitere Unterscheidungen getroffen werden. Somit ist es sinnvoll im ersten Schritt das Gerüst von der Fassade zu trennen. Der Algorithmus hierfür kann in drei Stufen eingeteilt werden.



1. Projektion und Selektion der Punkte
2. Extrahierung der ebenen Flächen
3. Gruppierung der ebenen Flächen

1. Im ersten Schritt wird die gesamte 3D

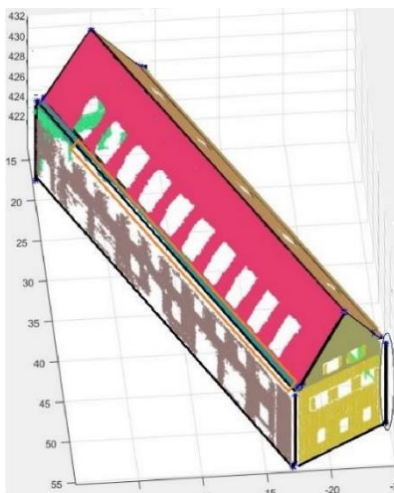
Punktwolke in 2D auf den Boden projiziert. Da sich die Punktwolke nach der Vorverarbeitung in Form eines Voxelgitters befindet kann sie direkt in Richtung der Spalten verarbeitet werden. Dafür muss für das entstehende 2D Bild am Boden eine geeignete Pixeleinheit gefunden werden. Die Anzahl der Pixel ergibt

sich indem man die Länge der x und y Ausrichtung durch die Dimensionierung der Pixel teilt. Um einerseits Störungen der Daten zu vermeiden und andererseits die Notwendigen Merkmale zu erfassen muss eine geeignete Abmessung der Pixel gefunden werden. Sie orientiert sich an der Abmessung der Rohre und liegt in einer Größenordnung von circa dem dreifachen Rohrdurchmesser. Als Selektionskriterium in Gebäude und Gerüst wird die Intensität der Pixel gewählt. Je mehr Voxel, beziehungsweise Punkte in ein Pixel projiziert werden, desto größer dessen Intensität. Da es sich bei dem Gebäudewand um eine monolithische Struktur handelt ist dessen Intensität um ein Vielfaches höher als die der Gerüstkonstruktion.

2. Dieser Schritt erfolgt unter der Vorannahme, dass die Hauptflächen der Fassade eben sind und lotrecht auf dem Boden gerichtet sind. Unter den Hauptflächen sind die Wand und die innere und äußere Reihe der Gerüstrohre zu verstehen. Die Extrahierung dieser Flächen gelingt mit Hilfe eines RANSAC Algorithmus. (Xu et al. 2015) Dieser zeichnet sich vor allem dadurch aus, dass er in innerhalb einer sehr großen Stichprobe von Messwerten mit vielen Ausreißern in der Lage ist ein stabiles Modell zu generieren (Fisher 2002).
3. Im letzten Schritt der Segmentierung werden alle ebenen Flächen, die zu einer Fassade gehören zusammengefasst. Dafür wird anhand der Winkel der Normalenvektoren auf den Flächenstücken beurteilt, ob diese parallel sind oder nicht. Falls ersteres zutrifft wird im Anschluss mit Hilfe der euklidischen Distanz ermittelt, wie weit die Teilflächen voneinander entfernt sind. Falls sie nahe genug beisammen liegen werden sie als zusammenhängende Fläche definiert. (Xu et al. 2015)

2.4.2 Identifizierung der Rand- und Eckpunkte

Das Ziel dieses Schrittes ist es alle Rand- und Eckpunkte eines Modells zu isolieren. Der Input hierfür ist die segmentierte Punktwolke mit Informationen in x, y, z Koordinaten und Information über ihre Ausrichtung und Wölbung. Die Ecken innerhalb des Objekts können aufgrund des großen Wechsels der Wölbung leicht gefunden werden. Da bei den Randpunkten kein Krümmungswechsel vorliegt können diese mit dieser Methode nicht identifiziert werden. In 2D-Flächen können die Randpunkte leicht ermittelt



werden, da bei ihnen der Winkel zu den Nachbarpunkten um ein Vielfaches größer ist als bei innenliegenden Punkten. Die so ermittelten Punkte werden mit den Rändern der Öffnungen und Materialwechseln gleichgesetzt werden. Indem um die einzelnen segmentierten Flächen eine konvexe Hülle gelegt wird kann so das Modell kalkuliert werden. (Wang et al. 2015) (Malihi et al. 2016)

Wie in **Fehler! Verweisquelle konnte nicht gefunden werden.** schematisch dargestellt können so auch die Schnittstellen an verdeckten Bereichen der Aufnahme

ermittelt werden. (Malihi et al. 2016)

Abbildung 18 Kanten und versteckte

Kanten an Gebäude (Malihi et al. 2016)

2.4.3 Klassifizierung

Ziel der Klassifizierung ist es die einzelnen 3D-Punkte der einzelnen Gerüstkomponenten aus dem Aufbau zu definieren und von anderen Bauteilen zu differenzieren. Dafür müssen in vielen Fällen nicht alle Teile erkannt werden, sondern es reicht oft aus, nur bestimmte markante Bestandteile wie Rohre oder Bordbretter zu bestimmen. Aus ihrer Konstellation können dann die restlichen Bauteile abgeleitet werden. Im Folgenden werden drei Ansätze beschrieben, um die einzelnen Punktwolken aus ihrer Umgebung auszulösen und zu klassifizieren.

Random Forest Algorithmus (RFA)

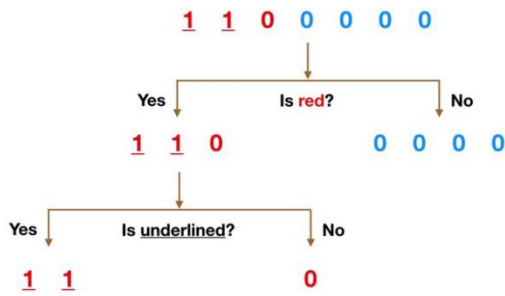


Abbildung 19 Entscheidungsbaum (Yiu 2019)

Eine Möglichkeit die einzelnen Gerüstkomponenten zu erkennen ist durch den sogenannten Random Forest Algorithmus. Um die Scan Punkte den speziellen Elementen eindeutig zuzuordnen wird dem eigentlichen RFA ein Algorithmus zur 3D-Formgebung vorgeschaltet, um alle abweichenden

Punkte zu entfernen. Beim eigentlichen RFA handelt es sich um eine große Anzahl an individuellen Entscheidungsbäumen, die als Ensemble zusammenarbeiten. Dabei spaltet jeder Baum die Menge der zu klassifizierenden Objekte nach einem bestimmten Merkmal auf. Der Entscheidungsbaum in **Abbildung 19** kennt beispielsweise die Merkmale Farbe und unterstrichen/nicht unterstrichen. Im Fall der Erkennung von Gerüsten werden als Klassifizierungsmerkmale hauptsächlich geometrische Daten an den Eckpunkten verwendet. Anschließend wird für jede Klasse ein Konfidenzintervall entwickelt, das repräsentiert, wie groß die Wahrscheinlichkeit ist, dass eine bestimmte Stichprobe zu der jeweiligen Klasse gehört. Das finale Ergebnis ergibt sich wie in

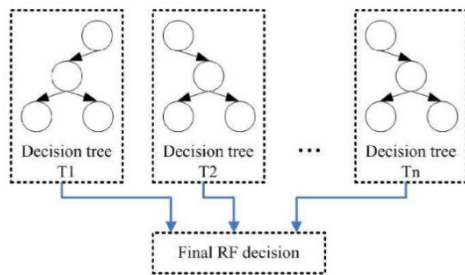


Abbildung 20 dargestellt durch ein Votum aus den Mittelwerten der einzelnen Entscheidungsbäumen. (Yiu 2019)

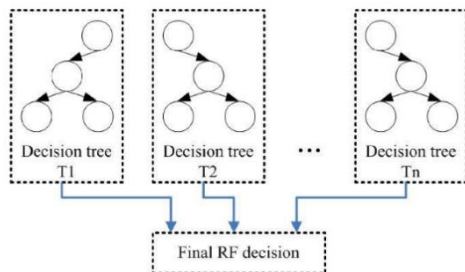


Abbildung 20 Votum der einzelnen Entscheidungsbäume (Yiu 2019)

Klassifizierung des Materials

Den Input für die Klassifizierung des Materials stellen quadratische Bildausschnitte dar, die anhand von Textur und Farbe in die entsprechende Kategorie eingeordnet werden. Das Prinzip dahinter, ist die sogenannte Bag-of-Words Methode, bei der die Klassifizierung für jede Materialkategorie separat trainiert wird. Dabei werden alle Materialkategorien isoliert, das heißt ohne Anordnung oder Zusammenhang, betrachtet und mit Nummern versehen. Ein entscheidender Parameter ist dabei die Häufigkeit des Auftretens einzelner Kategorien. Im Anschluss wird das betrachtete Material mit allen zur Verfügung stehenden Kategorien aus dem Bag-of-Words abgeglichen und die mit der höchsten Übereinstimmung ausgewählt. (Becker 2019)

Die Farbgebung kann mithilfe des HUE Saturation Values bestimmt werden. Dies beinhaltet eine Analyse des Farbwertes, der Farbsättigung und der Helligkeit (Hue, Value, Saturation 2019).

Die Textur wird durch die sogenannten Leung-Malik Filter ermittelt. Dabei handelt es sich um eine Anzahl von circa 40 Filtern, die die einzelnen Strukturelemente nach Ausrichtung und Dimension einteilen.

Indem man die Merkmale Farbe und Textur gleichzeitig betrachtet kann man daraus das entsprechende Material bestimmen. Dafür verwendet man einen k-Means-Clustering Algorithmus. Im ersten Schritt muss dafür die genaue Anzahl der resultierenden Materialtypen festgelegt werden. Anschließend werden die beiden Histogramme überlagert und durch Vektorquantifizierung die Wertebereiche für die zuvor festgelegten Materialgruppen zugeordnet. Ein Überblick über die Vorgehensweise ist in **Abbildung 21** dargestellt.

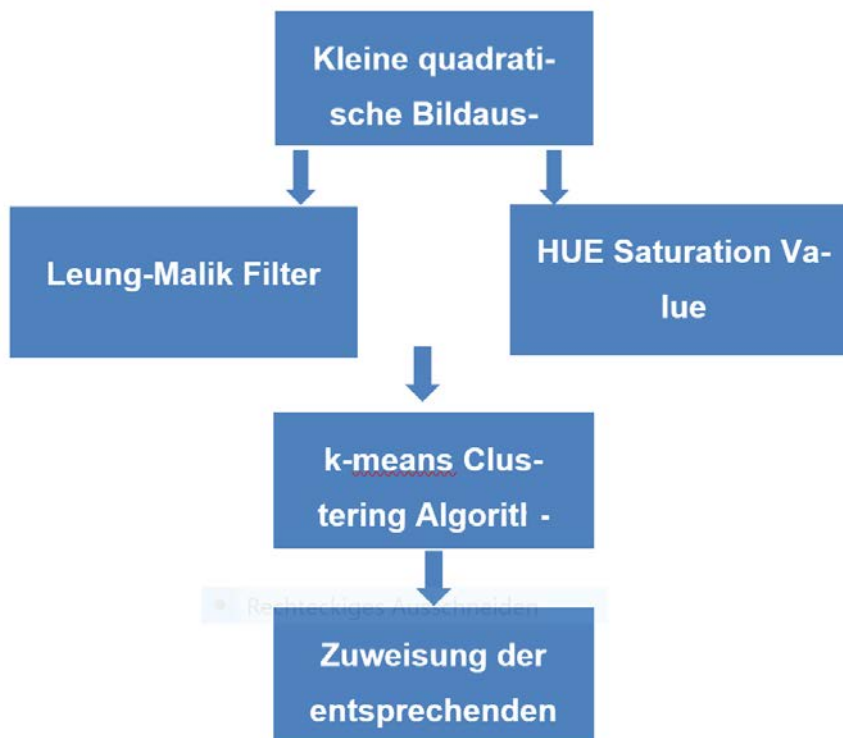


Abbildung 21 Chronologischer Ablauf der Materialklassifizierung

(Han und Golparvr-Fard)

Segmentierung der Punktwolke

Bei diesem Ansatz geht es vor allem darum die störenden Elemente aus der Punktwolke, wie Baumaschinen, Menschen oder andere Bauelemente, zu entfernen. Um dies zu erreichen müssen Elemente definiert werden, die in jeder Version des Bauteils vorhanden sind und leicht identifiziert werden können. Im Folgenden wird die Vorgehensweise am Beispiel eines Baugerüsts erläutert, wobei die konstanten Elemente die vertikalen Rohre sind. Anschließend werden die weiteren Bestandteile des Gerüsts (Beläge, Bordbretter) identifiziert.

Im ersten Schritt wird die Punktwolke in einem bestimmten Intervall in horizontale Scheiben geteilt (**Abbildung 22 a**). Das Intervall ist dabei abhängig von der Dichte der Punktwolke. Wie in **Abbildung 22 b**) dargestellt, werden alle Punkte innerhalb einer Schicht auf die x-y-Ebene projiziert. Anschließend werden mit Hilfe eines Region

Growing Algorithmus alle Punkte eines Rohrquerschnitts gebündelt und als ein Segment definiert. Das entscheidende Kriterium für die Gruppierung ist die räumliche Nähe der Punkte. Nachdem alle Kreise gefunden sind wird ihr Radius bestimmt und getestet, ob sich dieser in dem zulässigen Intervall befindet. Dieser Schritt wird durch die **Abbildung 22 c)** und **Abbildung 22 d)** dargestellt.

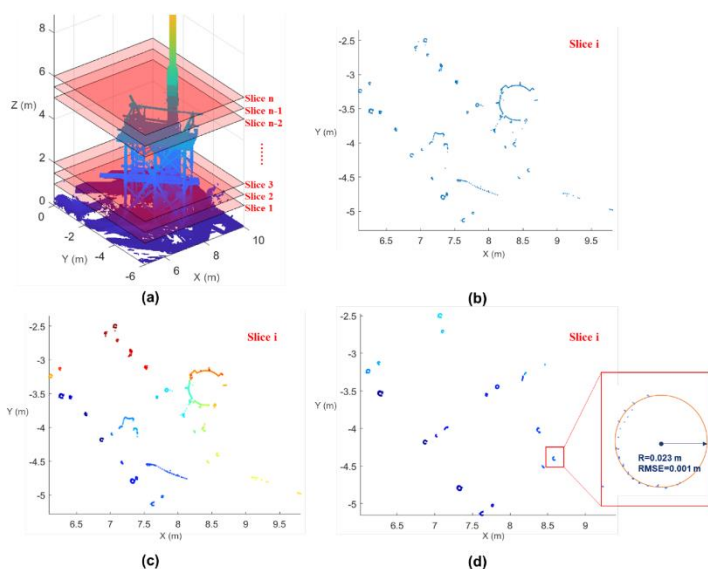


Abbildung 22 Zwischenschritte bei der Extrahierung der vertikalen Rohre aus der Punktwolke

Anschließend wird für jeden Kreis in den verschiedenen Schichten der Mittelpunkt bestimmt. Wenn eine bestimmte Anzahl an Kreisen in den gestapelten Schichten einen Mittelpunkt mit ähnlichen Koordinaten haben wird um sie eine konvexe Hülle gelegt. Alle darin enthaltenen Punkte werden so einem vertikalen Rohr zugeordnet.

Im nächsten Schritt werden die Beläge des Gerüsts identifiziert und extrahiert. Deren Alleinstellungsmerkmal ist ihre ebene Form und horizontale Lage. Durch die Methode der kleinsten Quadrate wird für jeden Punkt und seinen 8 Nachbarn eine Ebene gebildet. So kann für alle Punkte eine lokale Normale errechnet werden. Wenn deren Winkel mit der Lotrichtung weniger als 20° abweicht ist die Wahrscheinlichkeit sehr hoch, dass der entsprechende Punkt Teil eines Belages ist. Anschließend wird aus den gefundenen Punkten ein Histogramm ihrer z-Werte erstellt. Die Bereiche, in denen die Anzahl der Punkte eine bestimmte Marke überschreitet, werden als Potentielle Stellen für einen Belag definiert. Mit Hilfe eines M-estimator SAMple Consensus (MSAC) Algorithmus werden horizontale Ebenen innerhalb der einzelnen Punktemengen errechnet. Die Bedingung dafür ist, dass die Ebene maximal 20° von der Horizontalen abweicht. Daraufhin trennt ein Region Growing Algorithmus die Beläge voneinander und von anderen Objekten. Diese werden wiederum mit einer Bounding Box

eingeschlossen, die rechtwinklig sein muss und eine Mindestfläche besitzt. Ein weiteres Merkmal für die sichere Identifizierung eines Belages ist die Bedeckung von mindestens 50 % der Fläche eines Brettes mit Punkten.

Mit Hilfe der extrahierten Punktwolken der Arbeitsplattformen können im nächsten Schritt die Bordbretter gefunden werden. Zuerst wird dafür umlaufend an den vier Rändern eines Belages alle Punkte im Abstand von 0,1 m extrahiert und anschließend mit der Normalenbedingung eine Ebene definiert, auf die wiederum der MSAC Algorithmus angewendet wird. Da man von der Annahme ausgeht, dass Bordbretter vertikal sind kann man so ihre grobe Form bestimmen. Im Weiteren werden zwei Schritte angewandt, um das Rauschen in der Wolke zu entfernen. Zuerst wird anhand der z-Koordinate des Punktes evaluiert, ob sich die Lage im Zulässigen Spielraum befindet, welcher sich an der zulässigen Höhe des Bordbrettes und der Arbeitsebene als Referenz orientiert. Anschließend wird die Normale eines Jeden Punktes anhand seiner acht nächstgelegenen Nachbarpunkte ermittelt und geprüft, wie weit diese von der zuvor ermittelten Ebene abweicht. Falls die Differenz mehr als 20° beträgt wird der Punkt aussortiert. Das Ergebnis davon ist in **Abbildung 235** in rot dargestellt.

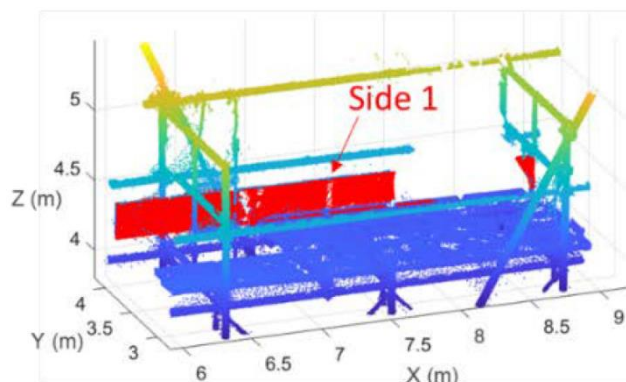


Abbildung 23 Extrahiertes Bordbrett

Um die extrahierte Punktwolke sicher als Bordbrett klassifizieren zu können wird die überdeckte Fläche getestet. Dafür wird das Mindestmaß eines Bordbrettes an die vorgesehene Stelle im Gerüst gesetzt. Die vorgesehene Fläche wird mit einem bestimmten Rastermaß an Pixeln versehen und wenn in einem Pixel mindestens ein Datenpunkt liegt wird dieses in der Farbe weiß markiert. Im Idealfall sind alle Pixel der Fläche weiß. Durch Okklusionen kann es jedoch dazu kommen, dass nicht alle Punkte erfasst wurden, weshalb es schon ausreicht, wenn mindestens eine Reihe der Pixel bis zur Mitte weiß gefärbt ist. (Wang 2019)

2.4.4 Überlegung eines auf den Gerüstbau anwendbaren Ansatzes

Da es im Fassadengerüstbau nur eine sehr begrenzte Anzahl an Rastermaßen gibt könnte man die Anwendung sehr leicht um einige Schritte ausweiten. Beispielsweise könnte man in einem Feld semi-automatisch eine Minimalzahl von vertikalen Rohren auswählen und komplett berechnen lassen. Daraufhin erkennt der Algorithmus das entsprechende Rastermaß und kann den theoretisch korrekten Aufbau des Feldes abmessen. Kann er gezielt danach suchen und bei Übereinstimmung die geometrisch und semantisch korrekten Modellierungselemente einsetzen. So könnten auch teilweise okkludierte Bauteile mit größerer Wahrscheinlichkeit leichter erkannt werden, da sie nicht komplett gefunden und erkannt werden müssen, sondern nur ein Ausschnitt davon abgeglichen werden muss.

