

Geometrische Modellierung von Balkenstrukturen am Beispiel des Bautzner Dachstuhls

Torsten FRENZEL, Lorent MUCHA und Babett HÜBSCH

Kurzzusammenfassung

In dieser Arbeit wird die geometrische Modellierung einer komplexen Balkenkonstruktion beschrieben. Der dabei gewählte Ansatz zielt auf eine möglichst automatische und generische Bearbeitung des Problems mittels der Point-Cloud-Bibliothek (PCL) ab. Als Ergebnis berechnet der Algorithmus aus der Punktwolke ein geometrisches Datenmodell, welches zur weiteren Auswertung genutzt werden kann.

1 Einleitung

Eine wichtige Teilaufgabe in der automatischen Bildverarbeitung ist die Erkennung und Modellierung geometrischer Strukturen, wie sie an und in Bauwerken vorkommen. Anhand der gewonnen Geometrie lassen sich später weitere baustatische Informationen, wie z.B.: der Schwerpunkt, gewinnen, oder die Daten können in dynamischen Simulationen genutzt werden.

Die Gliederung der Arbeit ist wie folgt: Nach der Beschreibung der Ausgangsdaten und des Aufnahmeobjektes, folgt die Darstellung des Workflows, welcher grob in 3 Schritte unterteilt werden kann: Datenaufbereitung, Zerlegung in Balken und geometrische Rekonstruktion. Nach der Darstellung und Bewertung der Ergebnisse erfolgt eine Beschreibung des Projektmanagements und ein abschließendes Fazit

2 Ausgangsdatensatz und Aufnahmeobjekt

Der zur Verfügung gestellte Datensatz von Laserpunktwolken zeigt den Dachstuhl des Bautzener Doms (Abb. 1). Die Aufnahme erfolgte im Rahmen einer Diplomarbeit von Frau A. Bienert, TUD (2004) mit einem *Riegel LMS-Z420i* Laserscanners. Die Punktwolke besteht aus 12 Aufnahmestandpunkten und ca. 34 Millionen Punkten. Die einzelnen Aufnahmen waren bereits Registriert und die Genauigkeit der einzelnen Standpunkte zueinander beträgt 5 -7 mm. Für die Segmentierung und Modellierung eines Stützbalkens diente die gesamte Punktwolke als Datengrundlage (SCHELLER & SCHNEIDER 2006).



Abb. 1: Laserscanneraufnahme eines Standpunktes im Dachstuhl Bautzener Dom als zweidimensionales Intensitätsbild (Quelle: SCHELLER & SCHNEIDER 2006)

3 Workflow

Der gesamte Workflow gliedert sich in 3 Schritte: Zuerst wird die Punktwolke aufbereitet und in ein geeignetes Datenformat zu Weiterverarbeitung exportiert. Dann wird die Punktwolke verschnitten und unwichtige sowie störende Daten (Boden, Dach, Wände) werden entfernt. Hierbei erfolgt auch eine Zerlegung in einzelne Elemente, den Balken. Für diese Balken wird in einem dritten Schritt die Geometrie extrahiert und modelliert.

3.1 Datenaufbereitung

Mittels der scannereigenen Software *RISCAN PRO*¹ erfolgte ausschließlich der Export der Messdaten, da die Registrierung der Aufnahmepositionen bereits vorhanden war. Als Exportformat wurde eine Textdatei gewählt, in der jede Zeile ein Koordinatentripel enthält. Für die Nutzung von *PCL* für das erstellte C++ Programm erfolgte für jede Aufnahmeposition eine Anpassung der Header-Information in ein *PCL*-Konformes Textformat *PCD* (POINTCLOUD.ORG 2014). Diese Konvertierung wird für die Unterstützung einiger Erweiterungen der Punktwolkenverarbeitung in *PCL* benötigt.

Der Algorithmus führt die Punktwolken der einzelnen Scanneraufnahmen zu einer Punktwolke als Merge zusammen. Um die Koordinaten aufnehmen zu können, wird eine leere Punktwolke erzeugt, in der die 3D-Punkte hineingeschrieben werden. Der Arbeitsablauf ist dabei als Schleife konzipiert und öffnet die zuvor erstellten *PCD*-Dateien. Die Punkte des jeweiligen Scans werden mit *loadPCDFile* eingelesen und mit dem Additions-Operator an die zuvor erstellte leere Punktwolke angefügt. Nach dem Durchlauf aller Scans wird das Ergebnis des Merges in die Ausgabedatei geschrieben. Dieser Teil erfolgt vollautomatisiert.

Nach Zusammenführen aller Scans in eine Punktwolke wird ein Downsampling durchgeführt, um die Anzahl der zu verarbeitenden Datenmenge zu verringern. Dazu wird der VoxelGrid-Filter genutzt (POINTCLOUDS.ORG 2014,1). Neben dem Input ist die Definition der Blattgröße *setLeafSize* erforderlich. Hier ist die Parameterwahl abhängig von der gegebenen Punktwolke. Im Algorithmus wird keine Abfrage an den Operateur gestellt, sondern führt ein Downsampling für drei verschiedene Blattgrößen von 5 cm, 10 cm und 20 cm durch. Die entsprechenden Blattgrößen werden dabei in den Dateinamen codiert, um die verschiedenen Ergebnisse für den weiteren Ablauf nutzen zu können. Die Wahl der Blattgröße

¹ Riscan Pro Software: <http://www.riegl.com/products/software-packages/riscan-pro/>

bestimmt, wie stark die Punktmenge reduziert wird und damit die Lauf- und Rechenzeit sowie Genauigkeit der nachfolgenden Algorithmen. Eine große Blattgröße führt zu einer starken Reduktion der Punktmenge und Reduzierung der Rechenzeit. Für ein exaktes Ergebnis sollte die kleinste Blattgröße, bzw. nicht reduzierte Punktwolke verwendet werden.

