

FACHHOCHSCHULE AACHEN

- CAMPUS JÜLICH -

FACHBEREICH 9 - MEDIZINTECHNIK UND TECHNOMATHEMATIK

Verarbeitung von Punktwolken im Baubereich

SEMINARARBEIT

IM STUDIENGANG SCIENTIFIC PROGRAMMING

Autor: Patrick Dautzenberg
<patrickdautzenberg@gmx.de>
MatNr. 3145268

Version vom: 16. Dezember 2019

1. Prüfer: Prof. Dr. Martin Reißel

2. Prüfer: Jan Martens, M.Sc.

Eidesstattliche Erklärung

Hiermit versichere ich, dass ich die Seminararbeit mit dem Thema

Verarbeitung von Punktwolken im Baubereich

selbstständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel benutzt habe, alle Ausführungen, die anderen Schriften wörtlich oder sinngemäß entnommen wurden, kenntlich gemacht sind und die Arbeit in gleicher oder ähnlicher Fassung noch nicht Bestandteil einer Studien- oder Prüfungsleistung war.

Ich verpflichte mich, ein Exemplar der Seminararbeit fünf Jahre aufzubewahren und auf Verlangen dem Prüfungsamt des Fachbereiches Medizintechnik und Technomathematik auszuhändigen.

Name: Patrick Dautzenberg

Aachen, den 14.12.2019

Dautzenberg

Unterschrift der Studentin / des Studenten

Abstrakt

Verarbeitung von Punktwolken im Baubereich

Aufgabe eines Architekten ist es, zu Anfang den Plan eines Gebäudes für den Bau zu erstellen. Diese sind mittlerweile digital und dadurch sehr informationsreich. Ein Grundriss eines beliebigen fertigen Gebäudes (Altbauten mit eingeschlossen) wird für Bearbeitungen und Nachsichten zurückgelegt. Aktuelle Pläne solcher Building Information Models (BIM) können anschließend jederzeit mit Hilfe einer Punktwolke per Hand nachmodelliert werden oder aktualisiert werden. Diese müssen jedoch zuerst durch Laserscanning über Punktvermessungen erstellt werden. Solche Aufnahmen/Scans können Punktkoordinaten sowie -Farben, Laserintensität und Weiteres enthalten, um schließlich bis hin zu einer kompletten virtuellen Gebäudeabbildung zu führen.

Ein Abgleich des digitalen Bauplans mit einem aktuellen Bestandsmodell liefert jederzeit einen Überblick über Änderungen, die Wartung und ein Facility Management baut auf diesen Darstellungen auf. Dafür kann auch ein Punktwolken-Scan mit einem fertigen Modell verglichen werden. Solche Zuteilungen lassen sich anschließend mit Hilfe von Programmen analysieren und bearbeiten. Beispielsweise wäre dies mit der Software Revit möglich, welche während diesem Seminar auch genutzt wird.

Da dieses Programm jedoch auf Gebäudemodellierung ausgelegt ist und für Punktwolken nur eingeschränkte Funktionen zur Verfügung stellt, werden neue, eigens erstellte, Funktionen über Plugins eingefügt. Diese dienen anschließend der *Verarbeitung von Punktwolken*.

Diese Seminararbeit dient der Beschreibung von Techniken zur Punktwolkenverarbeitung mithilfe solcher Plugins anhand mehrerer Beispiele.

Inhalt

Abstrakt

1 Die Punktwolke	1
1.1 Erzeugung	2
1.1.1 Aufnahmearten und Geräte	2
1.1.2 Intensität	4
2 Punktwolken-Verarbeitung	5
2.1 Bearbeitung mit Software	7
2.1.1 Triangulierung	7
2.1.2 Programm - Revit	8
2.2 Eigene Handlungen	11
2.2.1 Erstellung -> Creator	11
2.2.2 Rotation -> Dreher	12
2.2.3 PunkteVerteilung -> BoundingBox/-Collection	12
2.2.4 Heat-Map	13
2.2.5 Vergleich	14
3 Ausblick und Zukunft	15
3.1 Eigene Plugins	16
Anhang	17

1. DIE PUNKTWOLKE

Über Punkte wie $p = (x, y)$, bzw. Vektoren ($\vec{v} = \begin{pmatrix} x \\ y \end{pmatrix}$), lassen sich in einem Koordinatensystem (Breite,Höhe) grobe Formen darstellen. Färbt man diese Punkte auch noch ein, so erhält man richtige Bilder. Mit weiteren Standort-Informationen, wie der Tiefe, schafft man sich durch mehrere Punkte $\vec{v} = \begin{pmatrix} x \\ y \\ z \end{pmatrix} \in \mathbb{R}^3$ im dreidimensionalen Raum mit $x = \text{Breite}$, $y = \text{Höhe}$ und $z = \text{Tiefe}$ eine *Wolke aus Punkten*. Durch richtige Anordnung lässt sich mit dieser die Rekonstruktion eines kompletten Gegenstandes, so wie ein Büro in Abbildung 1.1, darstellen. Diese Gegenstände können natürlich auch fiktiv sein.

