

IDP: Processing of Point Clouds of Excavation Sites to obtain Progress Information

Benjamin Steer¹

benjamin.steer@tum.de

Abstract. Dieses interdisziplinäre Projekt befasst sich mit der automatisierten Volumenberechnung von Baugruben, welche als dreidimensionale Punktwolken zur Verfügung stehen. Diese Punktwolken werden aus Fotoaufnahmen automatisiert erstellt und jede dieser Wolken stellt somit einen gewissen Fortschritt der Baumaßnahme zum Zeitpunkt der Aufnahme dar. Durch die Berechnung der Volumina der verschiedenen Punktwolken soll somit der Fortschritt der Grabungsarbeiten festgestellt und weiteren Analyseverfahren zur Verfügung gestellt werden. Dies ermöglicht bspw. dem Bauleiter eine Baufortschrittskontrolle, welche es ihm ermöglicht Fehler aufzudecken. Auf Grundlage dieser Daten können ebenso die benötigte Anzahl an LKWs für den Abtransport des Aushubs ermittelt werden.

1 Vorgehensweise

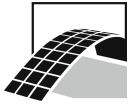
Die automatische Bearbeitung der Punktwolken zur Säuberung, sowie zur abschließenden Berechnung der Volumina besteht aus einigen Schritten, welche sukzessive durchgeführt werden müssen um ein optimales Ergebnis für die einzelnen Punktwolken zu erhalten.

1.1 Entfernung von Artefakten

Bei der Erzeugung der Punktwolken aus den Fotoaufnahmen entstehen durch umliegende Gebäude und Objekte Artefakte, die für die eigentliche Baugrube bedeutungslos und ausserhalb des zu betrachtenden Bereichs liegen. Diese Punkte dieser diversen aussenliegenden Konstruktionen gilt es zu entfernen.

1.2 Ermitteln der Boden- und Grundebene

Um die unterschiedlichen Voluminaberechnungen für eine Baugrube miteinander vergleichen zu können, muss gewährleistet werden, dass als Grundlage immer die gleiche Grundebene benutzt wird. Selbiges gilt auch für die Bodenebene, welche je nach Fortschritt selbstverständlich in einer anderen Tiefe liegt. Punkte, welche oberhalb der Grundebene bzw. unterhalb der Bodenebene liegen, werden dabei verworfen um so nach der Säuberung im Schritt 1 die Punktwolke weiter zu säubern und für die folgenden Maßnahmen vorzubereiten.



1.3 Standardisieren der Punktwolke

Die Punktwolke weist je nach Ort verschiedenen Dichten an Punkten auf. Dies ist für die folgenden Modifikationen an der Punktwolke nachteilig. Aus diesem Grund wird die Punktwolke standardisiert, d.h. es wird eine neue Punktwolke erzeugt, welche feste Abstände zwischen den einzelnen Punkten sicherstellt und somit eine feste Auflösung (= Abstand zwischen den einzelnen Punkten) besitzt. Diese Auflösung, d.h. wie genau die ursprüngliche Punktwolke dargestellt wird ist dabei variabel. Diese Maßnahme ermöglicht die folgenden Bearbeitungsvorgänge.

1.4 Entfernung von vertikalen Duplikaten

Für die spätere Volumenberechnung ist es wichtig, dass vertikale Duplikate, d.h. Punkte, welche vertikal übereinander liegen aus der Punktwolke entfernt werden um eine Verfälschung der späteren Volumenberechnung auszuschließen. Dabei werden falls Duplikate bestehen alle Punkte ausser die am tiefsten liegenden Punkte entfernt. Daraus resultiert eine Punktwolke, welche die komplette Bodenebene der Baugrube darstellt

1.5 Auffüllen von Löchern in der Punktwolke

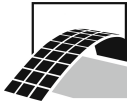
Durch Ansammlungen von Wasser in den Baugruben bzw. durch generelle Fehler in der Erstellung der Punktwolken aus den Fotoaufnahmen stellt die erzeugte Punktwolke meist nur ein Konstrukt mit etlichen Löchern dar. Diese Löcher müssen mit Hilfe der nebenliegenden Punkte geschlossen und die vertikalen Positionen durch diese angenähert werden um eine geschlossene Punktwolke zu erzeugen. Dieser Umstand ist Voraussetzung für den abschließenden Schritt 6.

1.6 Berechnung des Volumen

Durch die vorhergehenden Schritte wurde eine geschlossene Punktwolke mit einer festen variablen Auflösung erzeugt. Die Volumenberechnung wird nun durch Annäherung berechnet. Dies bedeutet, dass wie bei der Annäherung zur Berechnung eines Integrals, dieses Verfahren nun im dreidimensionalen Raum angewandt wird. Dazu benutzen wir unsere feste Auflösung, welche pro Punkt die Grundfläche eines Quaders darstellt und multiplizieren diese mit der Differenz aus Grundebene und Tiefenwert des Punktes. Somit ziehen wir für jeden Punkt der Grundfläche bildlich dargestellt einen Quader nach oben und nähern so über die Summe der Volumina dieser Quader das Volumen der gesamten Baugrube an.