3.2 Segmentierung

3.2.1 Entfernung des Bodens und des Dachs

Um im zweiten Schritt eine Extraktion der einzelnen Balken durchzuführen, mussten zu Beginn das Dach und der Boden entfernt werden. Für den Boden hat sich der morphologische Filter als geeignetsten Algorithmus herausgestellt. Dieser Filter trennt Boden- (Gelände-)Punkte von den "Nicht-Boden"-Punkten, indem ein Strukturelement erstellt wird, welches die Geländehöhenunterschiede in Abhängigkeit von der Entfernung beschreibt. Je größer die horizontale Entfernung, desto größer kann der Höhenunterschied zwischen den Bodenpunkten sein.

Für die Filterung wird dieses Strukturelement vertikal beginnend unterhalb der Punktmenge nach oben verschoben. Dazu wird es planimetrisch auf jedem Punkt zentriert und nach oben geschoben, bis es an einen Punkt innerhalb des Umkreises angepasst wird und die Verschiebung gestoppt wird. Der untersuchte Punkt ist kein Geländepunkt, wenn er nicht das Zentrum ist, auf den das Strukturelement zentriert wurde. Auf diese Weise wird die gesamte Punktwolke klassifiziert (SITHOLE & VOSSELMAN, 2003). Der Algorithmus ist in der PCL Bibliothek als `ProgressiveMorphologicalFilter` (POINTCLOUDS.ORG, 2017) implementiert.

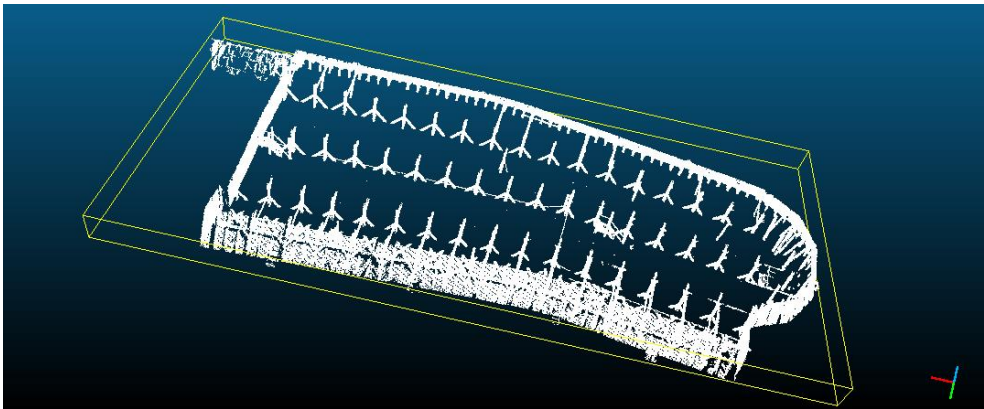


Abb. 2: Balkenwerk nach Filterung von Decke und Boden.

Da der Algorithmus sehr rechenintensiv ist, kam als Input nur die downgesampelte Punktwolke („merged.pcd“) mit der Blattgröße von 5cm aus dem vorangegangenen Segmentierungsschritt, in Frage. Die Anwendung auf die originale Punktwolke wurde nach drei Tagen Rechenzeit abgebrochen. Um jedoch eine größtmögliche Punktdichte zu erreichen, wurde als Workaround ein Merge mit der Originalen Punktwolke durchgeführt (Kapitel 3.2.3).

Um aus der eingelesenen Punktwolke das Dach zu entfernen, wurde ein PCL PassThrough-Filter (POINTCLOUDS.ORG, 2013) verwendet, welcher Punkte entfernt die nicht in einem festgelegten Bereich liegen. Die Parameter des PassThrough-Filters konnten mit Hilfe des Programms CloudCompare² bestimmt werden. Der gefilterte Bereich wurde für die z-Koordinate von 0 m und 3 m gesetzt. Abbildung 2 dokumentiert das Ergebnis der Filterung von Decke und Boden.

3.2.2 Clusterung und Segmentierung

Nach dem Filterprozess erfolgte die Extraktion der Balken durch eine Clusterung und Segmentierung. Um die Clusterung effizient zu gestalten und mögliche Fehlerbehaftete Ergebnisse zu vermeiden, wurde der Clusterungsprozess auf eine vorher gefilterte Balkenstümpfe-Punktwolke angewandt. Hierfür wurde erneut ein PassThrough Filter verwendet.

Als Ergebnis wurden alle irrelevanten und störenden Elemente bis auf die Seitenwände entfernt, was mit der Parametrisierung der z-Achse zwischen 1.8 m und 2 m erreicht werden konnte. Abbildung 3 stellt den Output des PassThrough Filters dar. Das Ergebnis wurde als Neue .ply Datei unter dem Namen „balkenstuempfe.ply“ gespeichert und dient als Input für die Clusterung.

