

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 automatische 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 werden die Punktwolke aufbereitet und in ein geeignetes Datenformat zu Weiterverarbeitung exportiert. Dann wird die Punktwolke zurechtgeschnitten und störende Daten 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* erfolgte ausschließlich der Export der Messdaten, da die Registrierung der Aufnahmepositionen bereits vorhanden war. Als Exportformat wurde eine Textdatei gewählt, in der jede Zeile einem Koordinatentripel entspricht. 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*. Diese Konvertierung wird für die Unterstützung einiger Erweiterungen der Punktwolkenverarbeitung in *PCL* benötigt.

Der Algorithmus beginnt den Import der Scanneraufnahmen mit der Zusammenfassung als Merge der Punktwolken. 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 Klasse *PointXYZ* des jeweiligen Scans wird mit *loadPCDFile* eingelesen und mit dem Additions-Operator in die zuvor erstellte leere Punktwolke erweitert. Nach durchlaufen aller Scans wird das Ergebnis des Merges geschrieben. Dieser Teil erfolgt vollautomatisiert.

Nach Zusammenführen aller Scans in eine Punktwolke wird ein Downsample angestoßen. Dazu wird ein *VoxelGrid*-Filter angelegt (basiert auf Algorithmus der *PCL*-Dokumentation). Nebst 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 drei Blattgrößen vor- und ausgegeben. Die Blattgrößen sind auf 5 cm, 10 cm und 20 cm eingeben. Die Blattgrößen werden dabei im Dateinamen mitgegeben, um die verschiedenen Downsample-Ergebnisse für den weiteren Ablauf nutzen zu können. Die recht grobe Wahl der Blattgrößen beruht auf die Arbeit während der Ausarbeitung des Algorithmus. Mit sehr kleinen Blattgrößen ist die Rechenzeit lang und produziert große Dateien. Für einen Finalen Algorithmus kann eine kleinere Blattzahl oder auch das Merge-Ergebnis verwendet werden.

3.2 Segmentierung

Um 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 die effizienteste Wahl 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 und Vosselman, 2003]. Da der Algorithmus sehr rechenintensiv ist, kam als Input nur die downgesampelte Punktwolke in Frage. Die Anwendung auf die originale Punktwolke wurde nach drei Tagen Rechenzeit unterbrochen.

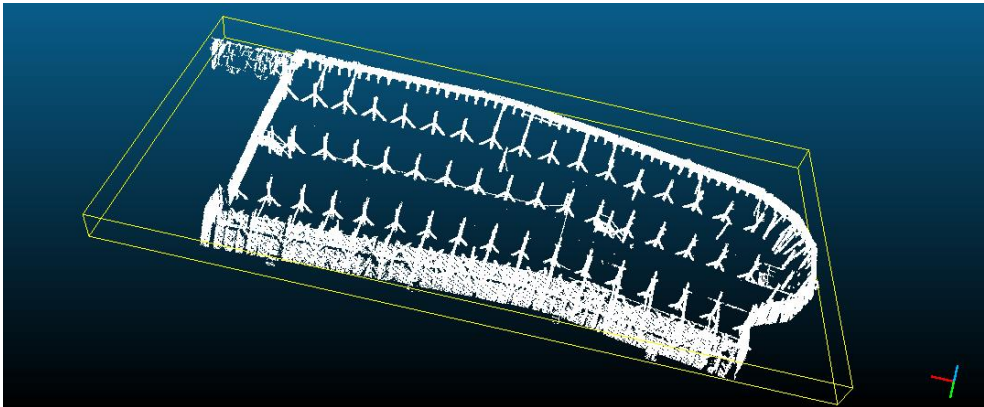


Abb. 2: Balkenwerk nach Filterung von Decke und Boden.

Um in der eingelesenen Punktwolke das Dach zu entfernen, wurde ein *PCL PassThrough* Filter verwendet. Dieser entfernt nicht benötigten Informationen in einem bestimmten Bereich heraus. Die Parametrisierung des *PassThrough* Filters konnten mit Hilfe des Programms *CloudCompare* bestimmt werden, indem eine sinnvolle parametrische Beschreibung der Decke gesucht wurde. In diesem Schritt wurde der Bereich, welcher gefiltert wurde zwischen 0 m und 3 m festgesetzt werden und für „z“ angewandt werden.