1.7 Alternativmöglichkeiten

Es wäre unter anderem auch möglich aus der Punktwolke, welche neben den XYZ-Werten auch RGB- sowie Normalenwerte beinhalten ein Mesh bspw. mittels Poisson Algorithmus zu erzeugen und dann aus diesem Mesh das Volumen berechnen zu lassen. Diese Möglichkeit wurde allerdings zu Gunsten des oben beschriebenen Verfahrens verworfen, da die automatisierte Verarbeitung



keine korrekte Berechnung des Meshs und somit des Volumens sicherstellen kann. Darüber hinaus musste man dabei immer mit den gegebenen Punktwolken arbeiten, da bei einer Standardisierung großer Aufwand betrieben werden hätte müssen um bspw. Durchschnittswerte von Normalen zu berechnen, welche für den Poisson Algorithmus Voraussetzung sind. Zudem war nicht sichergestellt, dass die gegebenen Normalen auch wirklich korrekt sind oder evtl. durch die Erstellung aus den Fotoaufnahmen fehlerhaft sind (bspw. durch Wasseransammlungen). Auch die konkrete Schließung der Löcher im Mesh, bzw. in der Punktwolke stellte sich als eher schwierig heraus.

Dies lag darin begründet, dass viele gegebene Punktwolken im gegenwärtigen Stadium keine entsprechende Qualität aufwiesen und deshalb für oben genannte Alternativmöglichkeiten für eine automatische Verarbeitung weniger geeignet sind.

Darüber hinaus stand der Zeitaufwand für die Umsetzung, als auch für die Berechnung der einzelnen Wolken in keinem Verhältnis zum Ergebnis, welches wie angedeutet nicht deterministisch ist und stark von der Qualität des gegebenen Dateninputs abhängt.

2 Umsetzung

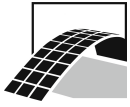
2.1 Verwendete Bibliotheken

Zur Umsetzung werden verschiedene Bibliotheken benutzt. Das Hauptaugenmerk liegt dabei bei der Point Cloud Library (PCL) zur Modifikation der Punktwolken. Weitere verwendete Bibliotheken resultieren aus der Abhängigkeit durch die Point Cloud Library.

2.2 Input Daten

Die Punktwolken liegen allgemein in zwei verschiedenen Auflösungen vor. Diese normalen, bzw. sparse (= spärlichen) Punktwolken unterscheiden sich dabei in der Anzahl ihrer Punkte um ca. den Faktor 10. Grundsätzlich liegt es beim Anwender welche Punktwolken er für die Berechnung der Volumina verwenden möchte, wobei auf der Hand liegt, dass die sparse Wolken eine schnellere Berechnung der Volumina erlauben. Dies ist begründet durch die Tatsache, dass sowohl die Ladezeit, sowie die anfängliche Bearbeitung wie in den folgenden Schritten erklärt, deutlich weniger Datenmengen verarbeiten muss. Differenzen im Ergebnis der Volumina begründet durch die Auflösung der Ausgangswolke werden nachfolgend zu diesem Kapitel behandelt. Neben der gerade erwähnten Tatsache, dass die Punktwolken in zwei verschiedenen Auflösungen vorhanden sind, sind diese im Format PLY kodiert. PLY steht dabei für Polygon File Format oder das Stanford Triangle Format. Wie in im vorigen Kapitel unter dem Punkt Alternativmöglichkeiten bereits beschrieben besitzen die Punktwolken dabei neben den XYZ-Koordinaten auch die Farbwerte im RGB-Format, sowie die Normalen in diesen Punkten.

Für die gewählte Vorgehensweise werden jedoch nur die XYZ-Koordinaten verwendet. Es wäre somit möglich die Datengrößen der einzelnen Punktwolken um rund ein Drittel zu verringern



indem man die Informationen über die Farbwerte, sowie über die Normalen bei der Erstellung der Wolken aus den Fotoaufnahmen nicht berücksichtigt. Dies würde sich in verringerten Ladezeiten der Punktwolke im Zuge des Algorithmus auf der einen und in der verringerten Zeit zur Erstellung der Punktwolke auf der anderen Seite widerspiegeln.

Im Folgenden ein Auszug aus einer Beispiel PLY Input Datei:

```
ply
format ascii 1.0
element vertex 262518
property float x
property float y
property float z
property uchar diffuse_red
property uchar diffuse_green
property uchar diffuse_blue
property int32 number_of_camera_sees_this_point
property float nx
property float ny
property float nz
end_header
-12.0672107105 1.93435489242 -2.93949643381 122 117 116 16 0.975803122996 -0.143215354121 -
-1.01331835048 3.23821363153 7.3257224935 88 82 84 3 -0.598040198947 -0.241207814117 -0.764
-0.0969164778794 4.17806764732 -8.72904718958 125 113 87 6 -0.888734860165 -0.437693211172
-10.3823097022 2.51261203835 4.51548050452 137 138 137 30 0.970506330437 -0.184653428957 -0
```

Hier ist zu beachten, dass die verwendete Point Cloud Library durchaus Probleme hat die PLY Datei zu lesen, falls diese Abweichungen im Header, sowie in den Line Endings am Ende der Zeilen hat. Diese Line Endings müssen dem Windows Format entsprechen, nicht aber bspw. dem UNIX Format. Ein weiteres sich ergebendes Problem war die Bezeichnung der Normalen Properties. Diese wurden in obigem Beispiel bereits angepasst auf nx, ny und nz. Zuvor wurden diese jedoch wie folgt definiert:

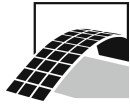
```
property float vsfm_cnx
property float vsfm_cny
property float vsfm_cnz
```

Dies führte zu Fehlern beim Lesen durch die von der Point Cloud Library verwendeten Funktionen. Es steht hierbei also ebenfalls im Raum, ob es nicht sinnvoll wäre, diese gleich bei der Erstellung mit nx, ny und nz zu bezeichnen oder das Problem gänzlich zum umgehen, indem man wie bereits erwähnt gänzlich auf die Normalen verzichtet, da diese für gewählte Vorgehensweise ohnehin unerheblich sind.

In späteren Datensätze wurde die korrekte Bezeichnung für die Normalen verwendet.