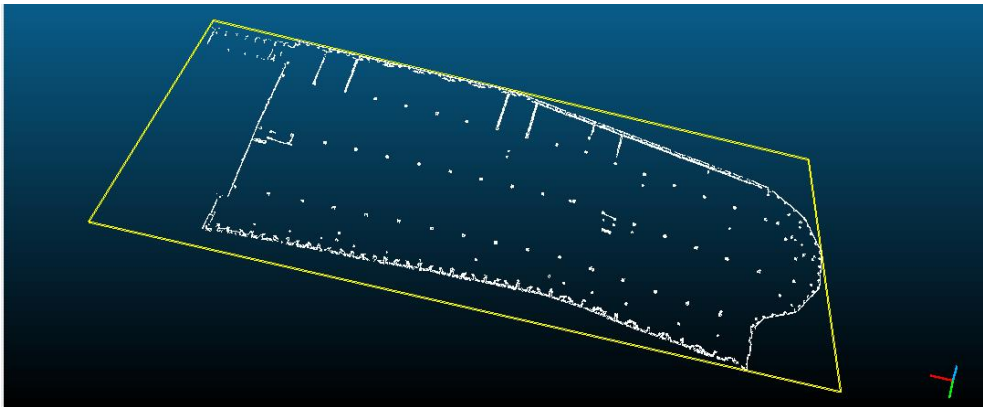


Abb. 3: Gefilterte Balkenstümpfe für das EuclideanClustering

Für die Clusterung der Balken wurde das Euklidische Clusterverfahren verwendet, welches ebenfalls in der PCL zur Verfügung gestellt wird. Die geclusterten Balkenstümpfe, also der Output der PCL *EuclideanClusterExtraction* (POINTCLOUD.ORG, 2003), wurden in einer Punktwolke gespeichert. Um eine Segmentierung der Balken anhand der Clusterungsergebnisse durchzuführen, ist es essentiell den Mittelpunkt zu ermitteln, an welchen im darauffolgenden Schritt der *MinCut* (POINTCLOUD.ORG, 2014,2) Segmentierungsalgorithmus angewandt wird. Aus diesem Grund wurde für jedes Cluster ein Centroid eingepasst, welcher den jeweiligen Mittelpunkt des Clusters darstellt. Die Menge aller Centroid-Punkte, wurde in einer weiteren Punktwolke gespeichert. Abbildung 4 zeigt diese Punktwolke mit den Mittelpunkten der Cluster.

² <http://www.danielgm.net/cc/>

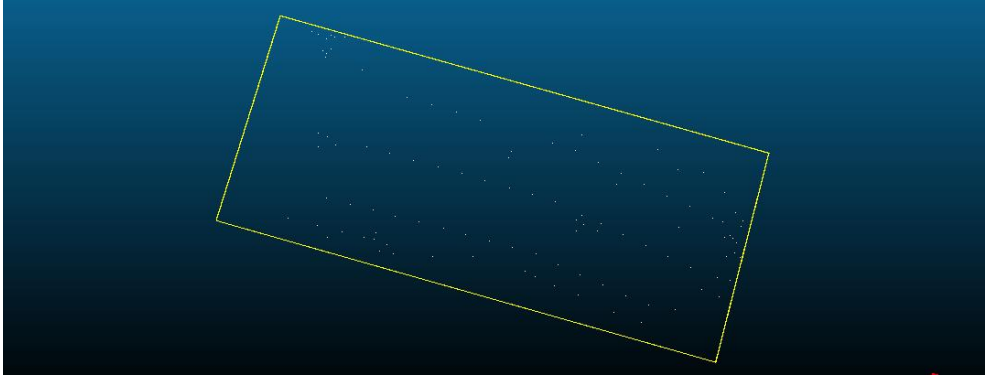


Abb. 4: Punktwolke der Centroide, welche anhand der Cluster gebildet wurden.

Diese Punktwolke der Centroide enthielt jedoch auch Punkte für die Seitenwände, welche nicht das Balkenwerk widerspiegeln, das extrahiert werden sollte. Deshalb wurde erneut der *PassThrough* -Filter zur Entfernung dieser Punkte angewandt. Der gefilterte Bereich für die y-Koordinate lag zwischen -7 m und 14 m und wurde mit Hilfe von CloudCompare ermittelt.

Als nächstes erfolgt die Segmentierung der Balken, aus der downgesampten Gesamt-Punktwolke mittels der *MinCut*-Segmentierung. Dieser Algorithmus führt eine binäre Zerlegung der Eingangsdaten anhand eines Mittelpunktes und eines Radius, durch. In diesem Fall in Vorder- und Hintergrundpunkte (Punkte, die zum Objekt gehören und welche, die nicht dazu gehören).

Für die Segmentierung wurde die Centroid-Punktwolke iteriert und jeder iterierte Punkt als Mittelpunkt gesetzt, der Radius betrug 1,5m. Jeder so gefilterte Balken wurde in eine separate Ausgabedatei gespeichert (balken_downsampelt*.ply). Als Erweiterung der Iteration diente ein Array, welches die Anzahl der extrahierten Balken speicherte.