2.5 Point Cloud-vs-BIM

In diesem Abschnitt wird auf den Abgleich eines physischen Bauabschnitts/Bauwerks (Feedback) mit seinem designten Modell (Forward) eingegangen. Die Information über das Modell wird dabei aus einem mindestens vierdimensionalem BIM Modell abgeleitet, welches Geometrie und den zeitlichen Ablaufplan enthält. Der aktuelle Stand der Bauarbeiten wird durch eine Punktwolke dargestellt, welche wiederum durch Fotogrammetrie oder Laser generiert werden kann. Generell kann man mit Scan-vs-BIM Qualitative und Quantitative Kontrollen durchführen. Neben der Analyse von Neu- und Umbauten ist die Kontrolle von Bestandsbauten auf Veränderung durch Umwelteinflüsse ein weiteres Anwendungsgebiet von Scan-vs-BIM. Vor der eigentlichen Analyse müssen das bestehende 3D Modell und die Punktwolke durch Registrierung in einem gemeinsamen Koordinatensystem vereinheitlicht werden. Dieser Vorgang wurde im Abschnitt 2.3.1 beschrieben. Im Folgenden werden die verschiedenen Ansätze zum Abgleich genauer erläutert.

2.5.1 Methode 1: Voxelzellen

Zu Beginn des Algorithmus wird das Modell rasterförmig in ebene Flächen wie Drei-

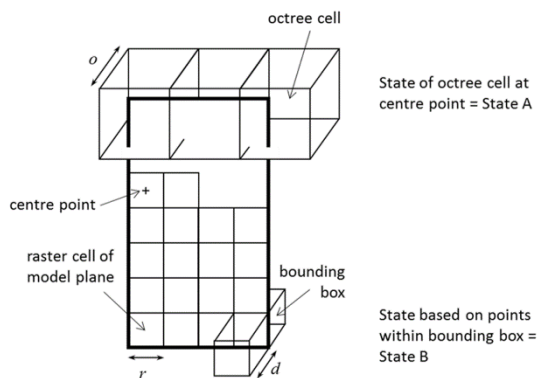


Abbildung 24 Gliederung des Modells in Voxel und Raster (Tuttas et al. 2014)

oder Rechtecke gegliedert. Die Rasterzellen kommen dabei in zwei verschiedenen Zuständen vor.

Abgleich A basiert wie in **Abbildung 24** dargestellt auf visuellen Merkmalen mit einheitlichem Voxelmaß. Anhand der Lage der Punkte im Oktonärbaum kann so ermittelt werden, welche Bauteile mit der Punktwolke belegt sind, welche ver-

deckt werden und welche noch nicht erstellt worden sind. Mit Hilfe von Raycasting von der Kamera zum Oktonärbaum kann zwischen drei Zuständen der Voxel unterschieden werden.

1. Das Voxel ist belegt - enthält 3D-Punkte
2. Das Voxel ist frei – enthält keine Punkte, aber mindestens ein Strahl geht durch die Zelle
3. Der Zustand ist unbekannt

Dabei ist anzumerken, dass die Voxelzellen des Oktonärbaums viel größer sind als die Rasterzellen auf dem Modell. So kann der Rechenaufwand reduziert werden, indem alle Rasterzellen innerhalb einer unbelegten Voxelzelle ebenfalls auf den Zustand „unbelegt“ gesetzt werden.

Beim Abgleich B wird der Abstand der 3D Punkte vom Modell getestet. Dies ist notwendig, da aufgrund der großen Voxelgröße nicht sicher gesagt werden kann, ob die Rasterzellen innerhalb des Bereichs eines Voxels wirklich existieren. Dafür werden für die Punkte innerhalb einer Voxelzelle insgesamt drei Berechnungen durchgeführt.

1. Der durchschnittliche orthogonale Abstand der Punkte zum Modell
2. Die Standardabweichung aus 1.
3. Der Winkel zwischen der Normalen des Modells und der Normale der konstruierten Ebene auf die Punkte

Das Bauteil wird nur erkannt, wenn die Werte klein sind und in einem gewissen Wahrscheinlichkeitsbereich liegen. (Zitat: Braun, Tuttas)

2.5.2 Methode 2: Projektion

Kerngedanke der Methode ist die Klassifizierung der 3D-Punkte in zwei verschiedene Zustände. Zustand 1 bedeutet dabei, dass der Punkt Teil des BIM Modells ist. In Zustand 2 existiert das Bauteil in der Realität, wurde aber noch nicht im BIM Modell aktualisiert. Für die Klassifizierung werde die Punkte orthogonal auf die Oberfläche des BIM Modells projiziert und die resultierende Distanz gemessen. Um als Zustand 1 klassifiziert zu werden müssen die Punkte Unterhalb eines gewissen Grenzwerts r liegen. Falls sie größer als der Grenzwert r sind oder ihre orthogonale Projektion keine Oberfläche trifft werden sie als Zustand 2 markiert. Die Dimensionierung von r ist abhängig vom Rauschen in der Punktwolke und vom Fehler bei der Registrierung von Punktwolke und Modell. Am Ende erhält man eine gelabelte Punktwolke, wobei alle im BIM Modell fehlenden Elemente sofort durch das Label 2 erkennbar sind. Ein Nebeneffekt ist die Kontrolle der Position der Elemente durch die Messung der orthogonalen Distanzen.

Wenn festgestellt wurde, dass ein Bauteil sowohl im Modell als auch in der Punktwolke existiert kann anschließend der Anteil der übereinstimmenden Flächen zwischen Punktwolke und Bauteil ermittelt werden. Dafür werden die Punkte wieder orthogonal auf das Modell projiziert um eine 2D alpha-Fläche zu erhalten. Diese wird durch die Delaunay Triangulierung gebildet, wobei alle Punkte die Ecken von Dreiecken mit maximalem kleinsten Innenwinkel sind. Der Übereinstimmungsgrad der Flächen wird durch folgende Formel ermittelt:

$$M_{\text{cov}}(\pi_j) = \text{area}(\alpha) / \text{area}(\pi_j)$$

Dabei ist α die Größe der übereinstimmenden Fläche und π die Größe der gesamten Fläche.

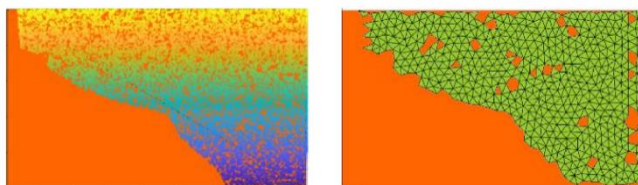


Abbildung 25 Projektion der Punktwolke auf das Objekt (Tran und Khoshelham 2019)

Der **Abbildung 25** Projektion der Punktwolke auf das Objekt zeigt die Oberfläche eines modellierten Objekts π in orange. Auf der linken Hälfte ist die orthogonal projizierte Punktwolke auf das Objekt abgebildet.

Die rechte Hälfte zeigt die Triangulierung α dieser Punkte. In diesem Fall wäre $M_{\text{cov}}(\pi) = 0,56$ (Tran und Khoshelham 2019)

3 Theoretische Grundlagen

3.1 Vorstellung der Firma Peri AG

Die Arbeit wurde in Zusammenarbeit mit dem BIM Competence Centers der Firma Peri AG in Weißenhorn erstellt. Dieses wurde 2017 gegründet, um eine führende Rolle bei der Entwicklung von digitalen Lösungen in der Baubranche einzunehmen. Weitere Standorte des BIM Competence Centers befinden sich in USA und UK. Peri ist mit einem Umsatz von circa 1,4 Milliarden Euro einer der weltweit führenden Hersteller von Schalungen und Gerüsten.

Ein Beispiel für ein von Peri mit BIM ausgeführtes Projekt ist Saldanha MOGS in Südafrika. Für das unterirdische Rohöllager mit einer Kapazität von 13 Mio Barrel Rohöl stellte Peri die Schalungen und Gerüste zur Verfügung.

Ein Projekt aus dem Gerüstbereich, das der Aufgabenstellung der vorliegenden Arbeit sehr nahe kommt, ist der Bau einer Prozessanlage für BASF in Ludwigshafen. Partner von Peri war dabei die Promaintain GmbH & Co. KG. Das Projekt stellt einen Meilenstein in der Verwendung von BIM im Gerüstbau dar, da es eines der ersten Industrie-gerüstprojekte handelt, das vollständig unter Verwendung von BIM ausgeführt wurde. Anwendungen, die dabei mit BIM umgesetzt wurden, waren die Planungskoordination, Kollisionsprüfungen, Sicherheitschecklisten und QR Codes zur Objektnavigation. (PERI AG 2018) Die folgende Abbildung zeigt das Logo der Firma Peri AG.



Abbildung 26 Logo der Firma Peri AG

3.2 Allgemeine Eigenschaften von Punktwolken

Fotogrammetrie- oder LIDAR-Punktwolken – beide haben ihre Vor- und Nachteile und eignen sich für jeweils unterschiedliche Anwendungen. Das ausschlaggebende Kriterium, das sie unterscheidet, ist, dass Punktwolken, die durch fotogrammetrische Aufnahmen entstehen einen RGB Wert besitzen. (GIM International 2018)

Dem RGB Farbschema liegen die drei Primärfarben rot, grün und blau in 255 verschiedenen Nuancen zu Grunde. Diese können beliebig vermischt werden, sodass insgesamt 16,7 Millionen verschiedene Farbwerte dargestellt werden können. Die Kehrseite einer sehr hohen farblichen Differenzierung ist die hohe Speicherkapazität, die dafür benötigt wird. (Wörgl)

Fotogrammetrische Punktwolken können so die Objekte in einer sehr realitätsnahen Darstellung wiederspiegeln. Mit LIDAR-Punktwolken hingegen können sehr viel höhere Präzisionsanforderungen erfüllt werden. Dabei können die Abweichungen durch fotogrammetrische Aufnahme, je nach Entfernung und Art des Objekts, im cm bis m Bereich liegen. (GIM International 2018)

Im Folgenden werden die maßgeblichen Faktoren beschrieben, von denen die Qualität der Punktwolken durch Fotogrammetrie und Laser abhängen.

3.3 Einflüsse auf die Qualität der fotogrammetrischen Methodik

3.3.1 Ground Sample Distance

Der Parameter beziffert das kleinstmögliche Element, das vom Kamera Sensor erkannt werden kann. Je größer der Wert des GSD ist, desto geringer ist die räumliche Auflösung und damit die sichtbaren Details. Der GSD kann durch folgende Formel ausgedrückt werden:

$$\text{GSD} = \frac{\text{Pixel size (on sensor)}}{\text{Focal length}} * \text{Distance object to camera}$$

Aus der Formel wird ersichtlich, dass der GSD sowohl von den Eigenschaften der Kamera, als auch von der Entfernung vom Objekt abhängt. (Faltýnová et al. 2016)

3.3.2 Kamerasensor

Die Wahl des Kamerasensors hängt maßgeblich vom geforderten Detailgrad des Modells ab. Daraus leiten sich die Modellierbarkeit der Objekte und die akkurate Anordnung ab. Der gängige Bereich reicht von 15 – 40 Megapixel. (Faltýnová et al. 2016) Des Weiteren gibt es verschiedene Sensoren, die Einfluss auf das spätere Datenformat haben. Beispiele hierfür sind Microsoft Kinect, Indoor Mobile Mapping Systems oder Google Tango. (Runceanu und Haala 2018)

3.4 Einflüsse auf die Qualität der Laser Methodik

Die Zeit für die Auswertung und Aussagekraft der Aufnahme können sehr stark variieren je nach Qualität des Scans. Dadurch variieren auch die Kosten der Verarbeitung sehr stark. Grundsätzlich kann man die Parameter in innere und äußere unterteilen. (SHEN Z. et al. 2013) Da die Präzisionsanforderungen in unserem Anwendungsfall lediglich im Bereich von 10mm liegen, haben die inneren Parameter nur geringen Einfluss auf die Genauigkeit des Modells. Aus diesem Grund wird der Schwerpunkt dieser Arbeit auf die äußeren Parameter gelegt.

3.4.1 Innere Parameter

Die inneren Parameter sind von der Beschaffenheit des Geräts abhängig. Jene sind teilweise durch Kalibrierung des Geräts zu bestimmen und können im Ergebnis durch Korrekturen ausgeglichen werden. (Gordon 2008) Im Folgenden wird der Parameter der Wellenlänge beschrieben. Da in unserem Fall Abweichungen von bis zu 2cm akzeptabel sind ist hauptsächlich nur dieser Parameter relevant.

Wellenlänge

Die Wellenlänge wird üblicherweise in Nanometer angegeben. Bei einer gegebenen diffusen Oberfläche ist das Rückstrahlvermögen des Ziels vor allem abhängig von der Wellenlänge des einstrahlenden Lasers. Dies wird ausgedrückt durch den sogenannten Albedo Wert, welcher in Form einer dimensionslosen Zahl angegeben wird. Dabei bezieht sich der Wert auf das Verhältnis von rückgestrahltem zu einfallendem Licht. Bei einem Albedo Wert von 0,7 bedeutet dies, dass 70% des einfallenden Lichts zurückgestrahlt wird. (Wunderlich et al. 2013) Die Wellenlänge von Laserstrahlen liegt, wie in **Abbildung 27** dargestellt zwischen der, der Röntgenstrahlung und der des Radars (Fromme 2014).

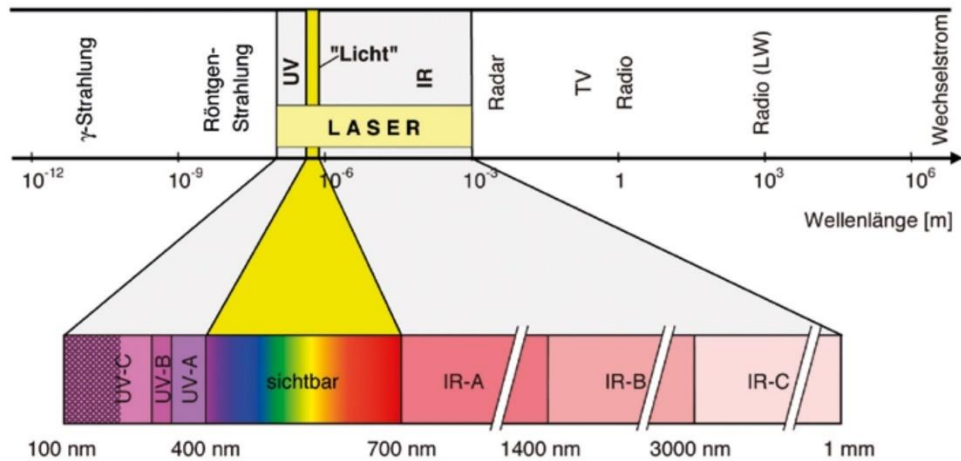


Abbildung 27 Übersicht Wellenlängen (Fromme 2014)

3.4.2 Äußere Parameter

Die äußeren Parameter hängen von den Gegebenheiten auf der Baustelle ab. Sie beeinflussen den Lichtstrahl vom Austreten aus dem Gerät bis zum Rückweg nach der Reflexion. In unserem Anwendungsfall sind sie bezüglich der Qualität des Scans viel höher zu gewichten als die inneren Parameter, da sie nur wenig durch Kalibrierung ausgeglichen werden können. (Gordon 2008) Im Folgenden werden die gravierendsten 5 Parameter ihrem Einfluss nach aufgelistet. Die drauf folgenden Größen haben einen geringeren Einfluss auf die Scan Qualität. (SHEN Z. et al. 2013)

Auflösung

Unter der Auflösung versteht man die Dichte der resultierenden Punktwolke, also den durchschnittlichen Abstand zwischen den einzelnen Messpunkten. Daraus ergibt sich direkt die Größe des kleinsten Objekts, das erfasst werden kann. Bei der Dichte der Punktwolke müssen vor allem zwei Aspekte beachtet werden. Erstens ergibt sich daraus der Grad der Detailliertheit des späteren Modells. Zweitens beeinflussen Messungsgenauigkeiten die abgeleitete Geometrie des Modells. (Macher et al.)

Entfernung

Je größer die Distanz zwischen Scanner und Modell wird desto ungenauer wird die Messung. Der Hauptgrund dafür ist die längere Laufdistanz des Lichtsignals, was zu einer Aufweitung des Strahls führt. (SHEN Z. et al. 2013)

Oberflächenfarbe des Modells

Ein entscheidender Parameter bei der Oberflächenbeschaffenheit ist die Farbe. Da weiße Körper Licht viel besser reflektieren als schwarze, ergeben sich bei ihnen viel genauere Scans. In einem Experiment, das von Shen und Tang durchgeführt wurde, ergaben sich dabei folgende Abweichungen vom as-built Radius für die verschiedenen Farben. (SHEN Z. et al. 2013)

	<i>Farbe</i>	
<i>Weiß</i>	<i>Grau</i>	<i>Schwarz</i>
<i>0,043</i>	<i>0,073</i>	<i>0,089</i>

Wie in **Fehler! Verweisquelle konnte nicht gefunden werden.** dargestellt kann man im Allgemeinen sagen, dass hellere Farben aufgrund ihrer höheren Reflexionsfähigkeit besser geeignet sind als dunkle. Auch die Farbe und das Material des Körpers haben einen maßgeblichen Einfluss auf den oben beschriebenen Albedo Wert. Als generelle Regel gilt je heller ein Körper ist, desto größer ist sein Albedo Wert. (Wunderlich et al. 2013)

Zugänglichkeit und Verdeckungen

Es ist sehr schwierig jeden Winkel eines Objekts so zu scannen, dass alle relevanten Oberflächen erfasst sind. Die Hauptursache hierfür sind Selbstverdeckungen der Objekte. Ein Beispiel hierfür ist die Rückseite von Leitungen, die sich an einer Wand befinden. Einer Studie zufolge betrifft diese Problematik 35 % aller Flächen von BIM Objekten. So existieren in jedem Modell Objekte, die mit unvollständigen Daten modelliert werden. Bei stellenweise verdeckten Objekten können die fehlenden Informationen durch Ableiten von Dimensionen und Oberflächenformen vervollständigt werden. Wenn diese Stellen vor der Aufnahme bekannt sind kann man die Aufstellorte optimieren. Dafür ist es oft hilfreich einen Plan zu erstellen. (Anil et al.) Die Anzahl der Aufstellorte des Lasers wird durch die Datenmenge und Zeit limitiert, denn jeder Scan beansprucht Speicherplatzvolumen und Arbeitszeit. Eine weitere Störquelle stellen Hilfskonstruktionen wie beispielsweise die Staubschutzplanen an Gerüsten dar.

Atmosphäre

Die Umgebungsbedingungen haben einen großen Einfluss auf die Leistungsfähigkeit des Lasers. Die Ursache dafür ist, dass die meisten Scanner bei sehr hohen Temperaturen und ungewöhnlich hoher Luftfeuchtigkeit nur ungenaue Messungen erzeugen. Diese Bedingungen können auch schon bei starkem Regen vorliegen. Da die Toleranzgrenzen von Gerät zu Gerät variieren muss vor einem Einsatz die Eignung geprüft werden.