2.3 Laden der Input Daten

Die Inputdaten werden mit Hilfe der PCL geladen und in einem von PCL definierten Datenformat, einer PointCloud<PointXYZ> gespeichert. Dabei ist an der Parametrisierung der PointCloud ersichtlich, dass für die Punkte nur die XYZ-Werte, nicht jedoch RGB- oder Normalenwerte geladen



werden. Es wäre hier auch möglich diese zusätzlichen Daten zu laden, welche aber für unsere Vorgehensweise nur unnötigen Ballast darstellen.

Die geladene Punktwolke hat dabei folgende Visualisierung 1.

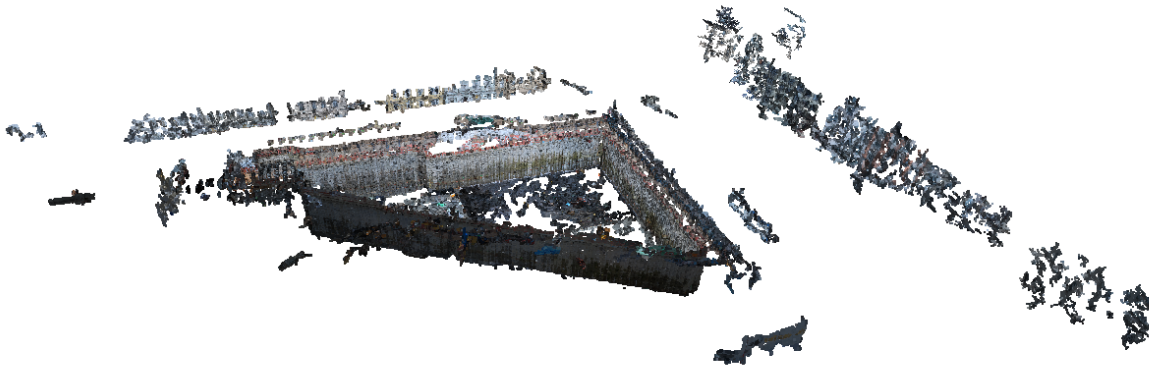
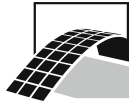


Figure 1: Beispiel der TranslaTUM Baugrube mit sichtbaren Fehler Artefakten

2.4 Clustering der Punktwolke

Zur Säuberung der Punktwolke wie in Kapitel I unter Punkt 1 erläutert, wird ein sogenanntes Conditional Euclidean Clustering verwendet. Dabei werden Punktcluster aus der Eingangspunktwolke erstellt. Alle Punkte in einem Cluster erfüllen zu einem oder mehreren Punkten im zugehörigen Cluster bestimmte Eigenschaften. In unserem Fall wird ein Punkt zu einem Cluster hinzugefügt, wenn er sich innerhalb einer durch einen Parameter vorgegebener euklidischen Distanz befindet und zusätzlich ein bestimmtes Konditional erfüllt, wenn er innerhalb des Radius liegt. Dieses Konditional ist in gegebener Form redundant, da sich ein Punkt für einen Cluster ohnehin schon durch die Erfüllung der euklidischen Distanz qualifiziert. Sollte der Algorithmus jedoch erweitert werden, können dort noch weitere Konditionale wie gleiche Farbwerte oder ähnliche Normalenausrichtungen abgefragt werden. Dies setzt natürlich voraus, dass diese Werte im vorausgehenden Schritt ebenfalls aus der PLY Datei geladen und auch verfügbar sind.

Ist das Clustering beendet entstehen eine Vielzahl von Cluster, welche unterschiedliche Mengen von Punkten aufweisen. Dabei werden alle Cluster mit Ausnahme des Hauptclusters, also der



mit der größten Punktmenge, verworfen.

Die Abbildung 2 zeigt das Ergebnis nach dem Entfernen von Artefakten im Umfeld der Baugrube.

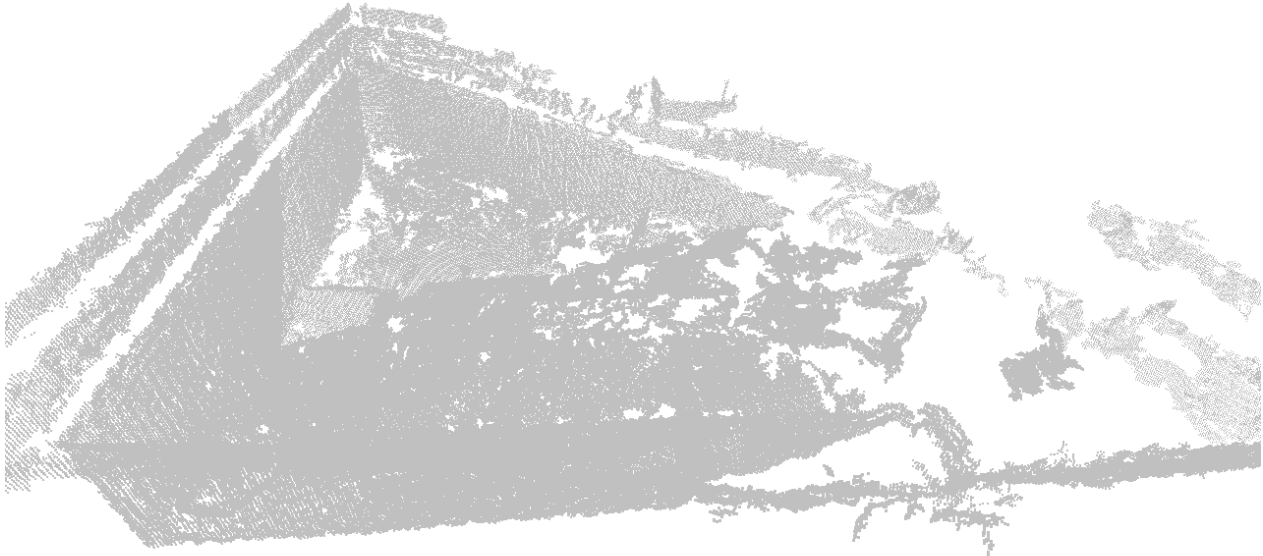
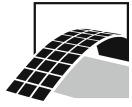