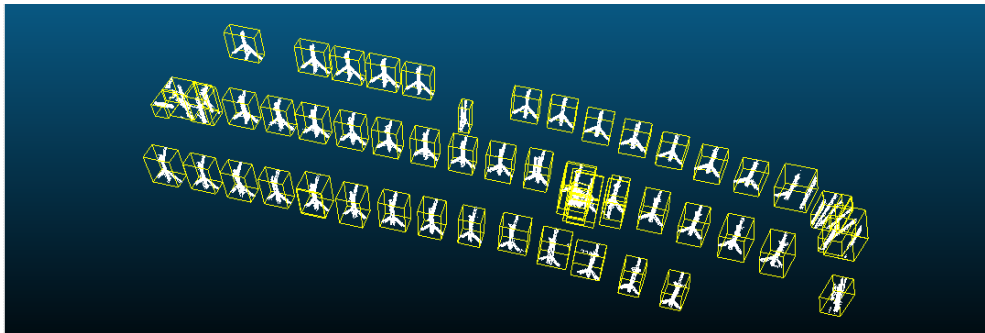


Abb. 5: Ergebnis der MinCut-Segmentierung mit den einzelnen Balken

3.2.3 Zusammenführen der Gedownsampelten Balken mit der Originalen Punktwolke

Die extrahierten Balken wiesen nur eine sehr geringe Punktdichte auf, was auf die Verwendung der downgesampelten Punktwolke zurückzuführen ist. Für ein besseres Ergebnis muss die originale Punktwolke einbezogen werden, welche die höchste Punktdichte besitzt. Für diesen Schritt wurde über alle extrahierten Balken iteriert und die Bounding-Box berechnet. Darauffolgend wurde geprüft, welche Punkte aus der originalen Punktwolke innerhalb der Bounding-Box lagen. Alle Punkte innerhalb der Bounding-Box, welche zur originalen Punktwolke gehörten, wurden in eine neue Punktwolke kopiert und in einer Ausgabedatei gespeichert („balken_original*.ply“). Die Anzahl der Punkte je Balken konnte dadurch von durchschnittlich 700 auf 50.000 erhöht werden.

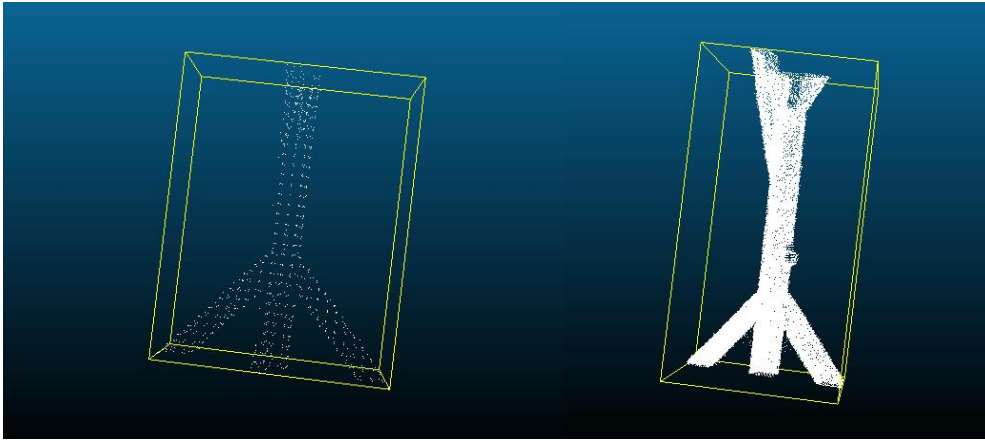


Abb. 6: Vergleich der Balkenextraktion aus downgesampelten und originalen Punktwolke, mit ihren Punktdichten

3.2.4 Ergebnis der Segmentierung

Mit der Segmentierung und Zusammenführung der Originalen Punktwolke wurden 56 Balken automatisch extrahiert. Die Originale Punktwolke besitzt jedoch nur 51 Balken, womit 5 Balken falsch extrahiert wurden. In Abbildung 7 sind die Problemfälle dokumentiert. Durch eine fehlerhafte Clusterung wurden einige Balken mehrfach segmentiert (Fall 2 und 3), hier liegt das Problem an zwei Leitern, welche als Balken geclustert wurden. In den Fällen 4 und 5 wurden keine Balken erkannt, weil diese innerhalb der *MinCut* Segmentierung außerhalb der Schwellenwerte lagen und somit nicht segmentiert wurden. Bei den verschiedenen Fällen in Punkt 1 handelt es sich um Wände, welche den Balken vorgelagert sind. Diese konnten durch den *PassThrough* Filter und der Setzung von Schwellenwerten bei der Euklidischen Clusterung nicht entfernt werden, da ihre Punktmenge ähnlich der Balken war.

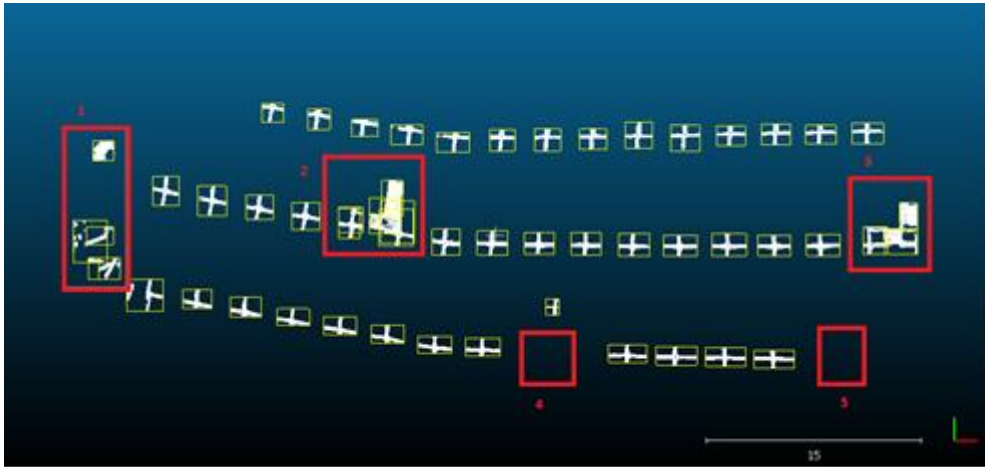


Abb. 7: Falsch extrahierte Balken mit den Fällen 1 bis 5

3.3 Modellierung der Balken

Die geometrische Modellierung der Balken erfolgt in mehreren Phasen, die nacheinander sequentiell ausgeführt werden. Zur Modellierung erfolgt zuerst eine weitere Zerlegung der Punktwolke einer Balkenkonstruktion in Cluster und danach eine sukzessive Extraktion der Extraktion der von Quader-Modellen mit 8 Eckpunkten in unterschiedlichen Ansätzen.