Abbildung 2 dokumentiert das Ergebnis der Filterung von Decke und Boden. Nach dem Filterprozess begann die eigentliche Extraktion der Balken. Um bei der Clusterung das bestmögliche Ergebnis zu erzielen, wurde der Anwendungsbereich eingeschränkt. Dies geschah indem ein weiterer *PassThrough* Filter angewandt wurde, welcher Balkenstümpfe bildete. Diese Stümpfe haben den Vorteil ein besseres Ergebnis beim *Clustering* zu liefern, da alle störenden Elemente innerhalb des Balkenwerkes gefiltert wurde und nur die gewünschten Balken vorhanden sind. Die Seitenwände als störende Elemente blieben jedoch erhalten und flossen in die Clusterung mit ein. Mit einer Parametrisierung der z-Achse

zwischen 1.8m und 2m konnte wie in Abbildung 3 dokumentiert, Balkenstümpfe gebildet werden.

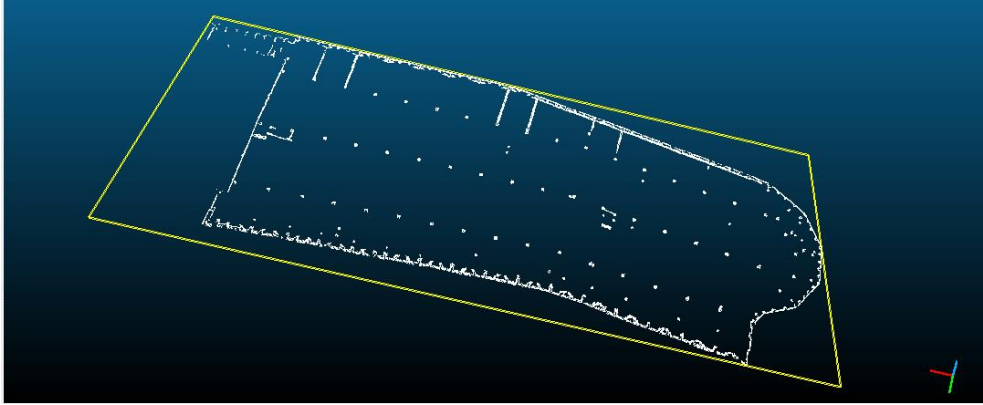


Abb. 3: Balkenstümpfe für das EuclideanClustering

Für die Clusterung der Balken wurde das Euklidische Clusterverfahren verwendet, welches in PCL zur Verfügung gestellt wird. Um die Ergebnisse des Clusterings zu speichern, wurden die Indices extrahiert und mit Hilfe einer Schleife in einer Punktwolke gespeichert. Für jedes Cluster wurde ein Centroid eingepasst, welcher den jeweiligen Mittelpunkt lieferte. Die Gesamtheit der aus den Centroiden gebildeten Punkte, wurden in einer neuen Punktwolke gespeichert, welche die Mittelpunkte aller Cluster enthielt. Abbildung 4 zeigt diese gebildete Punktwolke.

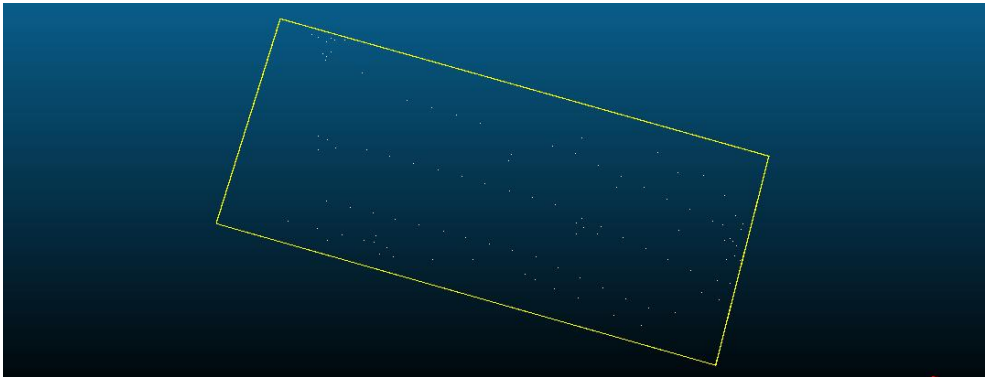


Abb. 4: Punktwolke der Centroiden, welche anhand der Cluster gebildet wurden.