Figure 2: Beispiel der TranslaTUM Baugrube mit sichtbaren Fehler Artefakten

2.5 Ermittlung der Boden- und Grundebene

Als nächstes gilt es die in Kapitel I unter Punkt 2 erwähnte Boden- und vorallem Grundebene zu ermitteln um Punkte die darüber, bzw. darunter liegen zu entfernen. Hier liegt ein besonderes Hauptaugenmerk auf der Grundebene, da diese bei verschiedenen Berechnungen für die gleiche Baugrube immer gleich sein sollte und nicht verschiedene Ergebnisse aufweisen soll um so eine Vergleichbarkeit der Fortschritte zu ermöglichen. Die darunter liegenden Punkte sind im Normalfall bereits durch das vorangehende Clustering entfernt worden.

Um die Grundebene zu ermitteln wird die Punktwolke zuerst in der Vertikalen in verschiedene, gleich große Intervalle unterteilt und für jedes Intervall festgestellt, wie viele Punkte sich in diesem befinden. Wertet man diese Intervalle mit zugehöriger Punktzahl nun aus, ergibt sich eine Kurve. An der Stelle an denen die Kurve fällt, befindet sich die Grundebene. Davor zeigt die Kurve über mehrere Intervalle eine konstante Anzahl an Punkten und somit eine Steigung von ungefähr 0. Dies entspricht den Wänden der Baugrube. Dort wo die Kurve ihr Maxima hat, befindet sich folglich



der Boden, da dort die meisten Punkte vorhanden sein müssen.

Ausgehend von dieser Betrachtung, wird die Boden- sowie Grundebene ermittelt. Zudem wird, falls nötig die Punktwolke automatisch um 180 Grad gedreht, da etliche Ausgangspunktwolken auf dem Kopf stehen.

Zur Visualisierung dieses Sachverhalts sei folgendes Histogramm 3 der Punkte in der y-Ebene zu betrachten.

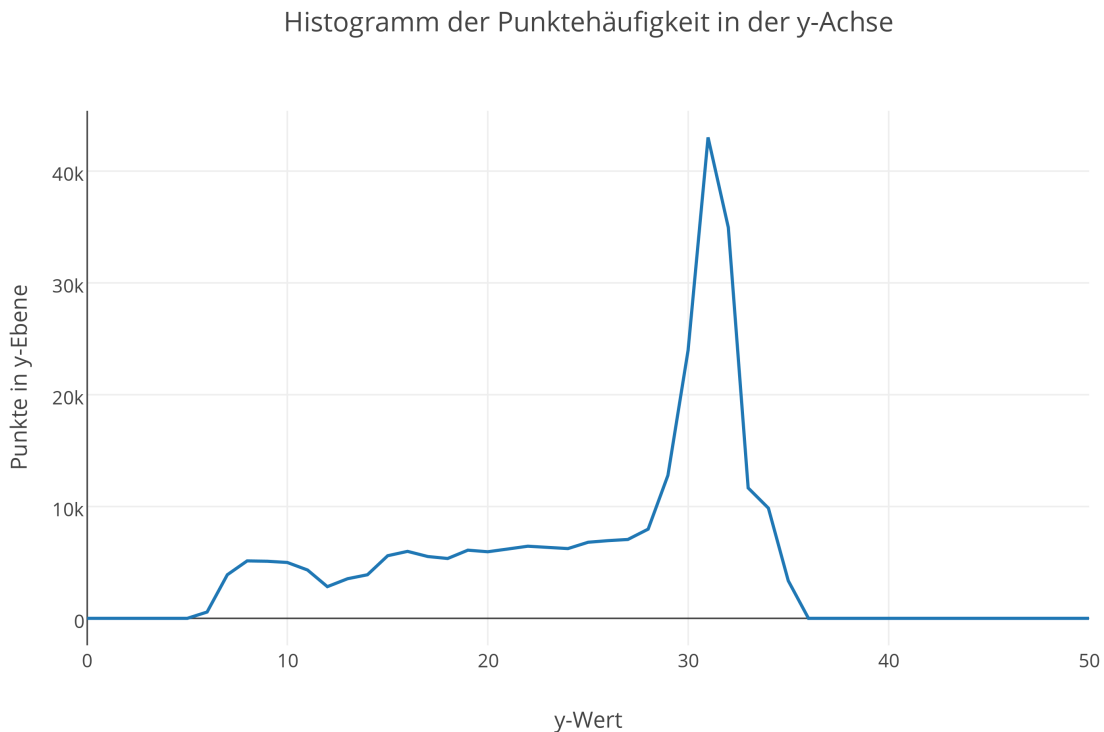


Figure 3: Histogramm der Punkteverteilung der y-Achse

Sind die Boden- und Grundebene ermittelt, werden alle Punkte die auerhalb dieser Grenzen liegen aus der Punktwolke entfernt. Die Punktwolke stellt sich visualisiert nun wie folgt in 4 dar.