3.3.1 Vorbereitende Phase

Zuerst wird die Punktwolke in 32 horizontale Scheiben entlang der z-Koordinate in Slices zerschnitten. Für jeden Slice erfolgt ein Clustering mittels euklidischer Distanzclustering. Dadurch zerfällt der Slice in separate Cluster, für welche die Bounding-Box bestimmt wird. In benachbarten Slices erfolgt die Bestimmung sich überlappender Cluster anhand der xy-Koordinaten der Bounding-Boxen. Ist die Größe der Bounding-Boxen sehr ähnlich, so werden die beiden Cluster einem gemeinsamen Balkenabschnitt zugeordnet. Durch die Transitivität von Überlappungen entstehen größere Balkenschnitte aus einer größeren Anzahl von Clustern

3.3.2 Phase 1: Extraktion der Seiten eines Balkenabschnittes mittels RANSAC

In der ersten Phase wird ein modifizierter RANSAC Ansatz gewählt, der weiteres Ad-hoc Wissen nutzt um die 4 Seiten des Modells zu extrahieren. Ziel der zusätzlichen Bedingungen ist es, sicher zu stellen, dass die extrahierten Ebenen den Seiten des Balkens entsprechen und nicht quer durch die Punktwolke schneiden.

Als Erstes werden zufällig 3 Punkte ausgewählt und darauffolgend Zwangsbedingungen geprüft. Haben diese Punkte einen definierten Mindestabstand zueinander, wird die Anzahl der Inlier und linken bzw. rechten Ausreißer berechnet. Die Anzahl der linken oder rechten Ausreißer muss, dabei einen Schwellwert unterschreiten. Des Weiteren wird die minimale

Distanz aller Inlier zum ihrem Mittelwert geprüft, die ebenfalls einen Schwellwert nicht unterschreiten darf.

Für zwei extrahierte Ebenen wird der Schnittwinkel auf Rechtwinkligkeit geprüft. Falls der Winkel zu klein ist, wird die zweite Ebene verworfen und eine neue Ziehung beginnt.

Wenn mittels RANSAC erfolgreich zwei rechtwinklig zueinanderstehende Ebenen detektiert wurden, erfolgt die Rekonstruktion der 4 senkrechten Kanten der Quader-Modelle. Dazu wird als erstes die Schnittgerade der beiden Ebenen errechnet. Der Start- und Endpunkt dieser Gerade ergibt sich durch zwei weitere Schnitte mit den oberen und unteren horizontalen Begrenzungsflächen der Bounding-Box. Die fehlenden 3 Kanten des Quader-Modells ergeben sich durch das Verschieben der Kante entlang der beiden Flächennormalen bis zum entferntesten Rand der Bounding-Box. Für die diagonal gegenüberliegende Kante muss entsprechend eine zweifache Verschiebung durchgeführt werden. Dabei ist zu Beachten, dass die Verschiebung nur entlang der x und y-Koordinate durchgeführt wird und die z-Koordinate unverändert bleibt. Dabei darf das durch das Quader-Modell aufgespannte Volumen, das Volumen der Bounding-Box nicht um die Hälfte unterschreiten.

Als Ergebnis erhält man die 8 Punkte quaderförmigen Modells, welches die Begrenzung des durch den Cluster verlaufenden Balken approximiert.

3.3.3 Phase 2: Erweiterung eines Balkenabschnittes

Als Ergebnis der vorhergehenden Phase gibt es vereinzelte Modelle in erfolgreich modellierten Clustern. Dabei sind innerhalb eines Balkenabschnittes noch Cluster vorhanden die noch nicht modelliert sind. Deshalb wird in einem weiteren Schritt eine näherungsweise Rekonstruktion dieser Cluster durchgeführt, indem die Quader-Modelle kopiert und entlang der Hauptrichtung des Balkens verschoben werden.

3.3.4 Phase 3: Optimierung der Einpassung der Modelle

In einem Zwischenschritt erfolgt die Verbesserung der bisher generierten Quader-Modelle. Dazu erfolgt eine Zerlegung der Punktwolke des Clusters entsprechend des minimalen Abstandes zu den modellierten 4 Seitenflächen des Quaders. Mittels Hauptkomponentenanalyse werden aus der separierten Punktwolken die 3 Hauptrichtungen extrahiert. Die dritte Hauptrichtung bildet die neue Flächennormale der Seite und der Durchschnitt der Punktkoordinaten definiert den Fußpunkt. Die neuen Eckpunkte des Quaders durch Verschneidung der 4 Seitenflächen und den oberen und unteren horizontalen Begrenzungsflächen bestimmt.

3.3.5 Phase 4: Rekonstruktion unmodellierter Balkenabschnitte

Nach der Phase 2 gibt es Balkenabschnitte die bereits vollständig modelliert sind und Balkenabschnitte die noch unmodelliert sind, da in Phase 1 kein Cluster modelliert werden konnte. Im letzteren Fall, wird ein mittlerer Richtungsvektor über alle Cluster im Balkenabschnitt berechnet. Dazu werden die Differenzen der Punkte der Bounding-Boxen der benachbarten Cluster berechnet und gemittelt. Als Ergebnis ergibt sich ein mittlerer Richtungsvektor, dessen Richtung näherungsweise der Hauptrichtung des Balkens entspricht. Danach wird in jedem Cluster ein Quader-Modell eingepasst, dessen vertikale Kanten in Richtung des mittleren Richtungsvektors verlaufen und dessen horizontale Kanten entlang

der x- und y-Achse verlaufen. Durch Verschneidung mit den Begrenzungsebenen der Bounding-Box lassen sich wiederum die 8 Eckpunkte des Modells extrahieren.