Diese „Mittelpunkt“ Punktwolke enthielt jedoch auch Punkte für die Seitenwände, welche nicht das Balkenwerk widerspiegeln, das extrahiert werden sollte. Hier wurde erneut der *PassThrough* Filter angewandt und die Seitenwände innerhalb der Mittelpunktwolke entfernt. Der gefilterte Bereich für die y-Werte lag zwischen -7 und 14. Diese Spanne konnte mit der zu Hilfenahme von *CloudCompare* ermittelt werden. Alle Punkte innerhalb der „Mittelpunkt“ Punktwolke wurden im darauffolgenden Schritt verwendet, um die Parameter für die MinCut-Segmentierung zu bilden. Anhand einer Iteration, wurde die MinCut-

Segmentierung für jeden Mittelpunkt durchgeführt und der Umkreis, in der diese stattfindet auf 1.5m gesetzt. Dieser Algorithmus bildet eine binäre Segmentierung der gegebenen Eingangswolke. Nach der Objektmitte und ihrem Radius dividiert der Algorithmus die Wolke auf zwei Teile: Vorder- und Hintergrundpunkte (Punkte, die zum Objekt gehören und welche, die nicht dazu gehören). Jeder Balken wurde in eine separate .ply-Datei geschrieben. Um die Anzahl der extrahierten Balken zu ermitteln, wurde jede Extraktion gezählt und in einem Array gespeichert.

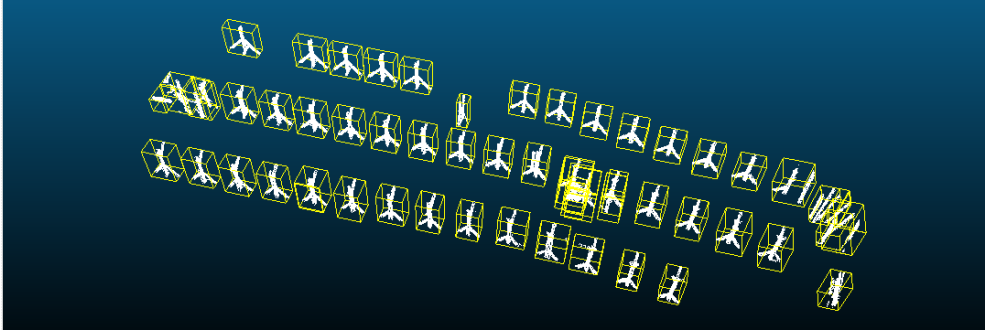


Abb. 5: Ergebnis der MinCut-Segmentierung mit den einzelnen Balken

Die extrahierten Balken weisen 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. Für diesen Schritt wurden alle extrahierten Balken iteriert und für jeden die Bounding Box berechnet. Darauf folgend wurde geprüft, welche Punkte aus originalen Punktwolke innerhalb der Bounding Box lagen. Alle positiven Punkte innerhalb der Box werden in eine neue Punktwolke geschrieben und als .ply-Datei gespeichert. Die Anzahl der Punkte konnte von durchschnittlich 700 auf 50.000 erhöht werden.

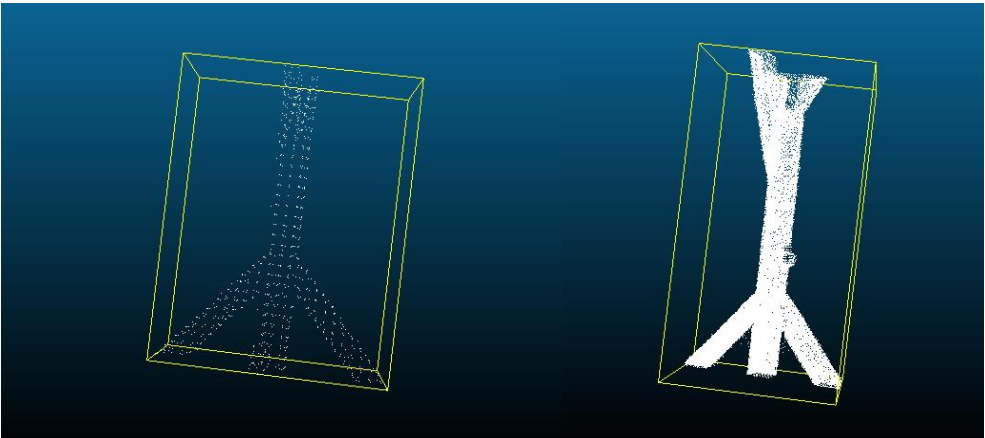


Abb. 6: Vergleich der Balkenextraktion aus downgesampelten und originalen Punktwolke

3.3 Modellierung einer Balkenkonstruktion

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 4 Begrenzungsseiten eines jeden Clusters in unterschiedlichen Ansätzen.