Das erstellte Programm bietet zudem die Mglichkeit die Grundebene als Parameter beim Programmaufruf einzugeben, sowie eine manuelle Rotation um 180 Grad durchzufhren. Dies kann durchgefhrt werden, falls die Punktwolke gerade im Bereich der Grundebene eine schlechte Qualitt aufweist und so verschiedene Grundebenen in den verschiedenen Iterationen des Aushubs ermittelt werden und so die Ergebnisse verflschen. Dieses schlechte Qualitt kann auch das Fehlschlagen der automatischen Drehung der Punktwolke mit sich ziehen, weshalb die Mglichkeit

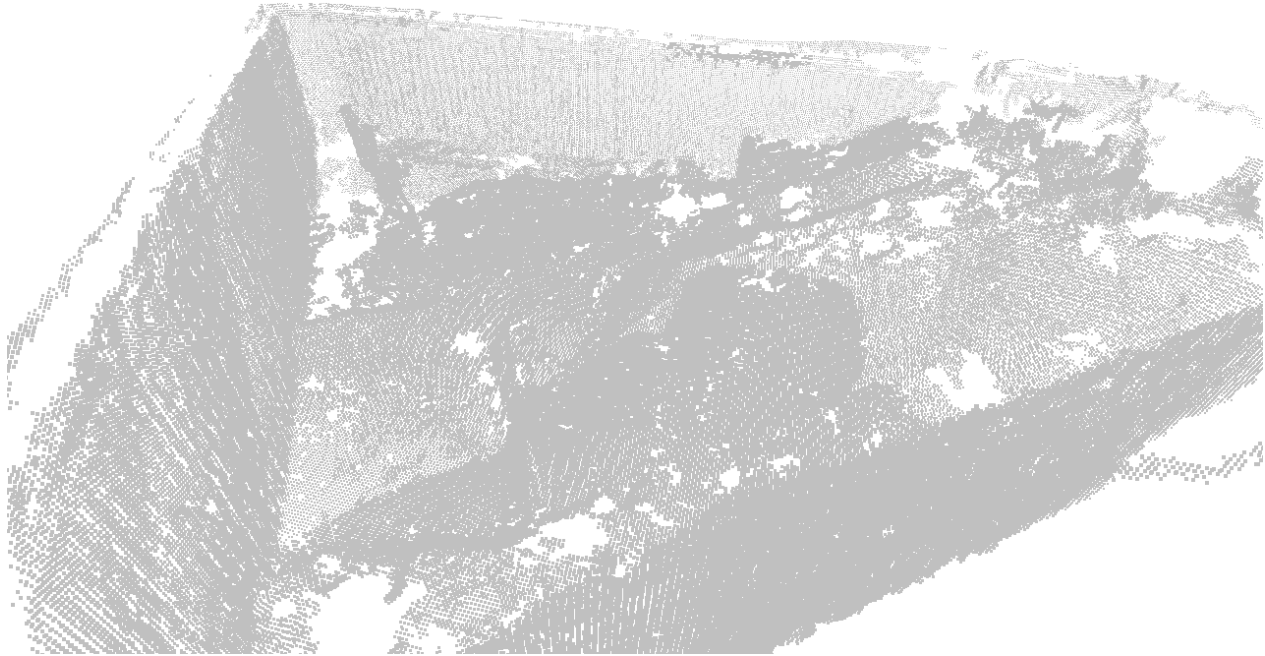


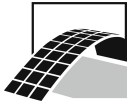
Figure 4: Punktwolke nach Entfernung der Punkte oberhalb der Grund- bzw. unterhalb der Bodenebene

besteht sie ebenfalls manuell über einen Parameter zu flippen. Die Feststellung, ob die Wolke gedreht werden muss, sowie der y -Wert der Grundebene können in einem Programm wie MeshLab bzw. CloudCompare festgestellt und ausgelesen werden. Zu den möglichen Parameter der Anwendung sei auf Kapitel 4 verwiesen. Zu beachten ist hierbei, dass falls man das Mesh manuell dreht, man auch den y -Wert negieren muss.

2.6 Standardisierung der Punktwolke

Im nächsten Schritt gilt es nun eine wichtige Vorbereitung für die Volumenberechnung zu treffen. Die Standardisierung der Punktwolke. Mit Standardisierung ist in diesem Fall gemeint aus der Ausgangspunktwolke eine neue Punktwolke zu erschaffen, welche die alte Punktwolke darstellt, jedoch eine parametrisierte Auflösung besitzt. Wie bereits unter Punkt 3 in Kapitel I beschrieben ist die Ausgangspunktwolke verschieden Dicht und somit ungeeignet für unsere Art der Volumenberechnung. Diese parametrisierte Auflösung kann vom Benutzer festgelegt werden. Falls nicht wird automatisch eine Standardauflösung, welche sich aus den Tests in Kapitel 3 ergeben hat verwendet.

Die resultierende Punktwolke hat folgende Eigenschaft: Alle Punkte sind in ihren jeweiligen Koordinatenachsen ausschließlich ein vielfaches der gewählten Auflösung voneinander entfernt. Visualisiert sieht dies aus frontalen Blickwinkeln wie ein Gitternetz aus und unsere in Kapitel 1.6 beschriebenen Quadergrundflächen werden sichtbar.



Zur Erzeugung dieser standardisierten Punktwolke entfremdet man einen Octree, den man aus der bereits bearbeiteten Ausgangspunktwolke erhält. Dieser Octree unterteilt die gesamte Punktwolke in eine Vielzahl an kleinen Kuben. Die Seitenlänge dieser Kuben ist in unserem Fall eben genau gleich unserer gewählten und durch einen Parameter steuerbaren Auflösung. Der Octree stellt nun eine Ansammlung aller Kuben dar, welche Punkte der Ausgangswolke beinhalten. Dabei spielt es keine Rolle ob dort ein Punkt oder etliche dutzend in einem Kubus liegen. Für unser Vorhaben ist nur wichtig, ob Punkte vorhanden sind oder nicht.