Kometenschweif

Da der Laserstrahl vom Gerät automatisch über das Umfeld geführt wird und nicht ein spezifischer Punkt angesteuert wird ergibt sich an den Kanten und Ecken der Effekt des sogenannten „Kometenschweifs“. Allgemein gibt es drei Möglichkeiten wie ein Kometenschweif entstehen kann. Die erste Variante ist in **Abbildung 28** zu sehen, wobei ein Teil des Laserstrahls auf einer Kante im Vordergrund liegt und der andere Teil auf einen Punkt in weiterer Entfernung projiziert wird. Ein weiterer Beispielfall liegt vor, wenn das angezielte Objekt dünner ist als der Laserstrahl oder eine Interaktion mit dem Boden stattfindet. Dies ist bei unteren Ecken möglich, die mit der Erde verbunden sind. Die Distanz wird dabei aus einem Mittel der beiden Entfernungen gebildet und der 3D-Punkt zwischen den beiden Raumpunkten gesetzt. Um diese Beeinträchtigung zu reduzieren sollte der Laserstrahl möglichst gut gebündelt werden. (Gordon 2008) (Hajian und Becerik-Gerber 2010)

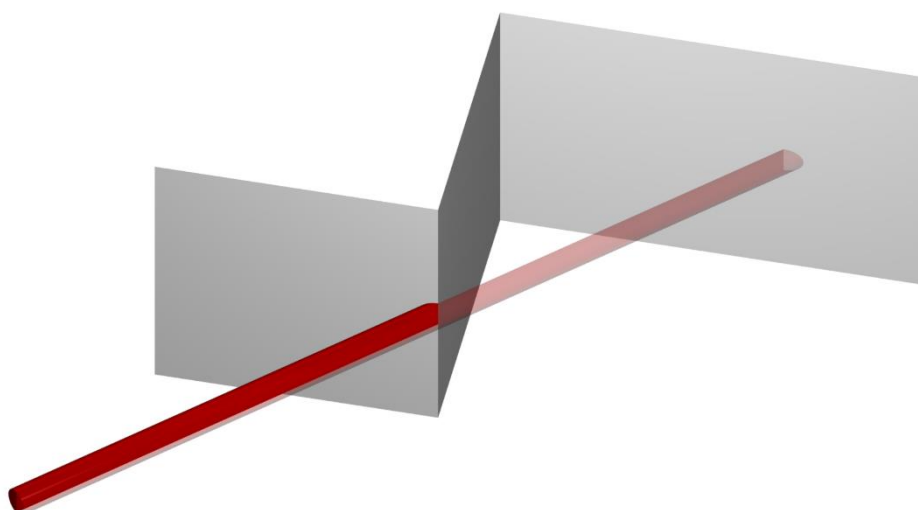


Abbildung 28 Schematische Darstellung des Kometenschweif Phänomens (Gordon 2008)

3.5 Darstellung und Datenformate

Building Information Modeling umfasst einen sehr weiten Bereich und meint den Vorgang digitale Informationen über Bauwerke zu erstellen und zu managen. Dementsprechend groß ist die Bandbreite an Software Systemen zur Bearbeitung der verschiedenen Applikationen. Das impliziert eine große Anzahl an Methoden wie Daten verarbeitet und gespeichert werden können. Als Resultat existieren hunderte von proprietären und nicht-proprietären Datenformaten.

Ein proprietäres Datenformat bedeutet, dass es ausschließlich von der Software eines Herstellers gelesen werden kann. Anderen Systemen ist es nicht möglich die Daten zu lesen. Dies kann zu Problemen bei der Zusammenarbeit bei Projekten zu Problemen führen. Beispiele dafür sind .rvt von Revit oder .dwg von AutoCAD. (File formats for BIM 2019)

Nicht proprietäre Formate sind herstellernerutral und können von jeder Software gelesen und bearbeitet werden. Das wohl bekannteste davon sind die weltweit offenen Industry Foundation Classes (IFC) von Building Smart.

Eine weitere Einteilung der Datenformate ist die Unterscheidung in Formate nach dem American Standard Code for Information Interchange (ASCII) und binäre Dateien. Auch der ASCII Standard basiert auf Binärcode, aber übermittelt die Informationen in Form einer Textdatei. Die Möglichkeit den Code in einem Texteditor zu Öffnen führt zu einer gewissen Zugänglichkeit, Universalität und der Eignung zur Langzeitspeicherung. Meistens Enthalten ASCII Dateien neben den XYZ Werten der Punkte Informationen zur Intensität und zum Farbwert. Beispiele dafür sind die Formate XYZ, OBJ, PTX und ASC. Andererseits ist bei dieser Form der Speicherung das Dateivolumen sehr groß, was zu einer relativ langen Verarbeitungs- und Lesedauer im alltäglichen Arbeiten führt. Binäre Dateien hingegen müssen nicht Zeile für Zeile gelesen werden, sondern können abschnittsweise ausgewertet werden. So können sie aufgrund ihrer Indizierung abschnittsweise ausgelesen werden und in einem Bruchteil der Zeit graphisch dargestellt werden. Nebenbei enthalten binäre Dateien viele erweiterte Informationen wie Metadaten, Daten über die Software oder Dateisignaturen. Beispiele für binäre Datenformate sind FLS, PCD und LAS. Mischformen von binären und ASCII Formaten sind PLY, FBX und e57. (Thomson)

Im Folgenden wird genauer auf die im Versuch verwendeten Datenformate .ifc, .e57, .las und .pts eingegangen.

3.5.1 Datenformat . ifc

IFC ist das seit 1997 existierende, herstellernerneutrale Datenformat und Informationsmodell von BuildingSmart. Es umfasst Informationen aller am Bau mitwirkenden Gewerke und kann über den gesamten Lebenszyklus hinweg verwendet werden. Mittlerweile ist IFC in der Planungs- und Ausführungsphase weit verbreitet und kann bei jeder Software angewendet werden, die den Import unterstützt. (Baunetz Wissen) Genauer gesagt, ist IFC ein standardisiertes Schema, das Informationen über folgende Merkmale enthält: Identität (Bezeichnung), Semantik (Funktion), Attribute (Farbe, Material), Beziehungen (Besitzer), Prozesse (Installation, Wartung) und Personen (Subunternehmer). Dabei erfolgt der Datenaustausch im STEP Format und die Strukturierung im EXPRESS Format. (Building Smart International)

3.5.2 Datenformat .e57

Bei e57 handelt es sich um das am meisten verbreitete Datenformat für den Austausch und die Speicherung von Punktwolken, Bildern und Metadaten, die von 3D Bildgebungssystemen produziert wurden. Ein Beispiel für 3D Bildgebungssysteme sind Laserscanner. Das Dateiformat wurde von American Society for Testing and Materials (ASTM) als herstellernerneutrales Format entwickelt.

Generell ist das e57 Datenformat in drei Teile aufgebaut: Einem Header, einer binären Sequenz und einer XML Sequenz.

Dabei wird jedes Set an Punkten in einer eigenen Struktur gespeichert. Diese sind dort in einem komprimierten Vektor gespeichert. Besonders vorteilhaft bei dem e57 Format ist, dass die Speicherform eines jeden Feldes an Punkten individuell an die Aufnahmeform des Sensors angepasst werden kann. Wenn also ein Sensor die Messpunkte in sphärischen Koordinaten mit Farbintensität erfasst werden nur diese Felder bei der Speicherung belegt. Beispiele für Attribute bei der Speicherung sind Koordinatensystem (kartesisch oder sphärisch), Farbe, Zeile, Spalte oder Intensität. Falls eine Speicherform erforderlich ist, die nicht dem Standard entspricht kann diese leicht mit der Extension Funktion hinzugefügt werden. (Huber 2011)

3.5.3 Datenformat .las

Bei .las handelt es sich um ein offenes, standardisiertes Industrieformat der American Society for Photogrammetry and Remote Sensing (ASPRS) und leitet sich von dem Begriff Laser ab. Es dient derzeit als das am weitesten verbreitete Format im Umgang mit Punktwolken, und kann für die Speicherung von beliebigen XYZ Touplet Daten

verwendet werden. Der Hauptzweck ist das Austauschen und Archivieren von Punktwolkendaten in binärer Form. Binär bedeutet, dass die Information ohne Verwendung von Text transferiert wird. LAS beansprucht circa doppelt so viel Speicherplatz wie .ply und gehört damit zu den ausgedehnteren Formaten. Der Grund dafür ist die Speicherung von weiteren Informationen neben der Position im Koordinatensystem und dem RGB Wert.

Grundsätzlich kann man jede LAS Datei in vier Abschnitte einteilen. Der erste besteht aus allgemeinen Informationen wie der Anzahl der Punkte oder die Abmessung der Gesamten Punktwolke. Der zweite und vierte Abschnitt besitzen eine variable Länge, mit Informationen über das verwendete Koordinatensystem und Metadaten. Die eigentlichen Punkte werden ausschließlich im dritten Abschnitt behandelt und umfassen Informationen wie die Koordinaten oder die Klassifikation der Punkte. (LAS (LASer) File Format, Version 1.4) (Kaarta 2020)

3.5.4 Datenformat .pts

Eine PTS Datei meint eine Textdatei zur Speicherung von Punktwolkendaten, die in zwei Teile gegliedert werden kann. Die erste Zeile gibt dabei die gesamte Anzahl der folgenden Punkte an. Ab der zweiten Zeile besteht jede Zeile aus 7 Werten und beschreibt einen neuen Punkt. Die ersten drei Werte legen die XYZ Position im Koordinatensystem fest. Der vierte Wert beschreibt die Intensität und die letzten drei legen den Farbwert im RGB Schema fest.

Im Allgemeinen ist das PTS ein sehr einfaches Format mit wenig Informationsgehalt. Da es keine Aussage über die ursprüngliche Scan Information enthält eignet es sich vor allem für das Arbeiten mit fertig registrierten und vereinten Punktwolken. (Bourke 2014) (Geospatial Modeling & Visualization 2018)

3.6 Vergleich von Fotogrammetrie und Laser

Laser, beziehungsweise LIDAR Systeme eignen sich generell eher für Anwendungsfälle, bei denen die Kosten durch die höheren Anforderungen an Präzision gerechtfertigt werden können. Probleme mit der Genauigkeit ergeben sich vor allem bei diffus reflektierenden Oberflächen. Typische Anwendungsbereiche sind Produktion, Logistik, Luftfahrt und die Baubranche.

Fotogrammetrische Systeme messen die Bauteile nicht direkt, sondern generieren die Maße erst durch nachträgliche Berechnungen. Somit erzielen sie oft Abweichungen,

die im Bereich von cm bis m liegen. Wenn jedoch große Flächen in kurzer Zeit und mit größeren Toleranzspielräumen erfasst werden sollen eignen sie sich sehr gut. Beispiele hierfür sind breitflächige Aufnahmen von Wäldern, Landwirtschaft, Kartografie und Landstrichen.

4 Case Study

Ziel unseres Versuches ist es einen Scan-vs-BIM Abgleich von einem Gerüstmodell und dessen Punktwolke durchzuführen. Dabei soll ermittelt werden, wie sich verschiedene Dichten der Punktwolken und Parameter auf die Qualität des Abgleichs auswirken.

4.1 Zielsetzung

Ziel der Bachelorarbeit ist es verschiedene Grade der Punktwolkendichte zu testen um deren Einfluss auf die Qualität des Scan-vs-BIM Abgleichs zu finden. Die beiden Extreme sind dabei einerseits eine Punktwolke mit sehr geringer Auflösung, die zwar wenig Rechenaufwand erfordert, aber hohe Ungenauigkeiten mit sich bringt und andererseits eine zu hohe Auflösung, die sehr hohem Detailgrad, aber extrem hohem Rechenaufwand. Des Weiteren werden weitere Parameter ermittelt und getestet, die einen maßgeblichen Einfluss auf die Durchführbarkeit und Qualität der Analyse haben. Die Tests werden mit Hilfe von Autodesk Point Layout und einem Python Code durchgeführt.

4.2 Datenakquise

Für die Erfassung der Punktwolke haben wir uns aufgrund der feingliedrigen Struktur der Punktwolke für die Verwendung eines Lasers entschieden. Fotogrammetrie ist für die Erfassung eines Gerüsts eher ungeeignet, da die Struktur sehr feingliedrig ist. Bei ehemaligen Versuchen mit Fassadengerüsten konnten nur 40-60% der Bauteile erkannt werden. Des Weiteren können mit Fotogrammetrie nur viel geringere Genauigkeiten erzielt werden, die im Fall von beispielsweise Industrierüsten im Kraftwerksbau essentiell ist. (Technische Universität München 2016)

Der von uns durchgeführte Versuch wurde im Fortbildungszentrum der Firma PERI in Weißenhorn durchgeführt, wobei es sich um eine geschlossene Halle mit Fenstern handelt. Man kann also von Laborbedingungen sprechen, da die Aufnahmen ohne Einfluss von Witterung und laufendem Baubetrieb durchgeführt wurden. Im Folgenden wird die Durchführung des Scans mit seinen technischen Details beschrieben.

4.2.1 Allgemein

Bei dem gewählten Scanner handelt es sich um das Modell Leica RTC 360, womit jeder einzelne Scan in Form eines sphärischen Fulldoms durchgeführt werden konnte. Dabei lag der horizontale Messbereich bei 360° und der vertikale bei 300°. Zusätzlich liefert das Gerät ein sphärisches HDR-Bild mit 432MP, wofür es eine Prozesszeit von einer Minute braucht. Um bewegliche Objekte, wie Menschen, Autos oder sich drehende Kräne, in Echtzeit aus dem Bild zu entfernen wurde jeder Scan in doppelter Lage ausgeführt.

4.2.2 Auflösung

Der Scanner liefert je nach den gewählten Einstellungen eine Auflösung von 3 mm, 6 mm oder 12mm auf 10m. In dem Versuch wurde die höchste Auflösung von 12mm gewählt. Diese Genauigkeit kann durch eine Geschwindigkeit von 1.000.000 Punkten pro Sekunde gewährleistet werden.

4.2.3 Entfernung

Der durchschnittliche Abstand des Lasers vom Gerüst betrug zwischen 5 m und 10m. So kann man unter gute Randbedingungen eine Genauigkeit erreichen. Der maximale Messbereich des Lasers liegt zwischen 0,5 m und 130m, wobei bei einem Abstand von 130m eine Genauigkeit von 2cm erreicht wird. In unserem Fall wäre das auf jeden Fall noch ausreichend, da auch der spätere Abgleich mit einer größeren Toleranz durchgeführt wird. Im laufenden Baustellenbetrieb ist es aufgrund von schlechter Zugänglichkeit oft nötig aus größerer Distanz zu messen.

4.2.4 Oberflächenfarbe des Modells

Die Farbe der Gerüstrohre ist am ehesten als grau zu beschreiben und eignet sich aufgrund seiner guten Reflektivität für eine Messung mit Laserstrahlen. Ein weiterer Faktor, der zur Messgenauigkeit beiträgt, ist der metallische Glanz. Viele Gerüste enthalten jedoch Holzbauteile wie Bordbretter oder Beläge. Diese sind opak und daher weniger geeignet Laserstrahlen zu reflektieren.

4.2.5 Zugänglichkeit und Verdeckungen

Der im Versuch gewählte Aufbau ist freistehend und weist keine Verdeckungen durch Planen oder andere Hilfskonstruktionen auf. Lediglich Selbstverdeckungen des Systems können nicht vermieden werden. Durch einen strategischen Scan Plan konnte

dieser Effekt teilweise reduziert werden. Dafür wurden die Scans in einem geschlossenen Halbkreis um den Gerüstaufbau angeordnet. Als Abstand zwischen den Standpunkten wurden immer maximal 5m eingehalten. Außerdem wurde darauf geachtet, dass jeder Standpunkt aus in Sichtlinie zu seinem Vorgänger und Nachfolger war, um eine korrekte Registrierung zu gewährleisten. Die Standorte des Laserscanners für die Aufnahmen sind in **Abbildung 29** abgebildet.

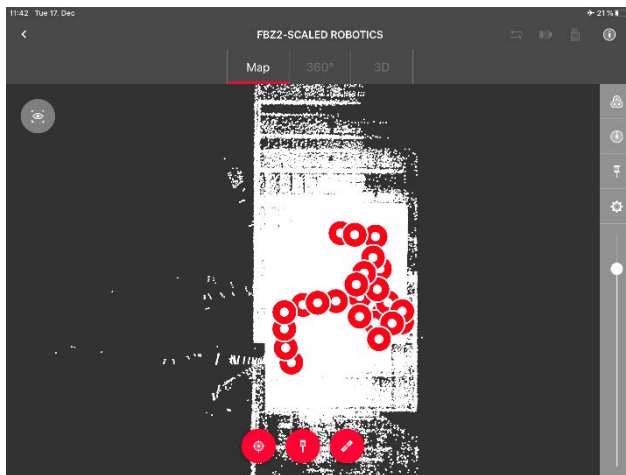


Abbildung 29 Standorte des Laserscanners

4.2.6 Registrierung

Bei großen Punktwolken mit vielen einzelnen Scans stellt die Registrierung und Zusammenführung oft eine große Herausforderung dar. Mit Hilfe von Edge Computing kann dieses Problem gelöst werden. Dafür werden die Stationswechsel in Echtzeit anstatt zu einem zentralen Speicherort zu der mobilen Station übertragen. So wird eine schnellere Verarbeitung der Daten und effizientere Auswertung ermöglicht. (Leica Geosystems AG 2019)

4.2.7 Atmosphäre und Umgebung

Der Versuchsaufbau befindet sich in einer geschlossenen Halle ohne Öffnungen. Man kann also von Laborbedingungen sprechen, da die Aufnahmen ohne Einfluss von Witterung und laufendem Baubetrieb durchgeführt wurde. Die Temperatur liegt circa bei 20°C und somit innerhalb des zulässigen Bereichs von -5°C bis 40°C. Durch sein zertifiziertes Aluminiumgehäuse wäre das Gerät theoretisch auch gegen Flüssigkeiten und Feststoffpartikel geschützt. (Leica Geosystems AG 2019)

4.3 Bearbeitung der Punktwolke

Die somit aufgenommene Punktwolke wurde automatisch auf dem angeschlossenen USB Chip gespeichert und kann schon während dem Aufnahmeprozess auf dem Tablet begutachtet werden. Im Büro wird der USB Chip an den Computer angeschlossen und mit Hilfe der Software Cyclone von Leica importiert. Als Exportformat haben wir uns für .e57 entschieden, da es sich um ein herstellernerutrales Datenformat handelt. Die resultierende Datei wird anschließend in die Punktwolkenbearbeitungssoftware Recap von Autodesk importiert. Es empfiehlt sich das Ausmaß der Punktwolke so weit wie möglich zu verkleinern, um die zu handhabende Datenmenge zu reduzieren. Die auf ein Minimum zugeschnittene Punktwolke wird in **Abbildung 31** dargestellt und beinhaltet lediglich die Gerüstkonstruktion.

Beim Export von Punktwolken aus Recap ist die Standardauflösung 5mm zwischen den einzelnen Punkten. Hierdurch entstehen riesige Datenvolumen mit 5×10^9 Messpunkten, die oft zu Problemen bei der Speicherung und Berechnung führen. Da meistens nur einige wenige Punkte zur Bestimmung einer Fläche benötigt werden kann die Anzahl der Punkte drastisch reduziert werden. Dieser Schritt wurde in diesem Versuch bei dem Export aus Recap durchgeführt. Dabei kann wie in **Abbildung 30** zu sehen nach dem wählen des Speicherortes und des Dateinamens ein Punktabstand zwischen 1mm und 100mm gewählt werden.