3.3.1 Vorbereitende Phase

Zuerst wird die Punktwolke in horizontale Scheiben entlang der z-Achse in Slices zerschnitten. Für jeden Slice erfolgt ein Clustering mittels euklidischer Distanzclustering. Dadurch zerfällt der Slice in einzelne Cluster. Für jedes Cluster wird die Bounding-Box bestimmt. In benachbarten Slices erfolgt die Bestimmung sich überlappender Cluster anhand xy-Koordinaten der Bounding-Boxen. Ist die Größe der Bounding-Boxen sehr ähnlich, so werden die beiden Cluster in einem gemeinsamen Balkenabschnitt zugeordnet. Durch die Transitivität der 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 dazu zufällig 3 Punkte ausgewählt und 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 Ebenen verworfen und eine neue Ziehung beginnt.

Nachdem durch RANSAC erfolgreich zwei rechtwinklig zueinanderstehende Ebenen detektiert wurden, erfolgt die Rekonstruktion der 4 Seiten. Dazu wird die Schnittgerade errechnet. Der Start- und Endpunkt dieser Kante ergibt sich durch zwei weitere Schnitte mit den oberen und unteren horizontalen Begrenzungsflächen der Bounding-Box. Die fehlenden 3 Kanten des Begrenzungsquaders ergibt sich durch das Verschieben der Kante entlang der 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. Zur Überprüfung wird, dass die erhaltenen Koordinaten aufgespannte Volumen berechnet und mit dem Volumen der Bounding-Box verglichen und welches einen bestimmten Prozentwert nicht unterschreiten darf.

Nach einer erfolgreichen Ausführung erhält man 8 Punkte die einen quaderförmige Begrenzung des im Balken enthaltenen Clusters darstellen.

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 Zwischen erfolgt die Verbesserung der bisher generierten Quader-Modelle. Dazu erfolgt eine Zerlegung der Punktwolke 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. Für die 4 neuen Seitenflächen werden die Eckpunkte des Quaders durch Verschneidung 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 nicht modelliert sind., da in Phase 1 kein Cluster modelliert werden konnte. Im letzteren Fall, wird ein mittlerer Richtungsvektor für die 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 eingepasst, dessen vertikale Kanten dem berechneten Richtungsvektor entspricht und dessen horizontale Kanten entlang der x- und y-Achse verlaufen.

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 mittlere Verschiebevektor berechnet über die einzelnen Quader eines Balkenabschnittes berechnet. Dieser Vektor ergibt sich aus der Differenz der zwei Punkte, die eine senkrechte Kante modellieren. Die mittlere Verschiebevektor ergibt sich aus dem Durchschnitt der 4 vorhandenen senkrechten Kanten. Haben zwei Balkenschnitte sehr ähnlich Verschiebevektoren und gibt es eine lückenlose Kette von Cluster über die diese Balkenabschnitte verbunden sind, so werden beide Balkenschnitte zu einem neuen Abschnitt zusammengesetzt, wobei die dazwischenliegenden Cluster integriert 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 einen Referenz-Quader in einem anderen Balkenabschnitt gibt. Mit die-

sem Referenzquader wird der generierte Zielquader verschnitten. Für die Verschneidung wird im Referenzquader zuerst heuristisch anhand der Ausrichtung und des Abstandes die Fläche ermittelt, welche unmittelbar an den Zielquader angrenzt. 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 dieser 4 Schnittpunkte unterschieden werden:

1. alle 4 Schnittpunkte liegen oberhalb der maximalen z-Koordinate: Der Ziel-Quader bleibt ungeschnitten
2. zwei Schnittpunkte liegen oberhalb und 2 Schnittpunkt liegen zwischen maximaler und minimaler z-Koordinate: Der Zielquader 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 maximalen z-Koordinate und 2 Schnittpunkte liegen unterhalb der minimalen z-Koordinate: Der 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 minimalen z-Koordinate. Der Zielquader ist vollständig abgeschnitten, d.h. das Ende der Balkenverbinding ist erreicht