Daraufhin ermittelt man mit Hilfe der PCL alle Mittelpunkte der mindestens mit einem Punkt aus der Altwolke gefüllten Kuben des Octrees und erhält somit die standardisierte neue Punktwolke.

Das Ergebnis dieser Berechnung ist in der folgenden Sektion in 5 sichtbar.

2.7 Entfernen von vertikalen Duplikaten

Da wir für die Berechnung des Volumen nur den Boden der Grube, sowie die Differenz der Bodenpunkte zur Grundebene benötigen, spielen vertikal übereinander liegende Punkte keine Rolle und würden das Ergebnis verfälschen. Aus diesem Grund werden alle Punkte, welche im Sinne ihrer XZ-Koordinaten Duplikate sind aus der Punktwolke entfernt. Mit Ausnahme der am tiefsten liegenden Punkte der einzelnen XZ-Koordinaten, welche den Boden darstellen. Im Allgemeinen sind die Punkte die bei diesem Schritt entfernt werden, Punkte der Aussenwände.

Es ist hierbei möglich, dass neben den Punkten am Boden noch Punkte der Aussenwände übrig bleiben, welche höher liegen, aber keine Punkte unter, bzw. über sich mehr haben. Dies resultiert bspw. aus schiefen Wänden. Da die Punkte das Resultat der vorangehenden Maßnahmen sind sollten sie in die Volumenberechnung mit einbezogen werden, da sie offensichtlich ein Teilvolumen darstellen und für einen Raum in der Baugrube stehen. Alternativ wäre es möglich hier ein erneutes Clustering anzuwenden, woraufhin solche "freien" Punkte entfernt werden würden und ausschließlich die zusammenhängende Bodenplatte resultiert, was zwar einen geordneteren visuellen Eindruck schaffen würde, allerdings das Ergebnis doch zum Teil verfälschen würde. Aus diesem Grund wurde von einem erneuten Clustering abgesehen.

In der Abbildung 5 das Ergebnis dieses Schrittes.

2.8 Schließen der Löcher in der Bodenplatte

Der letzte Schritt vor der Volumenberechnung ist das Schließen aller Löcher in der Bodenplatte. Dies muss gemacht werden, da das Volumina wie in Kapitel 1.6 beschrieben auf Grund der einzelnen Punkte errechnet wird und somit leere Flächen nicht beachtet werden.

Daraufhin werden die XZ-Grenzen der aktuellen Punktwolke ermittelt und alles was im inneren dieser Grenzen liegt mit Punkten gefüllt, falls Punkte im Raster fehlen. Die Y-Werte dieser

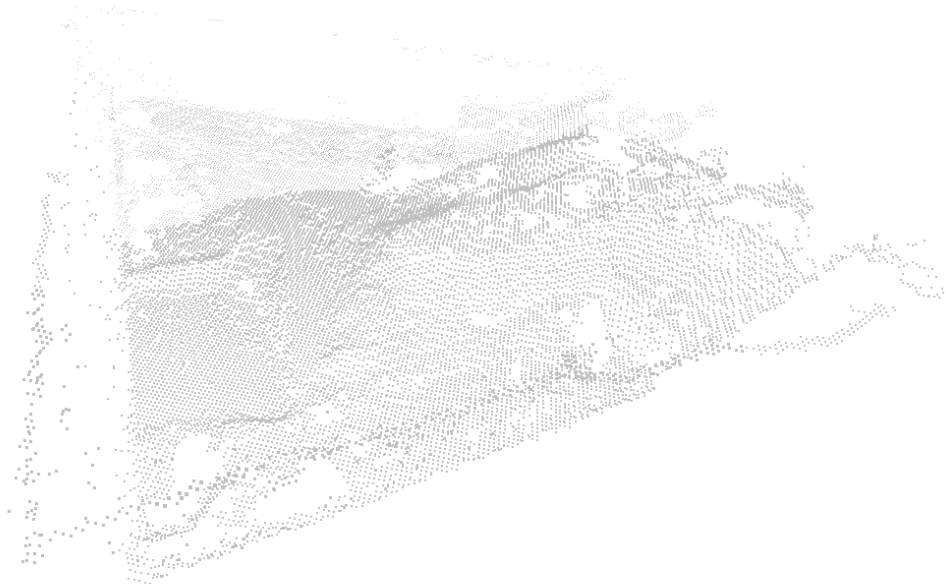
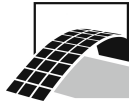


Figure 5: Beispiel der gesäuberten TranslaTUM Baugrube

Punkte werden angenähert und entsprechen entweder dem vorangehend platzierten bzw. gefundenem Punkt, oder aber, falls man das Programm mit dem Parameter `-m` aufruft dem Median 16 umliegender Punkte. Berechnet man den Median erhält man zwar ein besseres Ergebnis, allerdings auf Kosten zusätzlicher Rechenzeit. Grundsätzlich sollte dieser Parameter gerade bei schlechter Qualität der Ausgangswolke, d.h. wenn sie viele Löcher aufweist, gesetzt werden. Auf diese Weise wird die gesamte Bodenplatte mit Punkten gefüllt und ein lückenfreies Gitternetz (frontal von oben betrachtet) entsteht innerhalb der ermittelten Grenzen der Baugrube.

Dies ist in der nachfolgenden Abbildung 6 ersichtlich.