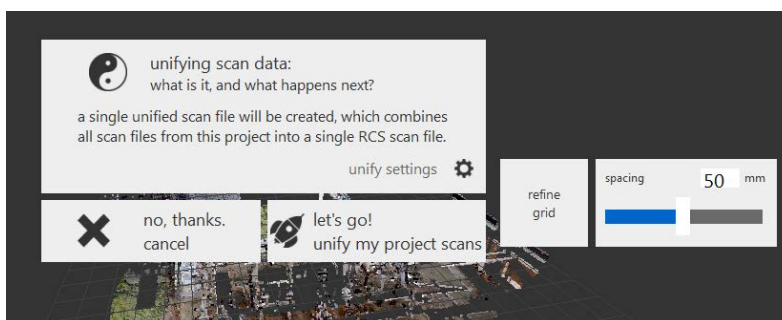


Abbildung 30 Recap-Einstellung Punktwolkenintensität

In den folgenden Abbildungen werden die Punktwolken mit den verschiedenen Auflösungen gezeigt. Die erste hat dabei den Standardabstand von 5mm, die zweite 15mm, die dritte 25m und die vierte 50mm.

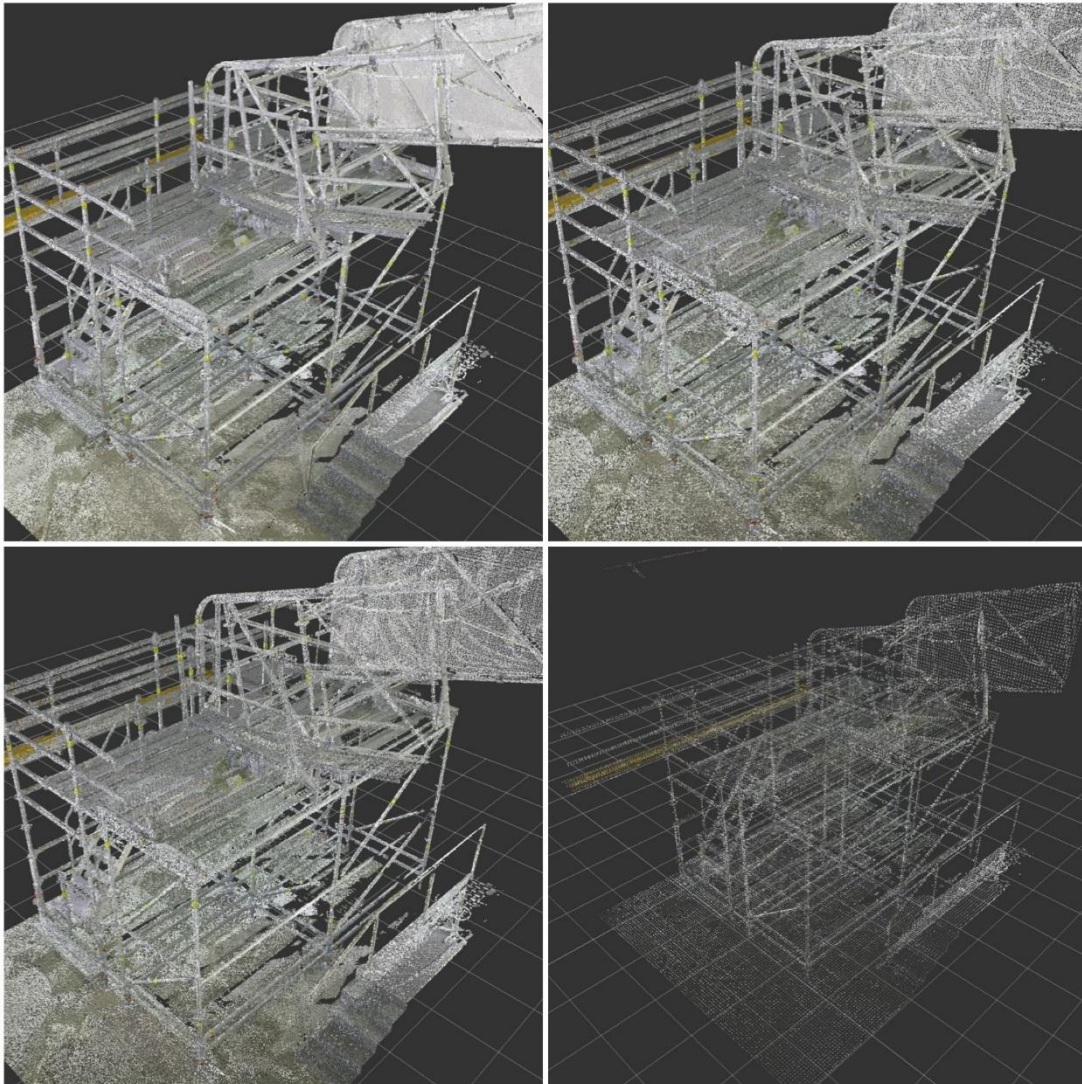


Abbildung 31 Punktwolke in den Auflösungen 5 mm, 15 mm, 25 mm und 50 mm

Bei genauer Betrachtung sind die optischen Unterschiede zwischen den Punktwolken deutlich erkennbar. Der Schritt der Datenreduzierung bei der Punktwolke ist besonders wichtig, wenn die Berechnungen an Standardcomputern mit einem üblichen Arbeitsspeicher von circa 16GB und einer handelsüblichen Grafikkarte von 4GPU durchgeführt werden soll. Denn die Experimente haben ergeben, dass im schlechtesten Fall Unterschiede in der Berechnungszeit von mehreren Stunden entstehen, die fast die gesamte Kapazität des Rechners beanspruchen. (Johnson 2018)

Um die Punktwolke in Revit importieren zu können muss sie aus Recap in den Formaten .rcs oder .rcp exportiert werden. Da die Punktwolke aus einzelnen miteinander kombinierten Scans besteht wurde in diesem Fall das Format .rcp gewählt. Dabei werden die einzelnen Punktwolken der verschiedenen Standpunkte zu einer großen

Punktwolke zusammengefügt und werden ab diesem Schritt von der Software wie ein einzelnes Objekt gehandhabt.

Für den Import der Punktwolke in die Modellierungssoftware Revit für die Überlappung mit dem 3D Modell des Gerüstaufbaus gibt es eine Reihe von Möglichkeiten zur Platzierung:

- Mittelpunkt zu Mittelpunkt (dabei werden die Mittelpunkte der Bounding Boxes übereinander platziert)
- Ursprung zu Ursprung (dafür muss vor dem Import in Revit in Recap ein Koordinatensystem für die Punktwolke definiert werden)
- Gemeinsame Koordinaten (hierbei muss ebenfalls ein Koordinatensystem in Recap definiert werden und zwei entsprechende Koordinaten ausgewählt werden)

Bei diesem Versuch wurde die Methode Mittelpunkt zu Mittelpunkt gewählt, was aufgrund der abweichenden Größe von Punktwolke und Modell zu einer leichten Abweichung führt. Das Resultat ist in **Abbildung 32** dargestellt. Diese Anpassung muss manuell ausgeführt werden, um die Punktwolke exakt mit dem 3D-Modell zu überlappen. Das Resultat davon ist in **Abbildung 33** zu sehen.

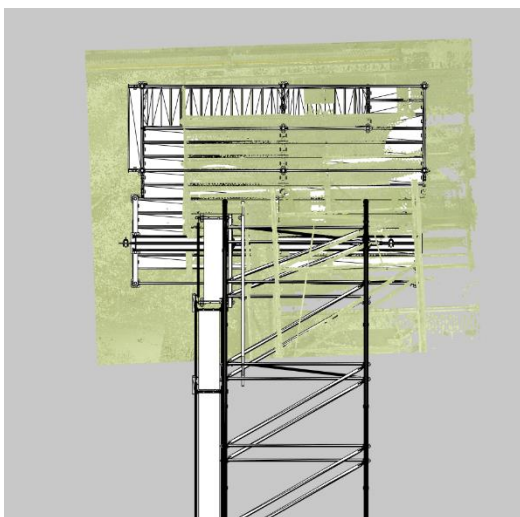


Abbildung 32 Einpassen der Punktwolke

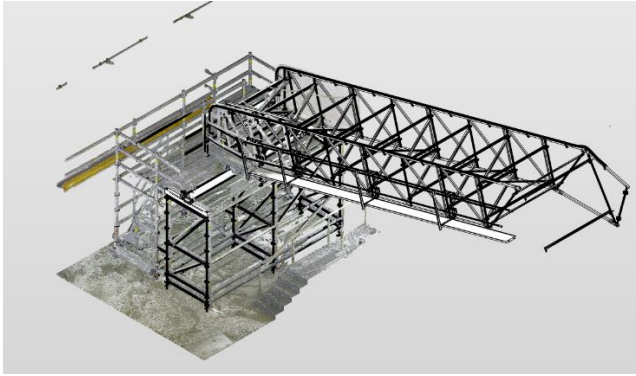


Abbildung 33 Eingepasste Punktwolke mit Modell

4.4 Software zum Scan-vs-BIM Abgleich

4.4.1 Autodesk Point Layout

Bei der Software Point Layout handelt es sich um ein herstellereigenes Plugin von Autodesk für die Modellierungssoftware Revit. Die Hauptfunktion dabei ist das Abbilden der Baustellenbedingungen in Revit. Beispiele dafür sind abgesteckte Punkte, Koordinatensysteme auf der Baustelle und das Analysieren von Abweichungen zwischen Modell und Punktwolke.

Ziel des Versuches mit der Autodesk Point Layout Software war es die Abweichungen zwischen dem Gerüstmodell und der Punktwolke zu finden, welche durch eine Farbskala quantifiziert werden. Dabei wurde untersucht, welche Parameter für eine erfolgreiche und aussagekräftige Versuchsdurchführung ausschlaggebend sind und welche Gewichtung diese haben.

Der erste Versuchsdurchgang wurde mit den Grundeinstellungen von Point Layout durchgeführt, welche in **Abbildung 34** dargestellt sind.

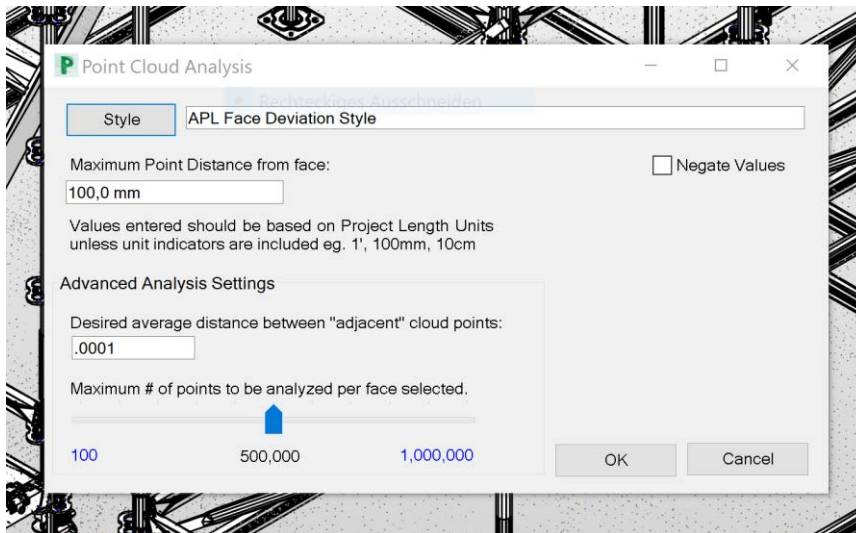


Abbildung 34 Grundeinstellungen bei der Versuchsdurchführung

Die erste Variable ist der maximale Abstand der Punkte von der Teilfläche, welche bei der Grundeinstellung 100 mm beträgt. Die zweite Variable ist die gewünschte maximale Dichte der Punktwolke. Diese ist in diesem Fall irrelevant, da die verwendete Punktwolke eine viel geringere Dichte von höchstens 5 mm besitzt. Die dritte Variable beschreibt die maximale Anzahl an Punkten, die pro Einzelfläche analysiert werden soll. In den Standardeinstellungen liegt diese wie in **Abbildung 34** bei 500.000 Punkten pro Fläche.

Im ersten Versuchsdurchlauf wird die Punktwolke mit einer Dichte von 5 mm gewählt. Da sich viele der Gerüstkomponenten gegenseitig okkludieren, kann nur der momentan sichtbare Teil der Elemente selektiert werden. Nach Start der Berechnung fällt die extrem lange Berechnungszeit auf. Diese wurde erst nach einem Tag und einer Fehlermeldung ohne Ergebnis abgebrochen.

Im darauffolgenden Versuch wurde die Punktwolkendichte drastisch verringert mit der Intention die Berechnungszeit zu verkürzen. Wie in **Abbildung 31** dargestellt handelt es sich dabei um Abstände zwischen den Punkten von 15 mm, 25 mm und 50 mm. Nach einem äquivalenten Versuchsdurchlauf konnte auch hier die Berechnungszeit nicht nennenswert reduziert werden.

Im nächsten Iterationsschritt wurden die Gerüstelemente auf ihre Geeignetheit untersucht, da das Programm hauptsächlich für großflächige Betonformen im Hochbau konzipiert wurde. Die Analyse der Bauteile ergab, dass es sich wie in **Abbildung 35** dargestellt um insgesamt 326 Bauelemente handelt.

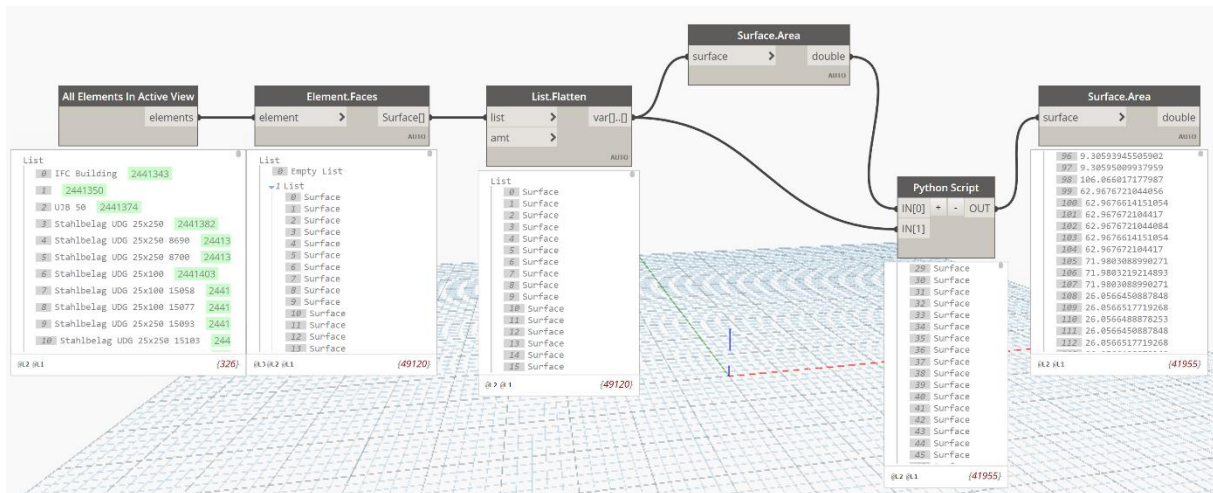


Abbildung 35 Berechnung der Oberflächen der Einzelemente in Dynamio

Diese bestehen wiederum wie in **Abbildung 35** gezeigt aus über 49.000 Einzelflächen, die bei jedem Durchgang einzeln berechnet werden müssen. Nach Kontaktaufnahme mit der Firma Autodesk und deren eigenen Versuchen wurde bestätigt, dass dies die Rechenleistung von herkömmlichen Rechnern übersteigt und eine Option für das Festlegen von Untergrenzen für die Flächen eingebaut werden sollte. Dies wird jedoch erst wie in **Abbildung 36** dargestellt ab der Point Layout Version 2021 zur Verfügung stehen.

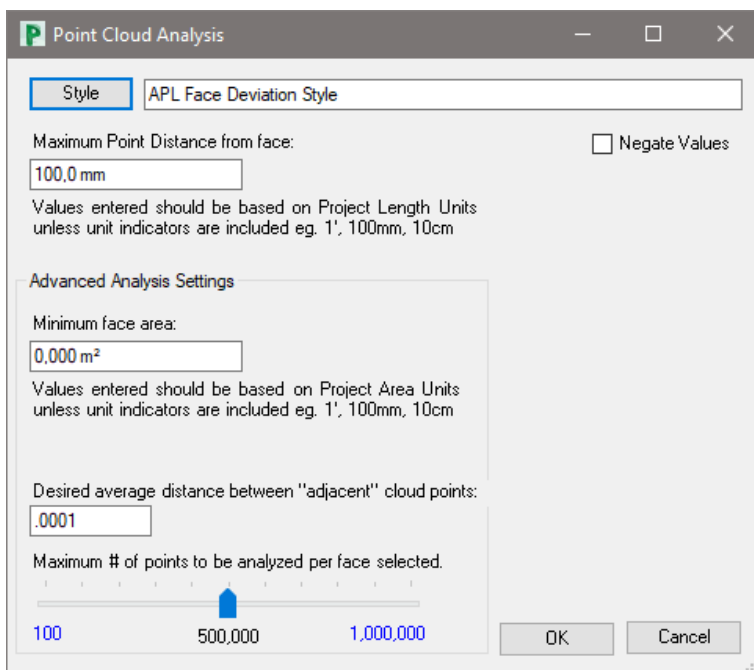


Abbildung 36 Autodesk Point Layout Release 2021

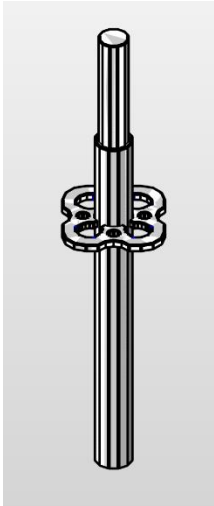


Abbildung 37 Gerüststiel mit Rosette

Schon ein kurzer Stiel (**Abbildung 37**) mit einer kleeblattförmigen Rosette besitzt 221 Einzelflächen. Der Flächeninhalt der kleinsten davon beträgt lediglich $2,1 \cdot 10^{-6} \text{ m}^2$ und der größte beträgt $0,0086 \text{ m}^2$. Dieser große Unterschied in den Flächeninhalten lässt auf eine große Divergenz, bezüglich den Größen der Flächen schließen. In **Abbildung 38** ist die genaue Verteilung der Flächeninhalte des Stiels und der Rosette in einem Histogramm dargestellt. Da die Flächen unter einer Größe von 10^{-5} hauptsächlich zur inneren Rundung der Rosette gehören könnten diese bei einer Analyse des Abstandes der Punkte zum Modell vernachlässigt und somit die Berechnungsgeschwindigkeit erhöht werden.