3.3.6 Zusammenfügung einzelner Balkenabschnitte

Nach der Phase 3 sind alle vorhandenen Balkenabschnitte vollständig modelliert. In der nächsten Phase werden Balkenabschnitte, die zu einem einzelnen Balken gehören zusammengefügt. Dazu wird ein mittlerer Verschiebevektor über die einzelnen Quader-Modelle eines Balkenabschnittes berechnet. Ein einzelner Verschiebevektor ergibt sich aus der Differenz von zwei gleichen Eckpunkte von benachbarten Modellen. Die mittlere Verschiebevektor ergibt sich aus dem Durchschnitt aller einzelnen so ermittelten Verschiebevektoren. Haben zwei Balkenschnitte sehr ähnlich Verschiebevektoren und gibt es eine lückenlose Kette von überlappenden Cluster zwischen diesen Balkenabschnitten, so werden beide Balkenschnitte zu einem neuen Balkenabschnitt zusammengefügt, wobei die dazwischensliegenden Cluster integriert und durch das Kopieren der bereits existierenden Quader-Modelle modelliert werden.

3.3.7 Phase 5: Verschneidung von Balken

In den Clustern in denen Balken zusammenlaufen, lassen sich die Seitenflächen nicht korrekt detektieren. Deshalb wird wiederum ein Ansatz verfolgt, der nur mit den abgeleiteten geometrischen Informationen arbeitet und dabei so wenig wie möglich ad-hoc Wissen nutzt. Dazu wird an einem Ende des Balkenabschnittes ein überlappendes Cluster ermittelt, zu dem es bereits ein Referenz-Quader-Modell in einem anderen Balkenabschnitt gibt. Mit diesem Referenz-Quader-Modell wird das kopierte Ziel-Quader-Modell verschnitten. Für die Verschneidung wird im Referenz-Quader-Modell zuerst heuristisch die Seite ermittelt, welche unmittelbar an den Zielquader angrenzt. Dies erfolgt anhand der Ausrichtung und der Messung des Abstandes zum Ziel-Quader-Modell. Maßgebend für das Ergebnis der Verschneidung sind die 4 Seiten des Quaders, die in vertikaler Richtung verlaufen. Dabei können prinzipiell 5 Fälle für die Lage der 4 Schnittpunkte unterschieden werden:

1. alle 4 Schnittpunkte liegen oberhalb der oberen horizontalen Begrenzungsfläche: Der Ziel-Quader bleibt ungeschnitten
2. zwei Schnittpunkte liegen oberhalb und 2 Schnittpunkt liegen zwischen oberer und unterer horizontaler Begrenzungsfläche: Der Ziel-Quader ist nur teilweise angeschnitten. Zur Modellierung werden kleinere zwei Quader generiert. Der erste Quader ist ungeschnitten und der zweite Quader ist vollständig geschnitten
3. zwei Schnittpunkte liegen oberhalb der oberen und 2 Schnittpunkte liegen unterhalb der unteren Begrenzungsfläche: Der Ziel-Quader ist vollständig angeschnitten
4. zwei liegen zwischen maximaler und minimaler z-Koordinate und 2 Schnittpunkte liegen unterhalb der minimalen z-Koordinate: Der Quader fast vollständig abgeschnitten, es bleibt nur noch ein Stumpf
5. alle 4 Schnittpunkte liegen unterhalb der unteren horizontalen Begrenzungsfläche. Der Ziel-Quader ist vollständig abgeschnitten, d.h. das Ende der Balkenverbindung ist erreicht

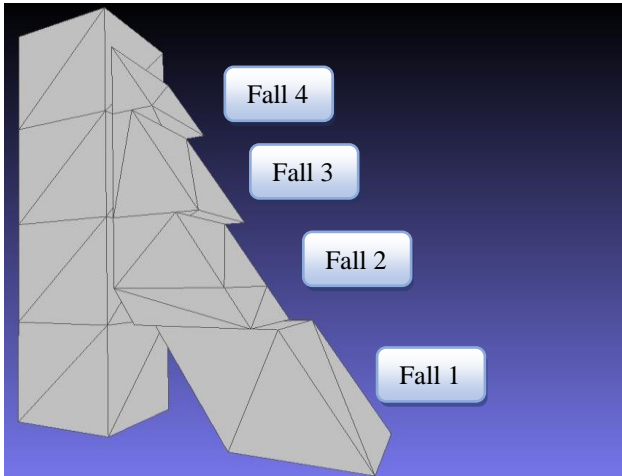


Abb. 8: Darstellung der Verschneidung zweier Balken

Je nach Fall müssen weitere Schnittpunkte horizontaler Seiten mit der Referenzfläche berechnet werden, um die Eckpunkte für den Ziel-Quader auszurechnen. Zusätzlich gibt es stets 4 weitere Unterfälle zu unterscheiden, die sich aus den 4 unterschiedlichen Richtungen ergeben, aus denen der Ziel-Quader mit dem Referenz-Quader zusammenlaufen kann. Außerdem ist zu unterscheiden, ob das Zusammenlaufen von unten nach oben oder von oben nach unten erfolgt. Der zweite Fall lässt sich durch geeignetes Vertauschen der z-Koordinaten der Eckpunkte auf den ersten Fall abbilden.