2.9 Berechnung des Volumens

Dieser abschließende Schritt ist im Grunde der einfachste. Alle Vorbereitungen für die Berechnung wurde getroffen und alle Daten sind vorhanden. Jeder Punkt steht für eine Gitterelement res^2 und wird mit der Y-Differenz y_i des Punktes zur Grundebene, d.h. der Tiefe der Baugrube multipliziert. Die Summe dieser vielen Quader ist dann das Volumen der Baugrube.

$$V = \sum_i y_i \times res^2 \quad (1)$$

Hierbei gilt wie bei der Annäherung eines Integrals: Umso höher die Auflösung des Intervalls (in unserem Fall des dreidimensionalen Gitternetzes), umso genauer die Annäherung. Je höher die

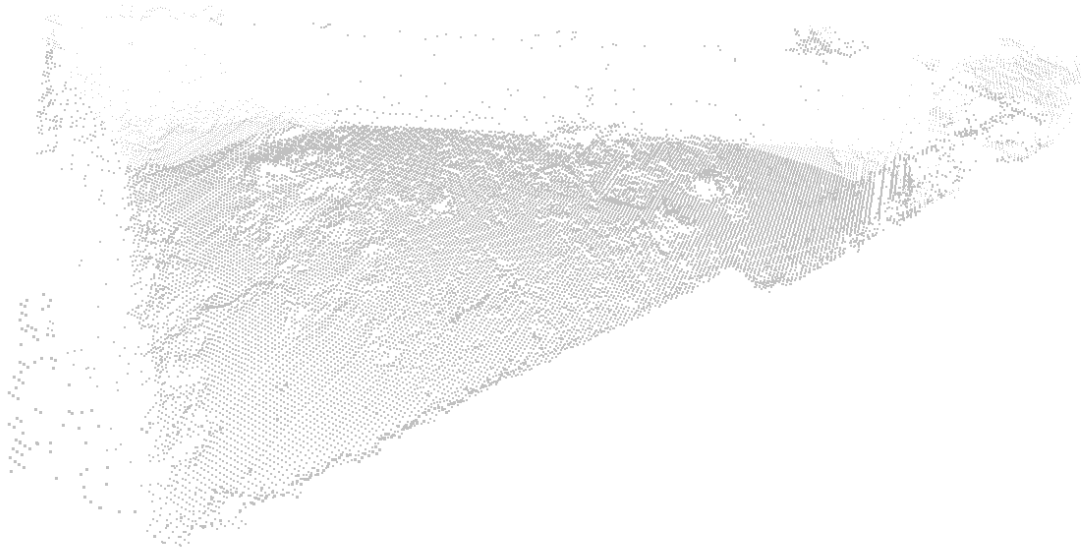
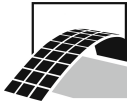


Figure 6: Beispiel aufgefüllten TranslaTUM Baugrube

Auflösung desto höher jedoch auch die Rechenzeit. Die daraus resultierenden Unterschiede werden im folgenden Kapitel noch einmal kurz durchleuchtet.

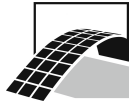
3 Test der Implementierung

Die Implementierung des Algorithmus wurde mit verschiedenen Grundformen der Baugruben exemplarisch getestet. Darunter sind Dreiecks-, Reckecks-, sowie Rundgrundformen.

3.1 Unterschiede zwischen dense und sparse Punktwolken und benötigte Rechenzeiten

Die sparse Punktwolken stellen eine dense Punktwolke in geringer Auflösung dar. Dieser Sachverhalt wird wie im vorherigen Kapitel auch im Zuge des Algorithmus zur Berechnung des Volumens genutzt um eine bessere Performanz des Programms zu erreichen.

Grundsätzlich ist anzumerken, dass sofern die sparse Punktwolke eine höhere Auflösung als die gewählte Segmentierung aus Kapitel 2.6 besitzt, es höchstens eine geringe, vernachlässigte Abweichung im Ergebnis gibt. Ein positiver Aspekt der sparse Punktwolken gegenüber ihren hochauflösenden Pendanten ist die deutlich verkürzte Ladezeit, sowie die schnellere Erstellung des initialen Octrees.



3.2 Unterschiede zwischen verschiedenen Auflösungen und benötigte Rechenzeiten

Generell gilt, umso höher die in Kapitel 2.6 gewählte Segmentierung, umso genauer das Ergebnis der Volumenberechnung. Dies zieht allerdings eine erhöhte Rechenzeit mit sich. Im folgenden wurde die Baugrube an der Erich Mann Straße einem Test mit verschiedenen Segmentierungen unterzogen. Die gewählte Standardparameter für die Segmentierung beträgt 1200. Dieser wird intern nochmals verdoppelt, was 2400 Segmente in jeder Achse entspricht. In folgenden Testfällen wird eine Punktwolke mit 3.267.389 Punkten untersucht. Dabei wird das Volumenergebnis, sowie die dafür benötigte Zeit mit den Programmparametern 400, 1200 und 2000 betrachtet. Zusätzlich wurde das Programm dabei mit dem Parameter -m gestartet, welches beim Setzen eines fehlenden Punktes umliegende 16 Punkte um die Fehlstelle überprüft und einen Mittelwert bildet um eine homogene Annäherung an die Umgebung sicherzustellen.

Dabei nimmt das Laden der Punktwolke jeweils bereits 49s in Anspruch.

| Auflösung | Volumen | Zeit |
|-----------|---------|-----------|
| 400 | 271.441 | 01min:03s |
| 1200 | 240.634 | 05min:43s |
| 2000 | 239.298 | 07min:01s |

Es ist erkennbar, dass der Unterschied zwischen dem Auflösungsparameter 400 und den beiden restlichen sich im berechneten Volumen niederschlägt. Die unterschiedlichen Ergebnisse sind in 7, 8 und 9 zu sehen.