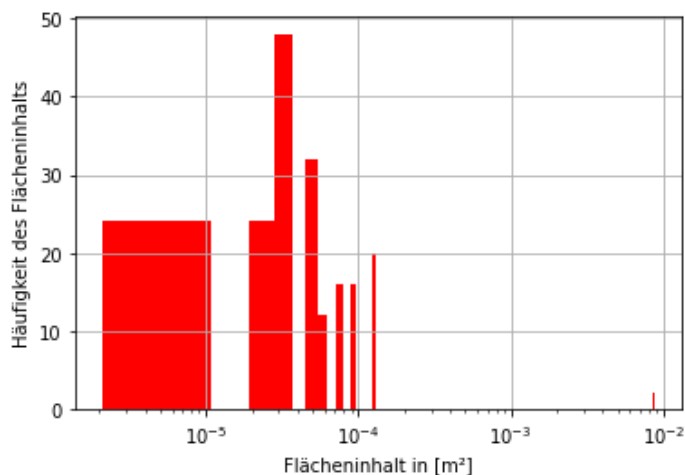


Abbildung 38 Histogramm mit Verteilung der Flächeninhalte des Stiels mit Rosette

Eine sehr ähnliche Verteilung der Flächeninhalte ist bei der Analyse der Teilflächen des gesamten Gerüstmodells in **Abbildung 39** zu finden. Wie in **Abbildung 40** ersichtlich legt die Verteilung nahe, dass man alle Flächen unter einer Größe von einem $0,001 \text{ m}^2$ aus der Abstandsberechnung ausgeschlossen werden können. Die kumulierte Fläche aller Teilelemente beträgt $80,5 \text{ m}^2$ wohingegen die Fläche der Teilelemente unter $0,001 \text{ m}^2$ mit $3,7 \text{ m}^2$ und einem Anteil von $4,6 \%$ verschwindend gering ist. Die Flächen unter $0,001 \text{ m}^2$ summieren sich insgesamt auf 44.448 Teilelemente. So machen sie fast 90% aller Teilflächen aus. Dies stellt eine gute Möglichkeit zur Einsparung von Rechenleistung dar und wird es ab der Point Layout Version 2021 ermög-

lichen große Gerüstmodelle in kurzer Zeit mit einer Punktwolke abzugleichen. Die Auswertung der Daten für den Versuch erfolgte unter anderem mit dem im **Listing A.1.** dargestellten Python Code.

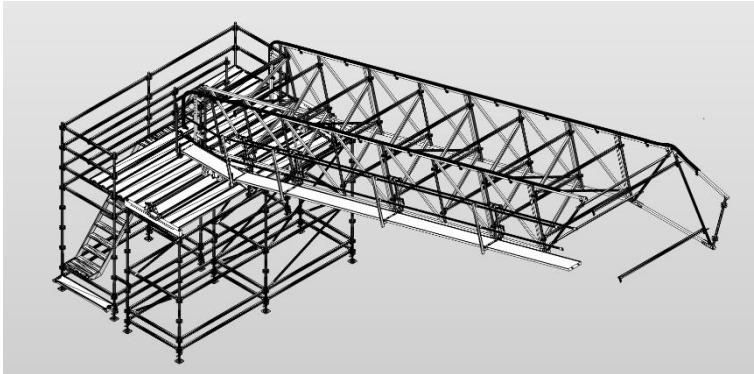


Abbildung 39 3D Modell des gesamten Gerüsts

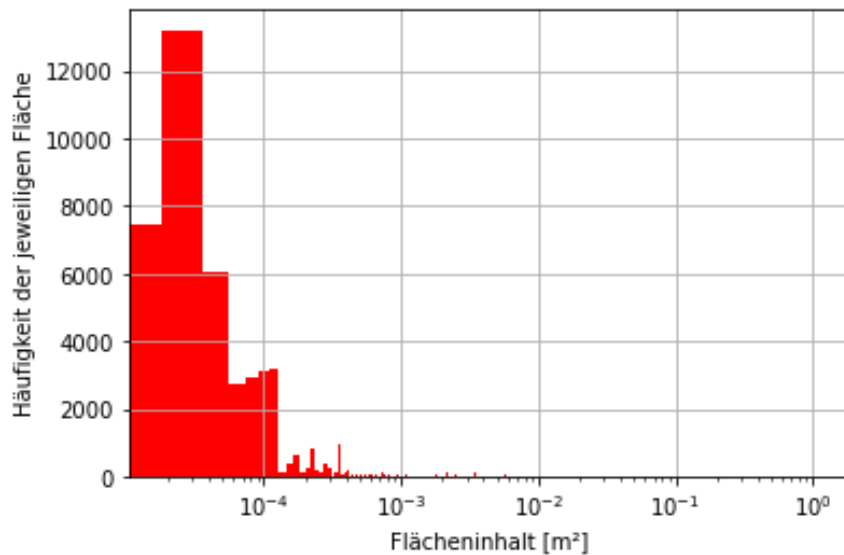


Abbildung 40 Histogramm über die Teilflächeninhalte des gesamten Modells

(Tammik 2020)

(Kerscher 2020)

(Revit 2020)

Listing A.1. Plotten des Histogramms

```
1. import matplotlib.pyplot as plt
2. import pandas as pd
3.
4. flaechen = pd.read_csv('Flächen_ganzes_Modell.csv')
```

```
5.
6. lst = flaechen.values.tolist()
7.
8. clean = []
9.
10. for i in range(len(lst)):
11.     clean_ = lst[i][0]
12.     clean.append(clean_)
13.
14. clean.sort()
15.
16. cleanmeter=[]
17.
18. for i in range(len(clean)):
19.     cleanmeter_ = float(clean[i]) * (10**(-6))
20.     cleanmeter.append(cleanmeter_)
21.
22. n = len(cleanmeter)
23.
24. plt.hist(cleanmeter, facecolor = 'red', bins=60000)
25.
26. plt.xscale('log')
27.
28. #Style
29. plt.ylabel('Häufigkeit der jeweiligen Fläche')
30. plt.xlabel('Flächeninhalt [m²]')
31. plt.grid(True)
32.
33. plt.show()
```

Zwar ist die Berechnung von großen Gerüstmodellen und Punktwolken derzeit aufgrund des großen Rechenaufwandes nur sehr schwer möglich, jedoch können lokal sehr genaue Analysen über den Abstand zwischen Modell und Punktwolke durchgeführt werden. In **Abbildung 41** ist das Ergebnis der Analyse von einem Gerüststiel und des entsprechenden Teils der Punktwolke dargestellt. Der grüne Farbbereich bedeutet dabei eine sehr genaue Übereinstimmung von Modell und Punktwolke, wohingegen die blaue Färbung eine Abweichung im Bereich zwischen 30 und 50 mm darstellt. Insgesamt dauerte die Analyse des einen Stiels circa 5 Minuten. Die Punktwolke wurde aus Gründen der Anschaulichkeit in dieser Darstellung ausgeblendet. Im Anschluss an die Analyse wurde eine Liste mit den Teilflächen ausgegeben, die außerhalb der Toleranzgrenze für die Distanz zwischen Punktwolke und Modell liegen. Diese ist in **Abbildung 42** zu sehen. In diesem Fall wurde die Toleranzgrenze auf 50 mm festgelegt, weshalb auch die Werte auf der Farbskala eine Spanne zwischen +50 mm und -50 mm besitzen.

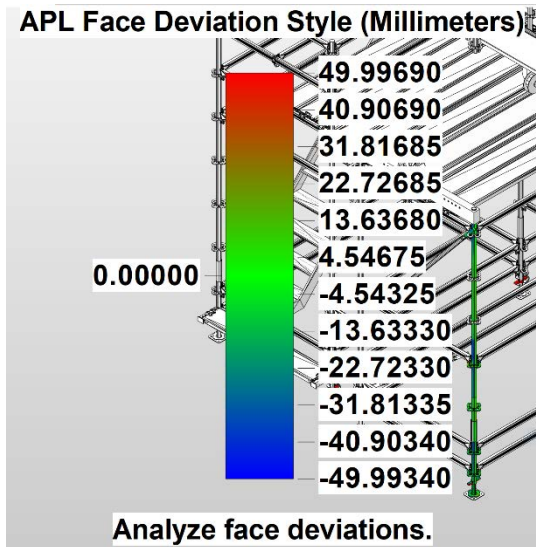


Abbildung 41 Farbskala als Ergebnis der Distanzanalyse

Face # 5: All points fell outside the maximum deviation of 50,0 mm .
 Face # 6: All points fell outside the maximum deviation of 50,0 mm .
 Face # 7: All points fell outside the maximum deviation of 50,0 mm .
 Face # 16: All points fell outside the maximum deviation of 50,0 mm .
 Face # 17: All points fell outside the maximum deviation of 50,0 mm .
 Face # 18: All points fell outside the maximum deviation of 50,0 mm .
 Face # 20: All points fell outside the maximum deviation of 50,0 mm .
 Face # 21: All points fell outside the maximum deviation of 50,0 mm .
 Face # 22: All points fell outside the maximum deviation of 50,0 mm .
 Face # 34: All points fell outside the maximum deviation of 50,0 mm .
 Face # 39: All points fell outside the maximum deviation of 50,0 mm .
 Face # 40: All points fell outside the maximum deviation of 50,0 mm .
 Face # 45: All points fell outside the maximum deviation of 50,0 mm .
 Face # 46: All points fell outside the maximum deviation of 50,0 mm .
 Face # 76: All points fell outside the maximum deviation of 50,0 mm .
 Face # 80: All points fell outside the maximum deviation of 50,0 mm .
 Face # 83: All points fell outside the maximum deviation of 50,0 mm .
 Face # 84: All points fell outside the maximum deviation of 50,0 mm .
 Face # 112: All points fell outside the maximum deviation of 50,0 mm .
 Face # 114: All points fell outside the maximum deviation of 50,0 mm .
 Face # 115: All points fell outside the maximum deviation of 50,0 mm .
 Face # 116: All points fell outside the maximum deviation of 50,0 mm .
 Face # 118: All points fell outside the maximum deviation of 50,0 mm .
 Face # 119: All points fell outside the maximum deviation of 50,0 mm .
 Face # 120: All points fell outside the maximum deviation of 50,0 mm .
 Face # 121: All points fell outside the maximum deviation of 50,0 mm .
 Face # 132: All points fell outside the maximum deviation of 50,0 mm .
 Face # 133: All points fell outside the maximum deviation of 50,0 mm .
 Face # 134: All points fell outside the maximum deviation of 50,0 mm .
 Face # 135: All points fell outside the maximum deviation of 50,0 mm .
 Face # 148: All points fell outside the maximum deviation of 50,0 mm .
 Face # 162: All points fell outside the maximum deviation of 50,0 mm .
 Face # 163: All points fell outside the maximum deviation of 50,0 mm .
 Face # 164: All points fell outside the maximum deviation of 50,0 mm .
 Face # 184: All points fell outside the maximum deviation of 50,0 mm .
 Face # 185: All points fell outside the maximum deviation of 50,0 mm .
 Face # 190: All points fell outside the maximum deviation of 50,0 mm .
 Face # 191: All points fell outside the maximum deviation of 50,0 mm .
 Face # 195: All points fell outside the maximum deviation of 50,0 mm .
 Face # 196: All points fell outside the maximum deviation of 50,0 mm .
 Face # 197: All points fell outside the maximum deviation of 50,0 mm .
 Face # 198: All points fell outside the maximum deviation of 50,0 mm .
 Face # 199: All points fell outside the maximum deviation of 50,0 mm .
 Face # 200: All points fell outside the maximum deviation of 50,0 mm .
 Face # 201: All points fell outside the maximum deviation of 50,0 mm .

Abbildung 42 Liste mir Teilflächen außerhalb der Toleranzgrenze

4.4.2 Python

Der folgende Abschnitt beschreibt, die für diese Arbeit entwickelte Vorgehensweise, um den Baufortschritt eines Gerüstes mit Hilfe der Programmiersprache Python durchzuführen. Die dafür verwendete Entwicklungsumgebung ist Spyder.

Im ersten Schritt wurde aus dem gesamten Revit Modell eine Stereolithographie (STL) Datei erzeugt. Dies erfolgte wie in Fehler! Verweisquelle konnte nicht gefunden werden.³ zu sehen durch einen Export aus Revit 2021. Als Format wurde dabei ASCII gewählt, um später in einem Texteditor auf die Daten zugreifen und gegebenenfalls Anpassungen vornehmen zu können. Unter der Kategorie Auflösung wurde die Option „Grob“ gewählt, um vorab die Datenmenge zu reduzieren und die Bearbeitungsgeschwindigkeit zu erhöhen.

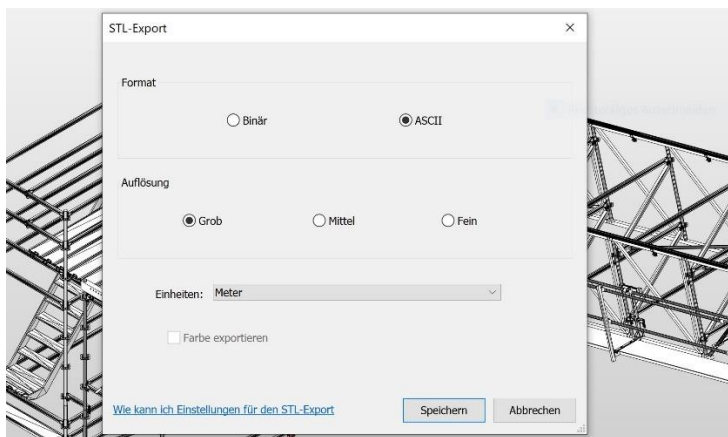


Abbildung 43 Einstellungen für den STL Export in Revit

Da man beim STL Export aus Revit 2021 lediglich sehr rudimentäre Anpassungen bezüglich der Anzahl der resultierenden STL Flächen vornehmen kann muss die Datei im Anschluss in Meshlab importiert werden. Wie im vorangehenden Abschnitt 4.4.1 erläutert ist die Anzahl der STL Flächen sehr hoch, wobei nur ein geringer Teil für eine sinnvolle geometrische Abbildung benötigt wird. Die Bearbeitung des Meshes erfolgt mit den in den **Abbildung 44** und **Abbildung 45** dargestellten Filtereinstellungen.

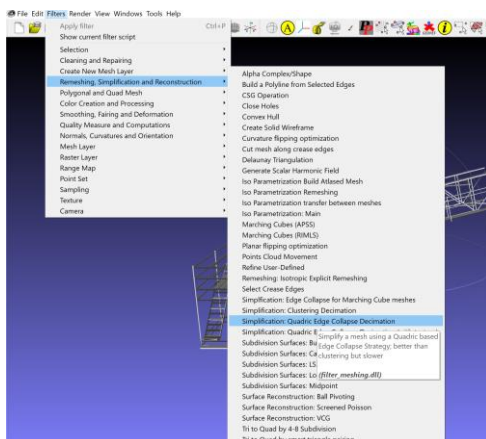


Abbildung 44 Filter für die Reduzierung der STL Flächen in Meshlab

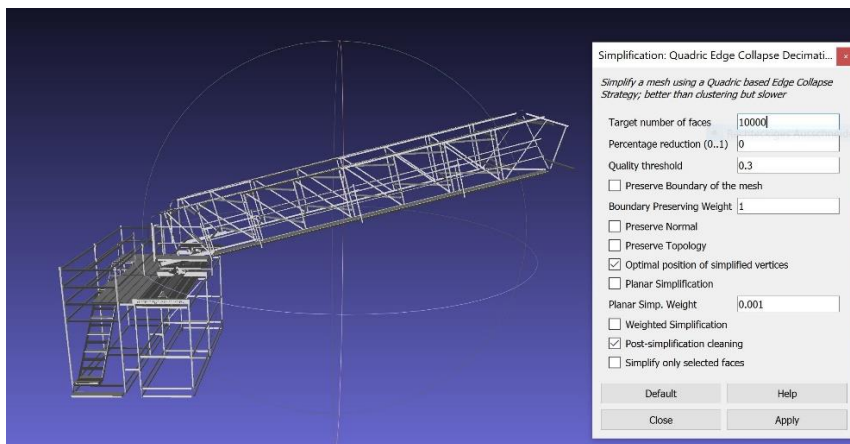


Abbildung 45 Einstellungen für die Reduzierung der STL Flächen in Meshlab

Als Ergebnis erhält man ein Mesh mit einer variablen Anzahl (in diesem Beispiel 8.000) an STL Flächen im Dateiformat .stl. Ein Ausschnitt davon ist in **Abbildung 46** zu betrachten. Der Aufbau der Datei besteht aus einem Header „solid ASCII“, den Normalenvektoren, den 3D Punkten und der Schlusszeile „endsolid vcg“.

```
solid ASCII
facet normal 1.000000e+00 0.000000e+00 0.000000e+00
  outer loop
    vertex 3.382256e+01 1.270219e+01 3.113363e+00
    vertex 3.382256e+01 1.270219e+01 3.100673e+00
    vertex 3.382256e+01 1.513919e+01 3.100673e+00
  endloop
endfacet
facet normal 1.000000e+00 0.000000e+00 -0.000000e+00
  outer loop
    vertex 3.382256e+01 1.513919e+01 3.100673e+00
    vertex 3.382256e+01 1.513919e+01 3.113363e+00
    vertex 3.382256e+01 1.270219e+01 3.113363e+00
  endloop
endfacet
facet normal 0.000000e+00 0.000000e+00 -1.000000e+00
  outer loop
    vertex 3.380256e+01 1.513919e+01 3.113363e+00
    vertex 3.382256e+01 1.513919e+01 3.113363e+00
    vertex 3.382256e+01 1.270219e+01 3.113363e+00
  endloop
endfacet
facet normal 0.000000e+00 0.000000e+00 -1.000000e+00
  outer loop
    vertex 3.382256e+01 1.270219e+01 3.113363e+00
    vertex 3.380256e+01 1.270219e+01 3.113363e+00
    vertex 3.380256e+01 1.513919e+01 3.113363e+00
  endloop
endfacet
```

Abbildung 46 Auszug aus einer der verwendeten STL Dateien

Die Punktwolke wurde mit CloudCompare bearbeitet und in den Dichten von 5 mm bis 50 mm in fünf Schritten exportiert. Für die Verarbeitung in Python wurde sie in dem Format .pts exportiert.

Im Folgenden wird die Funktionsweise des Python Codes erläutert. Dieser wurde dafür in einzelne Abschnitte unterteilt, die der Reihe nach analysiert werden.

Listing A.2.1. Distanzberechnung

```
1. from stl import mesh
2. from mpl_toolkits import mplot3d
3. from matplotlib import pyplot
4. import trimesh
5. import trimesh.points
6. import matplotlib.pyplot as plt
7.
8.
9. # STL importieren
10. stl = open(r'C:\Users\Caroline Kerscher\Desktop\color_dis-
    tance_matlab\STL_3_8000.txt' , 'r')
11.
12. # Mesh plotten
13. figure = pyplot.figure()
14. axes = mplot3d.Axes3D(figure)
15. your_mesh = mesh.Mesh.from_file(r'C:\Users\Caroline Kerscher\Desktop\color_dis-
    tance_matlab\STL_3_8000.txt')
16. axes.add_collection3d(mplot3d.art3d.Poly3DCollection(your_mesh.vectors))
17. scale = your_mesh.points.flatten(-1)
18. axes.auto_scale_xyz(scale, scale, scale)
```

Wie im Listing A.2.1. dargestellt müssen zuerst die Rohdaten für das Mesh in Python importiert und dann rekonstruiert werden. Anschließend wird ein leeres Koordinatensystem erzeugt, das sich durch die `auto_scale_xyz()` Funktion automatisch, an den Zoom auf das Modell anpasst.