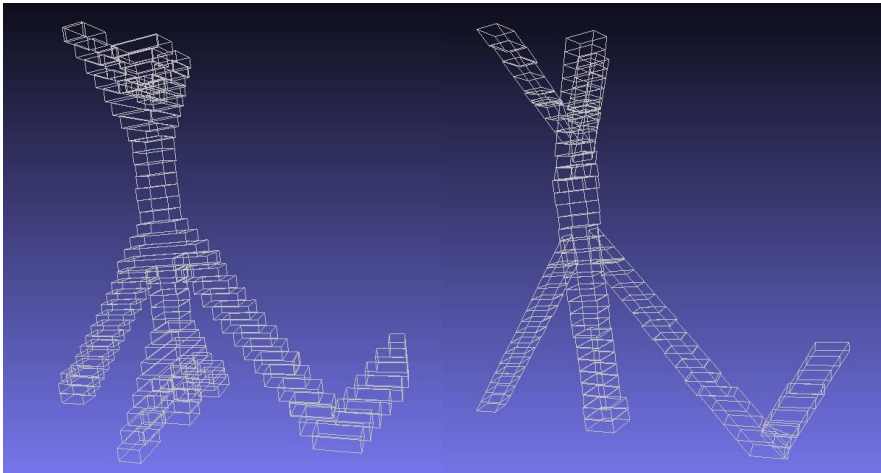


Abb. 9: Gegenüberstellung der Balkenstruktur mit umfassender Bounding-Box (links) und dem Endergebnis der geometrischen Rekonstruktion (rechts).

In Abbildung 9 ist links die Zerlegung einer Balkenstruktur in Cluster mit den umfassenden Bounding-Boxen dargestellt. Dabei erfolgt die vertikale Zerlegung in 32 Slices. Rechts ist

das entsprechende Endergebnis nach der Rekonstruktion der Geometrie zu sehen, welches als PLY-Datei mit den Kanteninformationen exportiert ist.

4 Ergebnisse

Allgemein lässt sich für die Balkenmodellierung feststellen, dass 10 der 55 segmentierten Balken erfolgreich modelliert wurden. Die fehlerhafte Modellierung lässt auf folgende Probleme zurückführen:

1. Auswahl falscher Schnittkanten bei der Verschneidung
2. Störende Elemente (Querstreben, Seile, Leitern)
3. Falsche Flächeneinpassung der Quadermodelle
4. Ungünstiges Clustering
5. Zu kurze Balkenabschnitte

Die detaillierte Auswertung der Balkenmodellierung wird exemplarisch für 5 Balken durchgeführt, die erfolgreich modelliert werden konnten. Wobei die ersten 2 Balken zum Test des Algorithmus benutzt wurden, während die anderen 3 Balken eine Teilmenge von 55 Balken aus der automatischen Segmentierung sind.

Für die einzelnen Phasen der Modellierung lässt sich bestimmen, wie viele Quader-Modelle jeweils extrahiert werden konnten, wie in Tabelle 1 dargestellt.

Tabelle 1: Anzahl der rekonstruierten Quader-Modelle in den einzelnen Phasen der Modellierung

	Phase 1	Phase 2	Phase 4	Phase 5	Phase 6	Cluster
Balken 1	2	5	39	51	92	68
Balken 2	4	6	37	47	97	74
Balken 10	0	0	16	22	52	37
Balken 22	4	10	15	15	42	37
Balken 44	3	10	17	22	52	37

Dabei ist zu erkennen, dass in der Phase 1 mittels RANSAC nur sehr wenig Boxen erfolgreich modelliert werden können. Daraus lässt sich zum einen folgern, dass die Daten nicht sehr genau sind, als auch, dass zu wenig Daten vorhanden sind, um die Flächen korrekt zu detektieren. In der Phase zwei erfolgt ein Kopieren der bisher extrahierten Modelle auf die einzelnen Balkenabschnitte. In der Phase 4 werden bisher nicht modellierte Balkenabschnitte näherungsweise anhand der Hauptrichtung modelliert. Dadurch erhöht sich die Anzahl der gewonnenen Quader signifikant. In der Phase 5 schließlich werden die Balkenabschnitte erweitert und zusammengefasst. In der letzten Phase bei der Verschneidung von Balken erhöht sich die Zahl der Quader-Modelle, wobei zu beachten ist, dass einem Cluster mehrere Modelle zugeordnet werden können und somit die Zahl der Modelle größer als die Zahl der Cluster werden kann.

Für die durchgeführte Extraktion der Geometrie lassen sich auch Genauigkeitsbetrachtungen durchführen. Dadurch lässt sich abschätzen, wie gut die modellierten Flächen sich in die Daten einpassen, und man erhält damit ein quantitatives Maß für die Qualität der Modellierung.

Dazu wird für jedes Modell die zugehörige Cluster-Punktwolke betrachtet. Für jeden Punkt der Punktwolke wird die Distanz zu allen vier senkrechten Flächen des Quaders bestimmt und davon wiederum die minimale Distanz ausgewählt. Der Durchschnittswert der minimalen Distanzen aller Punkte bildet den Fehlerwert für das Modell. Diese Berechnung wird für alle modellierten Cluster durchgeführt, um den Gesamtfehlerwert zu erhalten.