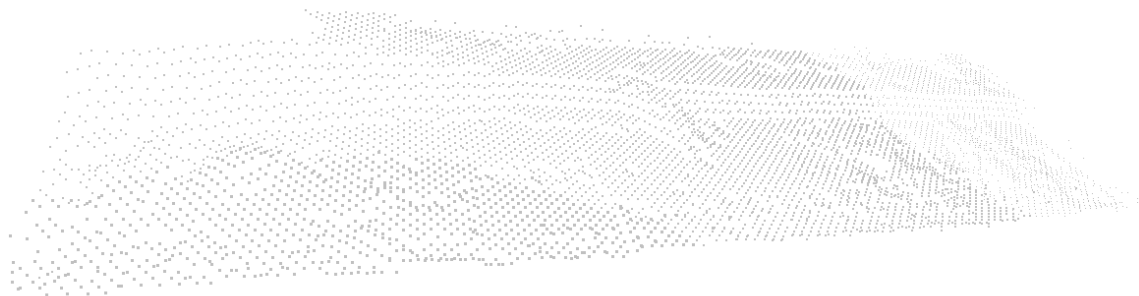


Figure 7: Erika Mann Straße mit Segmentierungsparameter 400

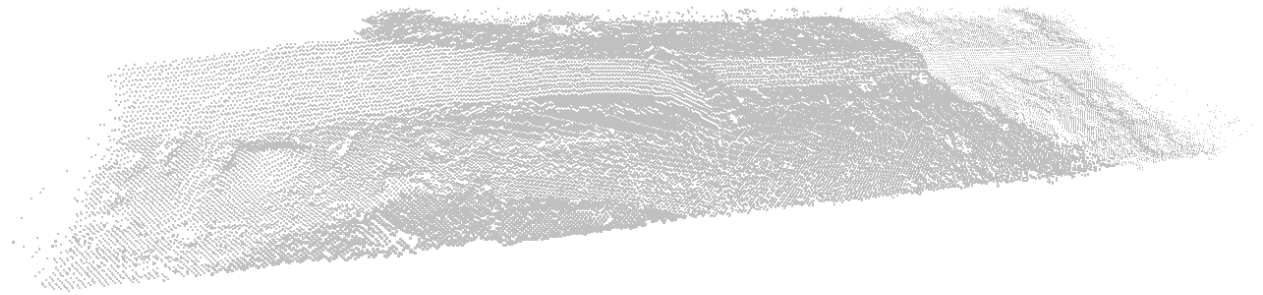
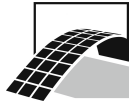


Figure 8: Erika Mann Straße mit Segmentierungsparameter 1200

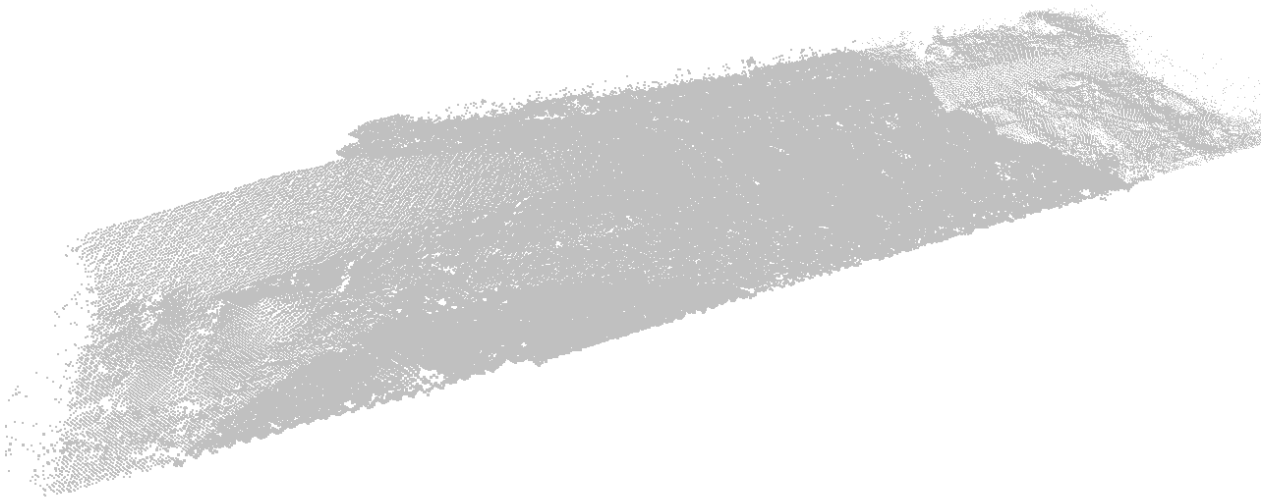
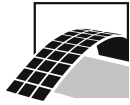


Figure 9: Erika Mann Straße mit Segmentierungsparameter 2000



4 Programmaufruf und Parameter

Das Programm besitzt etliche benötigte, sowie optionale Parameter, welche zum Teil in den vorherigen Kapitel bereits erläutert wurden. Ein Beispiel Aufruf sieht dabei wie folgt aus:

```
./CleanMesh -f translaTUM/2014-09-30.ply -o translaTUM_cleaned -s 1200 -m -r -b -0.586391
```

Dabei gilt:

```
-f      Pfad zur Punktwolke
-o      Directory für gesäuberte Punktwolken, sowie Datei mit Volumina
-s      Grad der Segmentierung (Standard = 1200)
-m      Medianberechnung (16 umliegende Punkte werden in y-Wert miteinbezogen)
-r      Rotate des Meshs um 180 Grad
-b      manuelle Angabe der Grundebene (falls ebenfalls rotiert wird, negieren)
```

5 Anhang

Verwendete Bibliotheken und Abhängigkeiten

- Point Cloud Library 1.7.2
- boost 1.56.0
- Eigen 3.2.2
- FLANN 1.8.4
- VTK 6.1.0
- QT 4.8.6

Verwendete Zusatzsoftware

- MeshLab 1.3.3
- CloudCompare 2.6.0