Listing A.2.2. Distanzberechnung

```
1. # PW importieren
2. pts_datei = open(r'C:\Users\Caroline Kerscher\Desktop\color_dis-
    tance_matlab\cc_5mm.txt')
3. pts = pts_datei.read()
4. pts_lst = pts.split('\n')
5.
6. x = []
7. y = []
8. z = []
9.
10. trenn = []
11. for i in range(len(pts_lst)):
12.     trenn_ = pts_lst[i].split(' ')
13.     trenn.append(trenn_)
14.
15. for i in range(len(trenn)-2):
16.     x_ = trenn[i][0]
17.     x__ = float(x_) + 32.5
18.     x.append(x__)
19.
20. for i in range(len(trenn)-2):
21.     y_ = trenn[i][1]
22.     y__ = float(y_) + 10
23.     y.append(y__)
24.
25. for i in range(len(trenn)-2):
26.     z_ = trenn[i][2]
27.     z__ = float(z_)
28.     z.append(z__)
29.
```

```

30. points = []
31.
32. for i in range(len(x)):
33.     points_ = [x[i], y[i], z[i]]
34.     points.append(points_)
35.
36. # Beschriftung der Axen
37. axes.set_xlabel('x')
38. axes.set_ylabel('y')
39. axes.set_zlabel('z')
40.
41. # Zoomen auf Axe
42. axes.set_xlim(30, 45)
43. axes.set_ylim(10, 16)
44. axes.set_zlim(0, 10)
45.
46. axes.scatter(x, y, z, c = 'r', marker = '.', s = 0.1, alpha = 0.3)
47. axes.view_init(elev = 20, azim = 45)

```

Im Anschluss daran wird die Datei mit den Koordinaten für die Punktwolke geöffnet. Da diese im .pts Format gespeichert wurde ist sie leicht lesbar und zugreifbar. So kann die gesamte Datei nach Zeilen getrennt, und alle Koordinaten in einer Liste gespeichert werden. Die addierten Zahlen auf die x- und y-Koordinate dienen zur Translation und damit Überlagerung mit der STL Datei. Durch die mit „axes.“ beginnenden Befehle werden die Eigenschaften des Koordinatensystems spezifiziert. Dazu zählen die Beschriftung der Achsen, die Wahl des Markers für die Punkte, die Transparenz der Punktwolke und der im Plot dargestellte Winkel.

Listing A.2.3. Distanzberechnung

```

1. mesh = trimesh.load(r'C:\Users\Caroline Kerscher\Desktop\color_distance_matlab\STL_3_8000.stl')
2.
3. res = trimesh.proximity.closest_point(mesh, points)
4.
5. PFacet = res[0]
6. dist = res[1]
7. facetID = res[2]
8.
9. sSet = list(set(facetID))
10.
11. ##### STL nach Distanz kürzen
12. schwelle_distanz = 0.05
13.
14. dist_kurz = []
15. IDkurz = []
16. Pkurz = []
17.
18. for i in range(len(PFacet)):
19.     if dist[i] < schwelle_distanz:
20.         dist_kurz_ = dist[i]
21.         dist_kurz.append(dist_kurz_)
22.         IDkurz_ = facetID[i]
23.         IDkurz.append(IDkurz_)
24.         Pkurz_ = PFacet[i]
25.         Pkurz.append(Pkurz_)
26.
27. dist_weit = []
28. IDweit = []
29. Pweit = []
30.
31. for i in range(len(PFacet)):

```

```

32.     if dist[i] > schwelle_distanz:
33.         dist_weit_ = dist[i]
34.         dist_weit_.append(dist_weit_)
35.         IDweit_ = facetID[i]
36.         IDweit_.append(IDweit_)
37.         Pweit_ = PFacet[i]
38.         Pweit_.append(Pweit_)
39. setWeit = set(IDweit)

```

Im darauffolgenden Schritt im **Listing A.2.3** wird der Abstand eines jeden Punktes zu seinem nächstgelegenen STL Flächenstück berechnet. PFacet = [...] ist dabei die Liste der exakten Punkte auf den STL Flächen. Die Liste dist = [...] stellt die entsprechenden Distanzen dar und facetID = [...] speichert die zu jedem Punkt zugeordnete STL Fläche. Anschließend können die Listen, je nach gewünschtem maximalem Abstand (schwelle_distanz) zwischen Punktvolke und STL Datei gefiltert werden.

Listing A.2.4. Distanzberechnung

```

1. ##### nach Anzahl Points pro Facet kürzen
2. Facet_count = []
3. for i in range(len(IDkurz)):
4.     Facet_count_ = list(IDkurz).count(IDkurz[i])
5.     Facet_count.append(Facet_count_)
6.
7. ID_kurz_häufig = []
8.
9. #für kurz, häufig
10. for i in range(len(Facet_count)):
11.     if Facet_count[i] >= 3:
12.         ID_kurz_häufig_ = IDkurz[i]
13.         ID_kurz_häufig.append(ID_kurz_häufig_)
14.
15. set_ID_kurz_häufig = sorted(set(ID_kurz_häufig))
16.
17. #für kurz, wenig
18. ID_kurz_wenig = []
19.
20. for i in range(len(Facet_count)):
21.     if Facet_count[i] < 3:
22.         ID_kurz_wenig_ = IDkurz[i]
23.         ID_kurz_wenig.append(ID_kurz_wenig_)
24.
25. set_ID_kurz_wenig = sorted(set(ID_kurz_wenig))
26.
27. ##### print new .txt file für IDviel
28. file = open(r'C:\Users\Caroline Kerscher\Desktop\color_dis-
29. tance_matlab\STL_3_8000.txt')
29. stl3 = file.read()
30. stltrenn = stl3.split('endfacet')
31.
32. stl_plot = []
33. for i in set_ID_kurz_häufig:
34.     stl_plot_ = stltrenn[i] + 'endfacet'
35.     stl_plot.append(stl_plot_)
36.
37. stl_plot.insert(0, 'solid ASCII')
38. stl_plot.append('\n' + 'endsolid vcg')
39. ''.join(stl_plot)
40.
41. fileW = open(r'C:\Users\Caroline Kerscher\Desktop\color_dis-
42. tance_matlab\ProSTL_5mm_5mm.txt', 'x')
42. fileW.writelines(stl_plot)
43. fileW.close()

```

Der zweite Parameter ist die minimale Anzahl von Punkten pro STL Flächenstück. Dadurch werden alle Flächen IDs, auf die eine Mindestanzahl von Flächenstücken zugeordnet ist in einer Liste gespeichert ($ID_kurz_häufig = []$). Die Flächen IDs, die unter der Mindestanzahl liegen werden in einer weiteren Liste gespeichert ($ID_kurz_wenig = []$). Dies ist im **Listing A.2.4** dargestellt. Im Anschluss werden die Parameter Distanz und die Mindestanzahl an Punkten pro STL Fläche genauer untersucht. Zuerst wird auf den Einfluss der Punktwolkendichte auf die Anzahl der erkannten STL Flächen eingegangen. Die Mindestanzahl der Punkte pro STL Fläche wurde dabei auf drei festgelegt. Dies ist damit zu begründen, dass man für die Definition einer Ebene im immer mindestens drei Punkte im 3D-Raum benötigt. Das Gerüstmodell besteht aus 8000 einzelnen STL Flächen. Theoretisch hätte man auch eine feinere Gliederung wählen können, jedoch handelt es sich bei dem Versuch noch um einen relativ kleines Modell und auf realen Baustellen ist mit viel größeren Datensätzen zu rechnen. Dies könnte bei einem zu hohen Detailgrad zu Problemen bei der Rechengeschwindigkeit führen. **Abbildung 47** und **Abbildung 48** bilden jeweils die Anzahl an erkannten STL Flächen in Abhängigkeit zur verwendeten Punktwolkendichte ab. **Abbildung 47** verwendet dabei eine Toleranzgrenze von 50 mm und **Abbildung 48** eine Toleranzgrenze von 100 mm. Im Durchschnitt werden bei der Toleranzgrenze von 100 mm 31,4 % mehr STL Flächen erkannt. Im Allgemeinen ist der Verlauf der beiden Kurven als linear bis leicht exponentiell abnehmend zu beschreiben.

Anzahl der erkannten Flächen in Abhängigkeit der Punktwolkendichte, bei 50 mm Toleranzgrenze

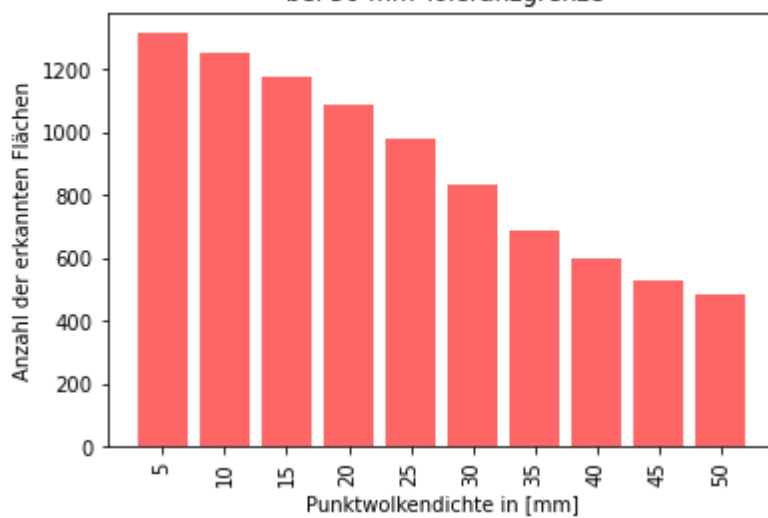


Abbildung 47 Punktwolkendichte in Relation zur Anzahl der erkannten STL Flächen

Anzahl der erkannten Flächen in Abhängigkeit der Punktwolkendichte, bei 100 mm Toleranzgrenze

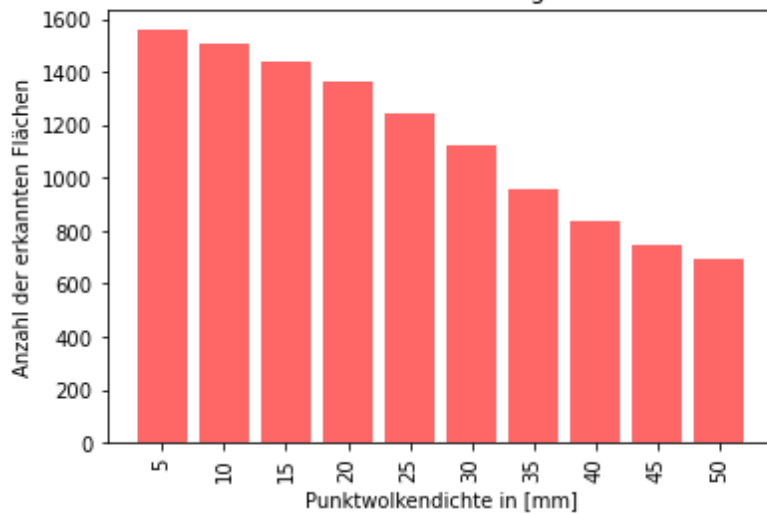


Abbildung 48 Punktwolkendichte in Relation zur Anzahl der erkannten STL Flächen

Wie in **Abbildung 49** dargestellt beinhaltet die Wolke bei einer Dichte von 5 mm 150.476 Messpunkte. Wenn die Dichte auf 50 mm abgesenkt wird verringert sich die Anzahl der Punkte um 84 % auf 23.946 Punkte. Dies ist gleichzusetzen mit einer erheblichen Steigerung der Berechnungsgeschwindigkeit. Überraschenderweise reduziert sich der Inhalt der erkannten Fläche prozentual viel weniger. Bei einer Dichte von 5 mm werden 62,655 m² der Oberfläche des Modells als erkannt registriert. Wenn man die Dichte auf 50 mm reduziert sich die erkannte Fläche nur um 48,5 % auf 32,284 m². Die Flächeninhalte sind in **Abbildung 50** dargestellt.

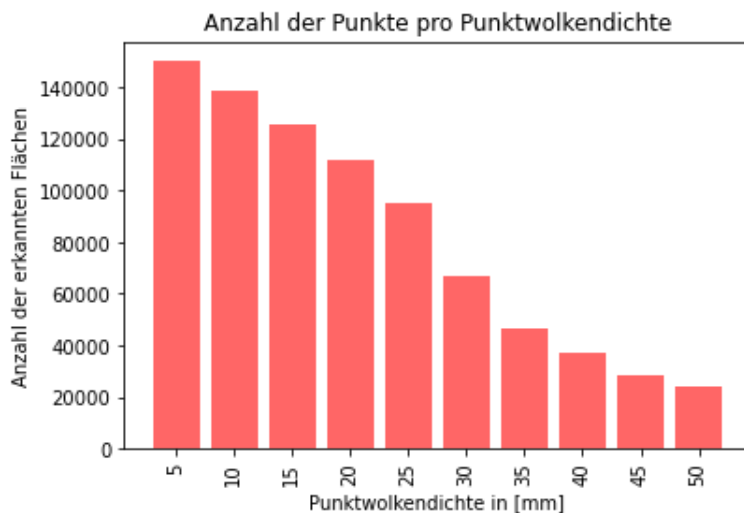


Abbildung 49 Anzahl an Punkten in Relation zur Punktwolkendichte

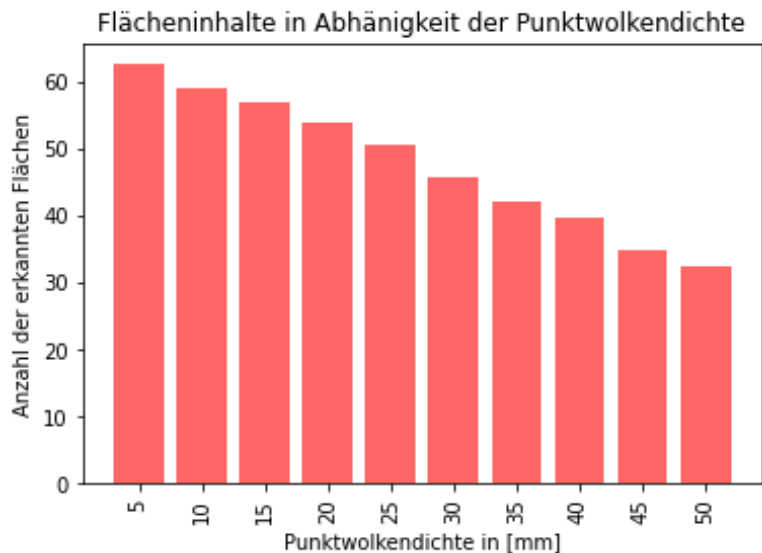


Abbildung 50 Gesamter Flächeninhalt des STL Meshes in Abhängigkeit zur Punktwolkendichte

Das folgende Histogramm (**Abbildung 51**) stellt die Häufigkeit der Anzahl an zugeordneten Punkten pro STL Fläche dar. Dabei wurde ein maximaler Abstand zwischen Punkt und Facet von 50 mm gewählt, bei einer Punktwolkendichte von 5 mm. Der Maximalwert betrug dabei 585 Punkte pro STL Fläche. Der Mittelwert liegt dabei bei 23 Punkten pro STL Fläche.

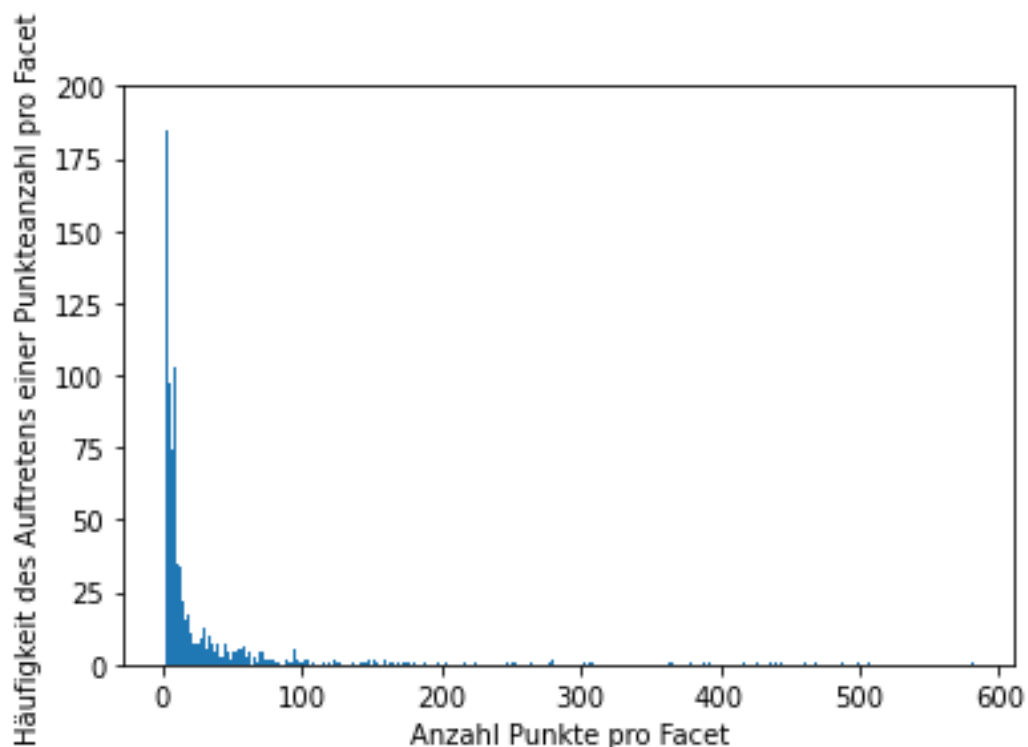


Abbildung 51 Häufigkeit des Auftretens an Punkten pro Facet bei einer Toleranz von 50 mm

Wenn man den höchsten Abstand eines Punktes zu einem Facet auf 100 mm verdoppelt erhält man einen Mittelwert von 30 Punkten pro Facet. Die höchste Anzahl

an Punkte pro Facet liegt dabei bei 594. Im Allgemeinen ist die Kurve annähernd als exponentiell abnehmend zu beschreiben.

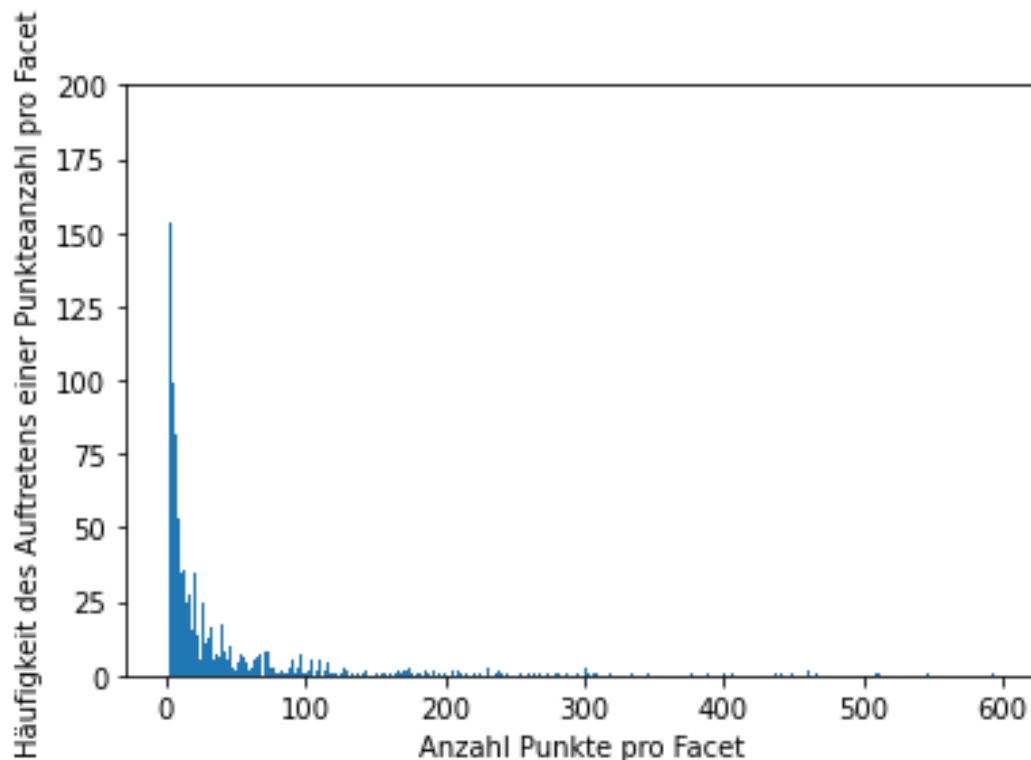


Abbildung 52 Häufigkeit des Auftretens an Punkten pro Facet bei einer Toleranz von 100 mm

Dies bedeutet im Umkehrschluss, wie in **Abbildung 52** dargestellt, dass auch die Anzahl der erkannten STL Flächen mit steigender Mindestanzahl an Punkten je STL, exponentiell abnimmt. Wenn jedoch die STL Teilflächen so gewählt werden, dass sie einen nominal größeren Flächeninhalt besitzen, sollte eine höhere Mindestanzahl von Punkten pro STL gewählt werden um potentielle falsch-positive Zuweisungen zu vermeiden. Falsch-positiv würde in diesem Sinne bedeuten, dass Flächen als „erkannt“ registriert werden, obwohl dies nicht richtig ist. Ein Beispiel dafür wäre eine sehr große Fläche mit drei zufällig darauf projizierten Punkten. Durch eine höhere Mindestanzahl von Punkten kann dieser Fall mit einer höheren Wahrscheinlichkeit vermieden werden.