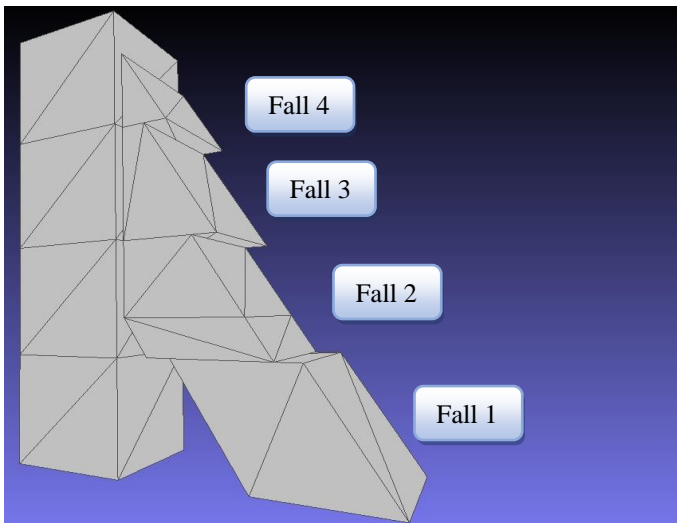


Abb. 7: Darstellung der Verschneidung zweier Balken

Je nach Fall müssen weitere Schnittpunkte horizontaler Seiten mit der Referenzfläche berechnet werden, um die Eckpunkte für den Zielquader auszurechnen. Zusätzlich gibt es stets 4 weitere Unterfälle zu unterscheiden, die sich aus den 4 unterschiedlichen Richtungen ergeben, aus den der Zielquader mit dem Referenzquader zusammenlaufen kann. Außerdem ist zu unterscheiden ob das Zusammenlaufen von unten nach oben oder von oben nach

unten erfolgt. Der letztere Fall lässt sich durch geeignetes Vertauschen der z-Koordinaten der Eckpunkte auf den ersten Fall abbilden

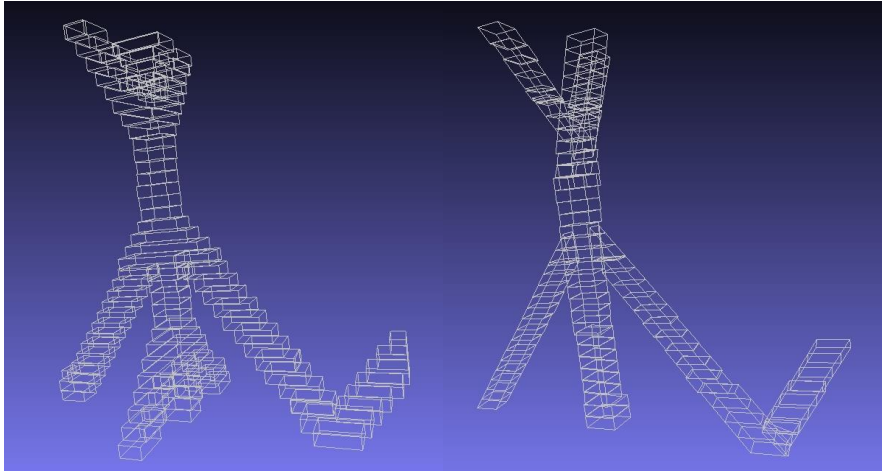


Abb. 8: Gegenüberstellung der Balkenstruktur mit umfassender Bounding-Box (links) und dem Endergebnis der geometrischen Rekonstruktion (rechts).

In Bild 3 links ist die Zerlegung einer Balkenstruktur in Cluster mit den umfassenden Bounding-Boxen dargestellt. Dabei erfolgt die vertikale Zerlegung in 32 Slices. In Bild 3 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 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 Flächen 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 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 Daten.

Dazu wird für jede Box die zugehörige Punktwolke betrachtet. Für jeden Punkt der Punktwolke wird die Distanz zu allen vier Flächen der Box bestimmt und davon wiederum die minimale Distanz ausgewählt. Der Durchschnittswert der minimalen Distanzen aller Punkte bildet den Fehlerwert für die Box. Diese Berechnung wird für alle modellierten Boxen 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 Abweichungen der 5 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.

Als Problem ergibt sich aber daraus, dass sich nur schwer Milestones mit Zwischenergebnissen definieren lassen. Außerdem ist die Aufwandsabschätzung kaum möglich. Dies hat sich insbesondere darin gezeigt, dass zur Rekonstruktion keine Standardalgorithmen verwendet werden konnten.

6 Fazit (und ggf. Ausblick)

Eine erfolgreiche Extraktion der Seitenflächen durch robuste Schätzer ist auf der Grundlage der Daten nicht wirklich 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 gewählt 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 vor allem an den Kreuzungsstellen mehrere Balken. Dies wird vor allem dadurch sichtbar, dass das Alignment nicht immer 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 durch Anpassung der Parameter auch andere Balkenkonstruktionen zu modellieren.

Die drei Hauptkomponenten des Algorithmus sind aufgrund der unabhängigen Entwicklung, als getrennte Programme entwickelt 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

Scheller, S., & Schneider, D. (2006). Extraktion von Primitiven aus Laserscannerpunktvolken zur Rekonstruktion von Tragwerken.