Tabelle 2: Genauigkeiten am Beispiel zweier Balken

	Mittlerer Abstand vor Einpassung	Mittlerer Abstand nach Einpassung
Balken 1	1,87 mm	0,926 mm
Balken 2	1,69 mm	0,691 mm
Balken 10	1,64 mm	0,732 mm
Balken 27	1,82 mm	0,651 mm
Balken 44	1,69 mm	0,69 mm

Tabelle 2 zeigt die mittleren Abstände der fünf modellierten Balken vor und nach der Einpassung der Ebenen in Phase 3. Dabei lässt sich erkennen, dass die Fehlerwerte sämtlich im Millimeterbereich liegen. Da die Genauigkeit der Ausgangsdaten aufgrund fehlender Referenzmessungen nicht bekannt ist, lässt sich nicht bestimmen, ob der Fehler nur modellbedingt verursacht ist oder auch signifikante Messungenauigkeiten beinhaltet.

5 Projektmanagement

Aufgrund der unbekannten Datenqualität und der Größe der Daten war der Aufwand für die Bearbeitung schwer abzuschätzen. Durch die unterschiedlichen technischen Kenntnisse der Projektteilnehmer hat sich eine Zerlegung der Aufgaben in drei relativ unabhängige Pakete als zweckmäßig erwiesen. Dadurch ergibt sich eine unabhängige Bearbeitung der Algorithmen. Zum besseren Austausch der Ergebnisse wurde das Projekt in ein Repository in GitHub hinterlegt, Änderungen konnten somit leichter verfolgt und Probleme jederzeit besprochen werden.

Als Problem ergibt sich aber daraus, dass sich nur schwer Milestones mit Zwischenergebnissen definieren ließen. Dies hat sich insbesondere darin gezeigt, dass zur Rekonstruktion nur bedingt Standardalgorithmen verwendet werden konnten.

6 Fazit (und ggf. Ausblick)

Der Ansatz der Arbeit verfolgte eine vollständig automatisierte Extraktion der Geometrie einer komplexen Balkenkonstruktion. Dabei zeigten sich aufgrund der Datenmenge und der Datenqualität verschiedene Probleme.

Eine erfolgreiche Extraktion der Seitenflächen durch robuste Schätzer ist auf der Grundlage der Daten nur eingeschränkt möglich, da die Kanten nur sehr schwach erkennbar sind. Dies führt dazu, dass die Flächen sehr oft nicht den tatsächlichen Seitenflächen der Balken entsprechen. Deshalb mussten für die Rekonstruktion der Geometrie neue Ansätze entwickelt werden.

Prinzipiell ist wenig zusätzliches Wissen nötig, um eine erfolgreiche Geometrie zu extrahieren. Allerdings gibt es in den verwendeten und implementierten Algorithmen viele Parameter, die angepasst werden müssen. Dies beginnt mit der Zerlegungsrichtung der Balken, über Wahl der Parameter für die Slice-Größe, das euklidische Clustering oder den RANSAC-Algorithmus, bis hin zur Bestimmung von Schwellwerten für das Alignment der Balken.

Probleme ergeben sich, bei der Erweiterung des Balkenabschnittes in angrenzende Cluster und an den Kreuzungsstellen mehrerer Balken. Dies wird vor allem dadurch sichtbar, dass die Verschneidung oft nicht korrekt durchgeführt wird, oder die modellierte Hauptrichtung eines Balkens von der tatsächlichen Hauptrichtung abweicht.

Die generische Anwendbarkeit der Lösung ist schwer abzuschätzen. Da jedoch außer der vertikalen Richtung keine explizites Ad-hoc Wissen verwendet wurde, sollte es prinzipiell auch möglich sein, durch Anpassung und Tuning der Parameter auch andere Balkenkonstruktionen zu modellieren.

Die drei Hauptkomponenten des Algorithmus sind aufgrund der unabhängigen Entwicklung, als getrennte Programme geschrieben worden. Jedes Programm läuft für sich ohne manuelle Eingriffe, so dass man sagen kann, die ausgehend von der extrahierten Punktwolke bis zu den modellierten Balken eine sehr hohe, wenn nicht sogar vollständige Automatisierung möglich ist.

Literatur

pointcloud.org. (2013,1). *pcl::EuclideanClusterExtraction< PointT > Class Template Reference*. Von Point Cloud Library (PCL): http://docs.pointclouds.org/1.7.1/classpcl_1_1_euclidean_cluster_extraction.html abgerufen

pointclouds.org. (2013,2). *pcl::PassThrough< PointT > Class Template Reference*. Von Point Cloud Library (PCL): http://docs.pointclouds.org/1.7.1/classpcl_1_1_pass_through.html abgerufen

pointcloud.org. (2014). *The PCD (Point Cloud Data) file format*. Von Documentation - Point Cloud Library (PCL): http://pointclouds.org/documentation/tutorials/pcd_file_format.php abgerufen

- pointclouds.org. (2014,1). *Downsampling a PointCloud using a VoxelGrid filter*. Von Documentation - Point Cloud Library (PCL): http://pointclouds.org/documentation/tutorials/voxel_grid.php abgerufen
- pointcloud.org. (2014,2). *Min-Cut Based Segmentation*. Von Documentation - Point Cloud Library (PCL): http://pointclouds.org/documentation/tutorials/min_cut_segmentation.php abgerufen
- pointclouds.org. (2017). *pcl::ProgressiveMorphologicalFilter< PointT > Class Template Reference*. Von Point Cloud Library (PCL): http://docs.pointclouds.org/trunk/classpcl_1_1_progressive_morphological_filter.html abgerufen
- Scheller, S., & Schneider, D. (2006). Extraktion von Primitiven aus Laserscannerpunktwolken zur Rekonstruktion von Tragwerken.
- Sithole, G., & Vosselman, G. (2003). Comparison of filtering algorithms. *International Archives of Photogrammetry and Remote Sensing*, Vol. XXXIV, 3/W13.