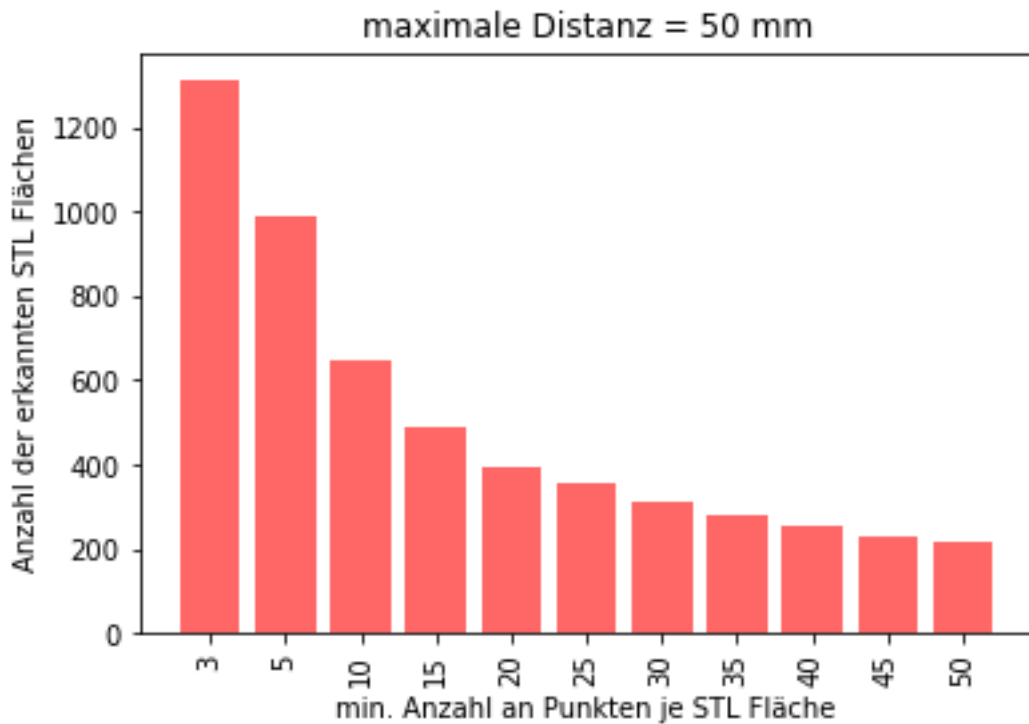


Abbildung 53 Anzahl der erkannten STL Flächen in Bezug zur minimalen Anzahl an Punkten je STL

Die Unterteilung der STL Flächen nach verschiedenen Mindestgrenzen von Punkten kann man auch grafisch im Plot darstellen. Im unten dargestellten Diagramm wurde eine Punktwolkendichte von 5 mm gewählt. Dabei sind alle STL Flächen mit weniger als 6 Punkten je Fläche in pink dargestellt. Die Flächen ab 6 Punkten je STL Fläche sind in grün dargestellt.

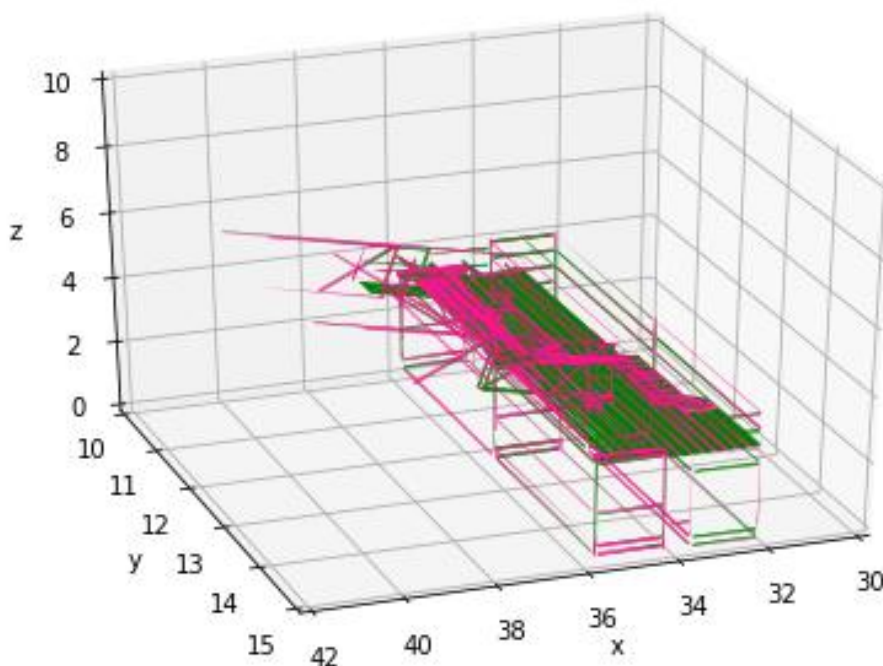


Abbildung 54 Gerüstmodell in Abhängigkeit zur Mindestanzahl an Messpunkten je STL

Die Versuche zeigen, dass eine Variation der Mindestanzahl an Punkten pro Facet einen größeren Einfluss auf die Anzahl der erkannten Teilflächen hat. Dies liegt vor allem an den Funktionstypen der beiden Variablen. Während sich eine Veränderung der Punktwolkendichte linear auf die Anzahl der erkannten Facets auswirkt, hat eine Variation der Mindestanzahl an Punkten pro Facet einen exponentiellen Einfluss darauf.

(Kerscher 2020)

(Tammik 2020)

(Revit 2020)

5 Zusammenfassung und Fazit

Abschließend kann man sagen, dass es auf dem Markt einige Software Lösungen gibt, die Teilbereiche der Herausforderung der Baufortschrittskontrolle im Gerüstbau lösen. Dazu zählt, die in der Arbeit betrachtete Software Autodesk Point Layout, die durch ihren in Revit integrierten und damit effizienten Workflow heraussticht. Nachteilig dabei ist die extrem hohe Rechenleistung, die dafür aufgewendet werden muss und sehr lange Berechnungszeiten zu Folge hat. Außerdem lassen sich im besten Fall nur sehr triviale Ergebnisse daraus ziehen. So muss jede einzelne Stelle des Aufbaus manuell mit der Farbskala verglichen werden, um etwaige Abweichungen zwischen Punktwolke und Modell festzustellen. Außerdem ermöglicht es das Programm nicht eigene statistische Analysen durchzuführen oder die Berechnung an besondere Umstände, wie sehr breit gestreute Flächeninhalte der Teilflächen, anzupassen.

Das für diese Arbeit erstellte Python Programm ermöglicht es durch Messung des Abstandes eines jeden Punktes zu seiner nächstgelegenen STL Fläche den Baufortschritt auf Abweichungen quantitativ und qualitativ zu erfassen. Um die Wahrscheinlichkeit zu erhöhen, dass die STL Flächen richtig erkannt wurden, kann man optional eine Mindestanzahl an Punkten pro STL festlegen. Außerdem lässt sich das Programm um beliebig viele Optionen erweitern und ermöglicht es auf alle Daten direkt zuzugreifen und auszuwerten. Dabei hat die Mindestanzahl an Punkten je STL Fläche einen größeren Einfluss auf die Anzahl der erkannten Flächenstücke als die Punktwolkendichte.

Grundsätzlich kann man sagen, dass in Anbetracht der vielen Spezialfälle in der Baubranche ein System mit leichter Zugänglichkeit und Option zur Variation, wie Python, geeigneter ist als ein System mit sehr beschränkten Funktionen wie Autodesk Point Layout. In Anbetracht der zunehmend strenger werdenden Dokumentationspflicht könnte man das Programm dazu nutzen um den Einbau von notwendigen Sicherheitseinrichtungen, wie Geländer oder Konsolen großflächig nachzuweisen.

Eine weitere Funktion, die von vielen Ingenieuren in der Baubranche gewünscht wird, ist die automatische Erkennung von gesamten Elementen. Dafür könnte der in der Arbeit enthaltene Weg als Ansatz verwendet werden, um mit Hilfe von maschinellem Lernen die einzelnen Elemente des Gerüstes automatisch zu erkennen. Im Fall von

komplexen Projekten in der Industrie mit sehr hohen Ansprüchen an Arbeitssicherheit und Zuverlässigkeit können diese Ansätze sehr gut verwendet werden. Für normale Fassadengerüste sind die oben aufgeführten Methoden in den meisten Fällen zu aufwendig und teuer.

Anhang

Datenpaket-Inhalt

In dem dieser Arbeit beigefügten Datenpaket ist folgender Inhalt, in chronologischer Reihenfolge der Bearbeitung, gespeichert:

- Die vorliegende Bachelorarbeit im PDF-Format
- Der Python Code Abgabe_Pointlayout.py zur Erstellung des Histogramms
- Die aus Revit Modell_Export_STL.rvt
- Die verkleinerte STL Datei STL_8000.stl
- Die Punktwolkendateien cc_5mm.pts bis cc_50mm.pts
- Die STL Datei IDviel_5mm.stl als Ergebnis der Berechnung, die alle STL Flächen innerhalb der Distanzschwelle enthält und eine Mindestanzahl an 3 zugeordneten Punkten enthält
- Die STL Datei IDweit.stl als Ergebnis der Berechnung, die alle STL Flächen überhalb der Distanzschwelle enthält
- Die STL Datei ID kurz.stl als Ergebnis der Berechnung, die alle STL Flächen unterhalb der Distanzschwelle enthält und nicht die Mindestanzahl von 3 zugeordneten Punkten besitzt
- Der Python Code plot_model_different_colors.py

Der Python Code liest die Flächen des Revit Modells ein und plottet deren Flächeninhalte als Histogramm.

Listing A.1. Plotten des Histogramms

```
34. import matplotlib.pyplot as plt
35. import pandas as pd
36.
37. flaechen = pd.read_csv('Flächen_ganzes_Modell.csv')
38.
39. lst = flaechen.values.tolist()
40.
41. clean = []
42.
43. for i in range(len(lst)):
44.     clean_ = lst[i][0]
45.     clean.append(clean_)
46.
47. clean.sort()
48.
49. cleanmeter=[]
50.
51. for i in range(len(clean)):
52.     cleanmeter_ = float(clean[i]) * (10**(-6))
53.     cleanmeter.append(cleanmeter_)
54.
55. n = len(cleanmeter)
56.
57. plt.hist(cleanmeter, facecolor = 'red', bins=60000)
58.
59. plt.xscale('log')
60.
61. #Style
62. plt.ylabel('Häufigkeit der jeweiligen Fläche')
63. plt.xlabel('Flächeninhalt [m²]')
64. plt.grid(True)
65.
66. plt.show()
```

Der Python Code berechnet die Abstände zwischen den Punkten der Punktwolke und der STL Datei. Des weiteren berechnet er für jede STL Fläche die Anzahl, der darauf zugeordneten Punkte.

Listing A.2. Distanzberechnung

```
1. from stl import mesh
2. from mpl_toolkits import mplot3d
3. from matplotlib import pyplot
4. import trimesh
5. import trimesh.points
6. import matplotlib.pyplot as plt
7.
8.
9. # STL importieren
10. stl = open(r'C:\Users\Caroline Kerscher\Desktop\color_dis-
    tance_matlab\STL_3_8000.txt' , 'r')
11.
12. # Mesh plotten
13. figure = pyplot.figure()
14. axes = mplot3d.Axes3D(figure)
15. your_mesh = mesh.Mesh.from_file(r'C:\Users\Caroline Kerscher\Desktop\color_dis-
    tance_matlab\STL_3_8000.txt')
16. axes.add_collection3d(mplot3d.art3d.Poly3DCollection(your_mesh.vectors))
17. scale = your_mesh.points.flatten(-1)
18. axes.auto_scale_xyz(scale, scale, scale)
19. # PW importieren
20. pts_datei = open(r'C:\Users\Caroline Kerscher\Desktop\color_dis-
    tance_matlab\cc_5mm.txt')
21. pts = pts_datei.read()
22. pts_lst = pts.split('\n')
23.
24. x = []
25. y = []
26. z = []
27.
28. trenn = []
29. for i in range(len(pts_lst)):
30.     trenn_ = pts_lst[i].split(' ')
31.     trenn.append(trenn_)
32.
33. for i in range(len(trenn)-2):
34.     x_ = trenn[i][0]
35.     x__ = float(x_) + 32.5
36.     x.append(x__)
37.
38. for i in range(len(trenn)-2):
39.     y_ = trenn[i][1]
40.     y__ = float(y_) + 10
41.     y.append(y__)
42.
43. for i in range(len(trenn)-2):
44.     z_ = trenn[i][2]
45.     z__ = float(z_)
46.     z.append(z__)
47.
48. points = []
49.
50. for i in range(len(x)):
51.     points_ = [x[i], y[i], z[i]]
52.     points.append(points_)
53.
54. # Beschriftung der Axen
55. axes.set_xlabel('x')
```



```
56. axes.set_ylabel('y')
57. axes.set_zlabel('z')
58.
59. # Zoomen auf Axe
60. axes.set_xlim(30, 45)
61. axes.set_ylim(10, 16)
62. axes.set_zlim(0, 10)
63.
64. axes.scatter(x, y, z, c = 'r', marker = '.', s = 0.1, alpha = 0.3)
65. axes.view_init(elev = 20, azim = 45)
66.
67. ##### nach Anzahl Points pro Facet kürzen
68. Facet_count = []
69. for i in range(len(IDkurz)):
70.     Facet_count_ = list(IDkurz).count(IDkurz[i])
71.     Facet_count.append(Facet_count_)
72.
73. ID_kurz_häufig = []
74.
75. #für kurz, häufig
76. for i in range(len(Facet_count)):
77.     if Facet_count[i] >= 3:
78.         ID_kurz_häufig_ = IDkurz[i]
79.         ID_kurz_häufig.append(ID_kurz_häufig_)
80.
81. set_ID_kurz_häufig = sorted(set(ID_kurz_häufig))
82.
83. #für kurz, wenig
84. ID_kurz_wenig = []
85.
86. for i in range(len(Facet_count)):
87.     if Facet_count[i] < 3:
88.         ID_kurz_wenig_ = IDkurz[i]
89.         ID_kurz_wenig.append(ID_kurz_wenig_)
90.
91. set_ID_kurz_wenig = sorted(set(ID_kurz_wenig))
92.
93. ##### print new .txt file für IDviel
94. file = open(r'C:\Users\Caroline Kerscher\Desktop\color_dis-
tance_matlab\STL_3_8000.txt')
95. stl3 = file.read()
96. stltrenn = stl3.split('endfacet')
97.
98. stl_plot = []
99. for i in set_ID_kurz_häufig:
100.     stl_plot_ = stltrenn[i] + 'endfacet'
101.     stl_plot.append(stl_plot_)
102.
103.     stl_plot.insert(0, 'solid ASCII')
104.     stl_plot.append('\n' + 'endsolid vcg')
105.     ''.join(stl_plot)
106.
107.     fileW = open(r'C:\Users\Caroline Kerscher\Desktop\color_dis-
tance_matlab\ProSTL_5mm_5mm.txt', 'x')
108.     fileW.writelines(stl_plot)
109.     fileW.close()
```

Der Python Code plottet die STL Flächen in unterschiedlicher Farbgebung je nach Abstand der Punkte zum Modell und der Anzahl der zugeordneten Punkte je STL Fläche.

Listing A.3. Plotten nach Kategorie der STL Flächen

```
1. from stl import mesh
2. from mpl_toolkits import mplot3d
3. from matplotlib import pyplot
4. import vtkplotlib as vpl
5. import stl
6. import trimesh
7.
8. # Create a new plot
9. figure = pyplot.figure()
10. axes = mplot3d.Axes3D(figure)
11.
12.
13. #mesh aus IDviel
14. your_mesh = mesh.Mesh.from_file(r'C:\Users\Caroline Kerscher\Desktop\color_distance_matlab\IDviel_5mm.txt')
15. axes.add_collection3d(mplot3d.art3d.Poly3DCollection(your_mesh.vectors, facecolors = 'green'))
16.
17. #mesh aus IDkurz
18. mesh_kurz = mesh.Mesh.from_file(r'C:\Users\Caroline Kerscher\Desktop\color_distance_matlab\IDkurz.txt')
19. axes.add_collection3d(mplot3d.art3d.Poly3DCollection(mesh_kurz.vectors, facecolors = 'deeppink'))
20.
21. #mesh aus IDweit
22. mesh_weit = mesh.Mesh.from_file(r'C:\Users\Caroline Kerscher\Desktop\color_distance_matlab\IDweit.txt')
23. axes.add_collection3d(mplot3d.art3d.Poly3DCollection(mesh_weit.vectors, facecolors = 'cyan'))
24.
25. #Beschriftung der Axen
26. axes.set_xlabel('x')
27. axes.set_ylabel('y')
28. axes.set_zlabel('z')
29.
30. #Zoomen auf Axe
31. axes.set_xlim(30, 42)
32. axes.set_ylim(10, 15)
33. axes.set_zlim(0, 10)
34. axes.view_init(elev = 30, azim = 70)
35.
36. # Auto scale to the mesh size
37. scale = your_mesh.points.flatten(-1)
38. scale = mesh_kurz.points.flatten(-1)
39. scale = mesh_weit.points.flatten(-1)
40. axes.auto_scale_xyz(scale, scale, scale)
41.
42. F1 = your_mesh.areas()
43. F1_ges = sum(F1)
44. print(F1_ges)
```

Literaturverzeichnis

- Anil, Engin Burak; Akinci, Burcu; Huber, Daniel: REPRESENTATION REQUIREMENTS OF AS-IS BUILDING INFORMATION MODELS GENERATED FROM LASER SCANNED POINT CLOUD DATA. Carnegie Mellon University, Pittsburgh. Department of Civil and Environmental Engineering. Online verfügbar unter <https://pdfs.semanticscholar.org/1045/d02e37922ee078aa1c796c606ab689064947.pdf>.
- Arbeitsproduktivität je Erwerbstätigenstunde. im Vergleich 1991-2018 (2018). Online verfügbar unter <https://blog.bulwiengesa.de/wohnen/wohnungsbau-am-rande-oekonomischer-vernunft>.
- Autodesk (Hg.) (2019): PHOTOGRAMMETRIE-SOFTWARE. Online verfügbar unter <https://www.autodesk.de/solutions/photogrammetry-software>, zuletzt geprüft am 14.11.2019.
- BAUMANN, Thomas; FREBER, Philipp-Stephan; SCHOBER, Kai-Stefan; KIRCHNER, Florian (2016): Bauwirtschaft im Wandel. Trends und Potentiale bis- 2020. Hg. v. Roland Berger und Hypo Vereinsbank. München. Online verfügbar unter [file:///C:/Users/Caroline%20Kerscher/AppData/Local/Packages/Microsoft.MicrosoftEdge_8wekyb3d8bbwe/TempState/Downloads/roland_berger_hvb_studie_bauwirtschaft_20160415_1_%20\(1\).pdf](file:///C:/Users/Caroline%20Kerscher/AppData/Local/Packages/Microsoft.MicrosoftEdge_8wekyb3d8bbwe/TempState/Downloads/roland_berger_hvb_studie_bauwirtschaft_20160415_1_%20(1).pdf).
- Baunetz Wissen (Hg.): IFC der offene Standard für BIM-Modelle. Online verfügbar unter <https://www.baunetzwissen.de/bim/fachwissen/standardisierung/ifc-der-offene-standard-fuer-bim-modelle-5288161>.
- Becker, Roland (2019): Word Embeddings – Methoden zur Repräsentation von Wörtern in Algorithmen und neuronalen Netzen. Bag-Of-Words (BOW). Hg. v. JAAI. Online verfügbar unter <https://jaai.de/word-embeddings-worteinbettung-word2vec-glove-bag-of-words-1872/>.
- Borrmann, André; König, Markus; Koch, Christian; Beetz, Jakob (2015): Building Information Modeling Technologische Grundlagen und industrielle Praxis. Wiesbaden: Springer Fachmedien. Online verfügbar unter <https://link.springer.com/book/10.1007%2F978-3-658-05606-3#about>.

- Bosche, Frederic (2011): Plane-based Registration of Construction Laser Scans with 3D/4D Building Models. Heriot-Watt University, Edinburgh. School of the Built Environment. Online verfügbar unter <http://web.sbe.hw.ac.uk/fbosche>.
- Bourke, Paul (Hg.) (2014): ptx - Laser scan plain data format. Online verfügbar unter <http://paulbourke.net/dataformats/pts/>.
- Braun, Alexander; Tuttas, S.; Borrmann, A.; Stilla, U. (2015): A concept for automated construction progress monitoring using BIM-based geometric constraints and photogrammetric point clouds. In: *Journal of Information Technology in Construction (ITcon)* 20/2015, S. 68–79. Online verfügbar unter <https://www.itcon.org/paper/2015/5>.
- Brownlee, Jason (2016): Parametric and Nonparametric Machine Learning Algorithms. Online verfügbar unter <https://machinelearningmastery.com/parametric-and-nonparametric-machine-learning-algorithms/>, zuletzt aktualisiert am 25.10.2019.
- Building Smart International (Hg.): Industry Foundation Classes (IFC) - An Introduction. What is IFC? Online verfügbar unter <https://technical.buildingsmart.org/standards/ifc/>, zuletzt geprüft am 24.12.2019.
- (2016): CLASSIFICATION OF PHOTOGRAMMETRIC POINT CLOUDS OF SCAFFOLDS FOR CONSTRUCTION SITE MONITORING USING SUBSPACE CLUSTERING AND PCA. XU, Yusheng; TUTTAS, Sebastian; HOEGNER, Ludwig; STILLA. Kongressbeitrag. Technische Universität München, München.
- Die deutsche Bauindustrie (Hg.) (2018): Bedeutung der Bauwirtschaft in Deutschland 2017. Unter Mitarbeit von Dipl.-Oec Heinrich Weitz. Online verfügbar unter https://www.bauindustrie.de/zahlen-fakten/bauwirtschaft-im-zahlenbild/bedeutung-der-bauwirtschaft_bwz/.
- Die Welt, Accenture (Hg.) (2014): Digitalisierungsgrad nach Branchen, 2014. Online verfügbar unter https://www.this-magazin.de/artikel/tis_Digitalisierung_im_Bausektor_3202823.html.
- Faltýnová, M.; Matoušková, E.; Šedina, J.; Pavelka, K. (Hg.) (2016): BUILDING FACADE DOCUMENTATION USING LASER SCANNING AND PHOTOGRAMMETRY AND DATA IMPLEMENTATION INTO BIM. XXIII ISPRS Congress.

- Prag, 12. - 16.07.2016. Department of Mapping and Cartography, Faculty of Civil Engineering (XLI-B3).
- File formats for BIM (2019). Online verfügbar unter https://www.designingbuildings.co.uk/wiki/File_formats_for_BIM.
- Fisher, Robert Bob (2002): The RANSAC (Random Sample Consensus) Algorithm. Hg. v. Robert Bob Fisher. Online verfügbar unter http://homepages.inf.ed.ac.uk/rbf/CVonline/LOCAL_COPIES/FISHER/RANSAC/.
- Folke, Santel (2001): Digitale Nahbereichsphotogrammetrie zur Erstellung von Oberflächenmodellen für Bodenerosionsversuche. Universität Hannover, Hannover. Institut für Photogrammetrie und Ingenieurvermessungen. Online verfügbar unter https://www.ipi.uni-hannover.de/fileadmin/institut/pdf/diplom_santel.pdf.
- Fromme, Dominic (2014): Grundlagen der Lasersicherheit. Hg. v. Allgemeine Unfallversicherungsanstalt. Auva, Seibersdorf Laboratories. Wien (M 080 - 01/2014). Online verfügbar unter <https://www.auva.at/cdscontent/load?contentid=10008.544574&version=1520426478>.
- Geospatial Modeling & Visualization (Hg.) (2018): File Formats – Exporting your data. Online verfügbar unter <http://gmv.cast.uark.edu/uncategorized/file-formats-exporting-your-data/>.
- GIM International (Hg.) (2018): Point Clouds: Photogrammetry or Lidar? GIM International. Online verfügbar unter <https://www.gim-international.com/content/news/point-clouds-photogrammetry-or-lidar>, zuletzt aktualisiert am 11.11.2019.
- Gordon, Bianca (2008): Zur Bestimmung von Messunsicherheiten terrestrischer Laserscanner. Dissertation. Technischen Universität Darmstadt, Darmstadt. Fachbereich Bauingenieurwesen und Geodäsie. Online verfügbar unter http://tuprints.ulb.tu-darmstadt.de/1206/1/Dissertation_BGordon.pdf.
- Hajian, Hamid; Becerik-Gerber, Burcin (Hg.) (2010): Scan to BIM: Factors Affecting Operational and Computational Errors and Productivity Loss. 27th International Symposium on Automation and Robotics in Construction, 20.06.2010. Online verfügbar unter https://www.researchgate.net/publication/320405691_Scan_to_BIM_Factors_Affecting_Operational_and_Computational_Errors_and_Productivity_Loss.

- Han, Kevin; Golparvr-Fard, Mani: Appearance-based Material Classification for Monitoring of Operation-Level Construction Progress Using 4D BIM and Site Photographs. University of Illinois at Urbana-Champaign, Illinois.
- Hauptverband der Bauindustrie (Hg.) (2019): Bedeutung der Bauwirtschaft: Schlüsselbranche für Deutschland. Anteile in Prozent 2018. Online verfügbar unter https://www.bauindustrie.de/zahlen-fakten/bauwirtschaft-im-zahlenbild/bedeutung-der-bauwirtschaft_bwz/.
- FROM POINT CLOUD TO BIM: A SURVEY OF EXISTING APPROACHES (2013). Unter Mitarbeit von N. Hichhri, Chiara Stefani, Livio de Luca, Philippe Veron und Gael Hamon. International CIPA Symposium. Strasbourg, 02. - 06.09.2013. Online verfügbar unter <http://sam.ensam.eu>.
- Huber, Daniel (2011): The ASTM E57 file format for 3D imaging data exchange. Carnegie Mellon University, Pittsburgh. The Robotics Institute.
- Hue, Value, Saturation (2019). Online verfügbar unter <http://learn.leighcotnoir.com/artsspeak/elements-color/hue-value-saturation/>.
- Johnson, Luke (2018): Decimating a Point Cloud in Recap for use in Revizto. Hg. v. What Revit Wants. Online verfügbar unter <https://www.is/decimating-a-point-cloud-in-recap-for-use-in-revizto/>.
- Kaarta (Hg.) (2020): Point Cloud Formats and What They Mean. Online verfügbar unter https://www.kaarta.com/scanning_tips/point-cloud-formats/.
- Katz, Dov (2018): Tag: structure from motion. Hg. v. Dov Katz: Computer Vision. Dubai. Online verfügbar unter <https://dovkatz.wordpress.com/tag/structure-from-motion/>, zuletzt geprüft am 14.11.2019.
- Kerscher, Caroline (2020): Python. Version 3.6.10.
- KOCIJAN, Matijas (2018): Digitalisierung im Bausektor. Bau-Boom gegen Kapazitäts-Engpässe. In: *THIS* 2018, 05.2018. Online verfügbar unter https://www.this-magazin.de/artikel/tis_Digitalisierung_im_Bausektor_3202823.html.
- LAS (LASer) File Format, Version 1.4. Online verfügbar unter <https://www.loc.gov/preservation/digital/formats/fdd/fdd000418.shtml>.

- Leica Geosystems AG (Hg.) (2019): Leica RTC360 LT – 3D-Reality-Capture-Lösung. Online verfügbar unter <https://leica-geosystems.com/de-de/products/laser-scanners/scanners/leica-rtc360>.
- Lliaifar, Amir (2019): Drone Dictionary — Essential Terms Every Pilot Should Know. Hg. v. Way Point. Online verfügbar unter <https://waypoint.sensefly.com/drone-dictionary-essential-terms-every-pilot-should-know/>, zuletzt geprüft am 23.05.2020.
- Luber, Stefan; Litzel, Nico (2018): Was ist ein Neuronales Netz? Hg. v. BIG Data Insider. Online verfügbar unter <https://www.bigdata-insider.de/was-ist-ein-neuronales-netz-a-686185/>.
- Macher, H el ene; Landes, Tania; Grussenmeyer, Pierre: From Point Clouds to Building Information Models: 3D Semi-Automatic Reconstruction of Indoors of Existing Buildings. Artikel, Strassburg. National Institute of Applied Sciences. Online verf ugbar unter https://www.google.de/url?sa=t&rct=j&q=&esrc=s&source=web&cd=10&ved=2ahUKEwi37PGtwMbkAhWQbVAKHdcvAGgQFjAJegQl-AxAC&url=https%3A%2F%2Fwww.mdpi.com%2F2076-3417%2F7%2F10%2F1030%2Fpdf&usq=AOvVaw2fldsTjfq6i0pV78OE_xvS.
- Malihi, S.; Valadan Zoej, M. J.; Hahn, M.; Mokhtarzade, M.; Arefi, H. (Hg.) (2016): 3D Building Reconstruction Using Dense Photogrammetric Point Cloud. XXIII ISPRS Congress. Prag, 12.-19.07.2016. The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences.
- Olsson, Calle: The Structure from Motion Pipeline. Video. Online verf ugbar unter <https://www.youtube.com/watch?v=i7ierVkXYa8>.
- Pa tra ucean, Viorica; Armeni, Iro; Nahangi, Mohammad; Yeung, Jamie; Brilakis, Ioannis; Haas, Carl (2015): State of research in automatic as-built modelling. University of Cambridge; University of Waterloo, Cambridge, Waterloo. Online verf ugbar unter <https://doi.org/10.1016/j.aei.2015.01.001>, zuletzt gepr uft am 01.01.2015.

- PERI AG (Hg.) (2018): Im Fokus: PERI bündelt BIM-Aktivitäten. Unter Mitarbeit von Martina Pankoke. Online verfügbar unter <https://www.peri.de/informationsportal-news-medien/veroeffentlichungen-presse/im-fokus-peri-buendelt-bim-aktivitaeten.html>.
- Renovation of Heritage Assets using BIM: a Case Study of the Durham Cathedral (2015). CIB W78 conference 2015. Eindhoven, 27.10.2015 - 29.10.2015. Online verfügbar unter <https://research.tees.ac.uk/en/publications/renovation-of-heritage-assets-using-bim-a-case-study-of-the-durha-3>.
- Revit (2020). Version 2021.1.1. Vereinigte Staaten: Autodesk. Online verfügbar unter <https://www.autodesk.de/products/revit/overview?plc=RVT&term=1-YEAR&support=ADVANCED&quantity=1>.
- Runceanu, L. S.; Haala, N. (Hg.) (2018): INDOOR MESH CLASSIFICATION FOR BIM. 3D Spatial Information Science – The Engine of Change. Delft, 01. - 05.10.2018 (XLII-4). Online verfügbar unter <https://doi.org/10.5194/isprs-archives-XLII-4-535-2018>, 2018.
- SHEN Z.; TANG P.; KANAAN O.; CHO Y. K. (2013): As-Built Error Modeling for Effective 3D Laser Scanning on Construction Sites. Arizona State University, Arizona. Webb School of Construction. Online verfügbar unter https://scholar.google.com/citations?user=wzUpk-sAAAAJ&hl=en#d=gs_md_cita-d&u=%2Fcitations%3Fview_op%3Dview_citation%26hl%3Den%26user%3DwzUpk-sAAAAJ%26citation_for_view%3DwzUpk-sAAAAJ%3A2osOgNQ5qMEC%26tzm%3D-120.
- Softtec (Hg.) (2018): PERI und SOFTTECH kooperieren: Gerüstbau-Software scaffmax bindet neues Fassadengerüst PERI UP Easy ein. Softtec. Online verfügbar unter <https://bausoftware.blogspot.com/2018/04/peri-softtech-geruestbau-scaffmax.html>.
- Tammik, Jeremy (2020): Autodesk Point Layout. Version 2020. Vereinigte Staaten: Autodesk. Online verfügbar unter <https://www.autodesk.com/products/point-layout/overview?plc=PNTLAY&term=1-YEAR&support=ADVANCED&quantity=1>.

- Thomson, Charles: Common 3D point cloud file formats & solving interoperability issues. Hg. v. Vercator. Online verfügbar unter <https://info.vercator.com/blog/what-are-the-most-common-3d-point-cloud-file-formats-and-how-to-solve-interoperability-issues>.
- Tran, H.; Khoshelham, K. (Hg.) (2019): BUILDING CHANGE DETECTION THROUGH COMPARISON OF A LIDAR SCAN WITH A BUILDING INFORMATION MODEL BUILDING CHANGE DETECTION THROUGH COMPARISON OF A LIDAR SCAN WITH A BUILDING INFORMATION MODEL. ISPRS Geospatial Week 2019. Enschede, 10–14 June 2019. The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences (XLII-2/W13, 889-893). Online verfügbar unter <https://doi.org/10.5194/isprs-archives-XLII-2-W13-889-2019>.
- Tuttas, S.; Braun, A.; Borrmann, A.; Stilla, U. (Hg.) (2014): Comparison of photogrammetric point clouds with BIM building elements for construction progress monitoring. ISPRS Technical Commission III Symposium. Zürich, 5 – 7 September 2014. The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences (XL-3).
- University of Cambridge: Klimawandel Was er für den Bausektor bedeutet Kernergebnisse aus dem Fünften Sachstandsbericht des IPCC Kernergebnisse 1 Im Jahr 2010 waren 32 Prozent des weltweiten Endenergieverbrauchs und 19 Prozent aller Treibhausgasemissionen auf Gebäude zurückzuführen. Gemäß Business as usual-Szenarien könnte sich die weltweit durch Gebäude verbrauchte Energie bis 2050 verdoppeln oder gar verdreifachen, unter anderem weil Milliarden Menschen Zugang zu angemessenem Wohnraum und zu Elektrizität erhalten werden. 2 Dagegen ließe sich der Energieverbrauch von Gebäuden bis 2050 stabilisieren oder sogar senken, wenn erprobte Erfolgsmethoden und -technologien in großem Maßstab eingesetzt würden. 3 Zahlreiche Hürden verhindern, dass vorhandene Möglichkeiten zum Energiesparen stärker genutzt werden. Dazu gehören beispielsweise unzureichende Markttransparenz, Risikoscheu und der begrenzte Zugang zu Kapital. Schon heute gibt es das Know-how für die Modernisierung von Bestandsgebäuden und den Neubau von Niedrigenergie- und Nullenergiegebäuden – was häufig mit nur marginalen Zusatzkosten verbunden ist. Außerdem steht eine breite Palette an wirksamen politischen

- Instrumenten zur Verfügung, mit denen sich die genannten Hürden beseitigen lassen. 4 Die sehr langen Investitionszyklen im Gebäudebereich bergen das Risiko, einen hohen Energieverbrauch langfristig gewissermaßen einzubetonieren ('lock-in' Effekte). 5 Die erwarteten Folgen des Klimawandels bergen ein großes Risiko für Gebäudeschäden. Klimawandel - Was er für die Baubranche bedeutet. Kernergebnisse aus dem Fünften Sachstandsbericht des IPCC. Online verfügbar unter <https://www.klimafakten.de/branchenbericht/was-der-klimawandel-fuer-den-bausektor-bedeutet>.
- Wang, Chao; Cho, Yong K.; Kim, Changwan (Hg.) (2015): Automatic BIM component extraction from point clouds of existing buildings for sustainability applications (56).
- Wang, Q.; Cheng, J.; Sohn, H.: Automated Quality Inspection of Precast Concrete Elements with Irregular Shapes Using Terrestrial Laser Scanner and BIM Technology. The Hong Kong University of Science and Technology, Hong Kong. Department of Civil and Environmental Engineering. Online verfügbar unter <https://pdfs.semanticscholar.org/f835/bd01c4830fd0342f29157b55843dd87fcc7e.pdf>.
- Wang, Qian (2019): Automatic checks for 3D point cloud data for safety regulation compliance for scaffolding work platforms. National University of Singapore, Singapore.
- Wikipedia-Autoren (2019): Photogrammetrie. Grundlagen. Hg. v. die freie Enzyklopädie Wikipedia (189571467). Online verfügbar unter <https://de.wikipedia.org/w/index.php?title=Photogrammetrie&oldid=189571467>, zuletzt aktualisiert am 15.06.2019.
- Wörgl: Farbenlehre. Farbmischung additiv – das RGB-Modell. Online verfügbar unter <http://www.brg-woergl.tsn.at/Grafik/>.
- Wunderlich, Thomas; Schäfer, Thomas; Reidl, Fabian (2013): Objektivierung von Spezifikationen Terrestrischer Laserscanner – Ein Beitrag des Geodätischen Prüflabors der Technischen Universität München. Technische Universität München, München. Lehrstuhl für Geodäsie. Online verfügbar unter file:///C:/Users/Caroline%20Kerscher/Documents/SS19/Bachelorarbeit/Literatur/2.2Laser/TUM_Geodäsie_Überblick_Laser.pdf.

-
- Xu, Y.; He, J.; Tuttas, S.; Stilla, U. (2015): RECONSTRUCTION OF SCAFFOLDING COMPONENTS FROM PHOTOGRAMMETRIC POINT CLOUDS OF A CONSTRUCTION SITE. In: *ISPRS Ann. Photogramm. Remote Sens. Spatial Inf. Sci.* II-3/W5, S. 401–408. DOI: 10.5194/isprsannals-II-3-W5-401-2015.
- Yiu, Tony (2019): Understanding Random Forest. How the Algorithm Works and Why it Is So Effective. Hg. v. Medium. Online verfügbar unter <https://towardsdatascience.com/understanding-random-forest-58381e0602d2>.
- Zhan, Quingming; Liang, Yubin; Xiao, Yinghui (Hg.) (2009): COLOR-BASED SEGMENTATION OF POINT CLOUDS. IAPRS. Paris, 1.-2. September 2009.

Erklärung

Hiermit erkläre ich, dass ich die vorliegende Bachelor-Thesis selbstständig angefertigt habe. Es wurden nur die in der Arbeit ausdrücklich benannten Quellen und Hilfsmittel benutzt. Wörtlich oder sinngemäß übernommenes Gedankengut habe ich als solches kenntlich gemacht.

Ich versichere außerdem, dass die vorliegende Arbeit noch nicht einem anderen Prüfungsverfahren zugrunde gelegen hat.

München, 15. November 2020



Vorname Nachname

Caroline Kerscher



Kerscher.cs@gmail.com