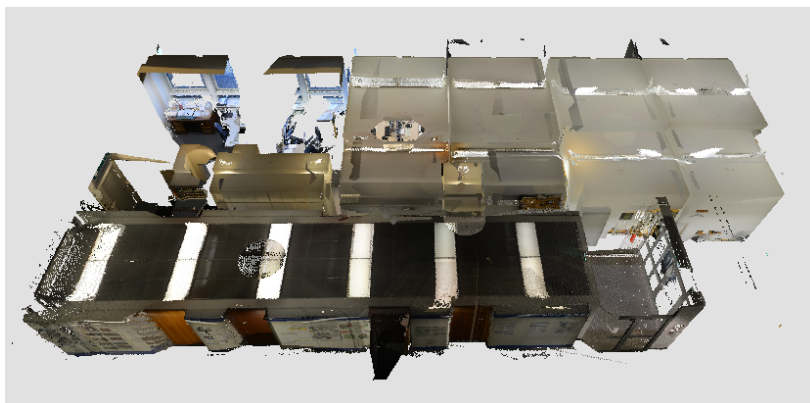


Abb. 1.1: Eine Büroabteilung, abgebildet aus ungefähr 1.6 Millionen farbigen Punkten, gespeichert über 60 mb

Aufnahmen/Scans einer Bauarbeit/eines Gebäudes können mit Sensoren in unten erwähnten Geräten aufgezeichnet werden. Diese analysiert man mit Hilfe von Programmen (Kapitel 2.1), um von Abweichungen oder sogar Fehlern gegenüber des vorher angelegten Bauplans auszugehen. Es können Änderungen vorgenommen werden, um sich Vorstellungen möglicher Umbauten virtuell anzusehen[1].

1.1 Erzeugung

Gewöhnliches Laserscanning arbeitet mit gesendeten Lichtimpulsen und der Dauer einer Reflexionsdetektion, gemessen durch einen Photomischdetektor. Mit diesen Werten wird die Entfernung festgelegt. Verfahren des Mobile-Mapping-System (MMS) dienen als Oberflächen-Scanner und stellen eine Liste von kartesischen Koordinaten im Standard Triangulation Language (STL)-Format zusammen. Entfernung und Winkel zum bekannten Punkt des Scanners werden über Algorithmen wie Simultaneous Localization and Mapping (SLAM) berechnet und festgehalten, um damit zu arbeiten.

1.1.1 Aufnahmearten und Geräte

Das Geodätische Institut und Lehrstuhl für Bauinformatik & Geoinformationssysteme (GIA) nutzt folgende Aufnahmearten und Geräte:

- a) Mobiles Laserscanning (MLS) wird während einer beliebigen Strecke (zu Fuß, per Fahrt) genutzt, um einzelne kleine Teilpunktwolken auf gleichzeitig erfasstem Weg zu erstellen. Dadurch kann man sie später zusammenfügen. Die Genauigkeit eines solchen Scanners ist einigermaßen gering, so dass mit unbekannter Laufbahn die Qualität der Aufnahme im Vergleich zu unten beschriebenem Terrestrisches Laserscanning gering ist. Ein mögliches Gerät ist das ZEB-REVO, mit dem sich schnell ein komplettes Gebäude scannen lässt.
- b) Photogrammetrie basiert auf der Erstellung von Bildern mit Kameras, die zueinander registriert sind. Über die Berechnung einer Kameradistanz und

1.1. ERZEUGUNG

Erstellung einer Tiefenkarte mit Stereoalgorithmen oder Structure from Motion (SFM). Eine Kamera-Methode ist gegenüber den anderen Aufnahmemethoden super günstig, führt jedoch auch zu höherer Ungenauigkeit als im MLS, vor allem bei gleichfarbigen Bildabschnitten.

- c) Terrestrisches Laserscanning (TLS) dient sehr hoher Genauigkeit aufgrund vieler Punkte und wenig Rauschen, auch wenn die Scanner an fester Position ihre Einzelscans relativ langsam aufnehmen. Diese werden anschließend zu einer kompletten Punktwolke zusammengesetzt (Registrierung). Durch Iterative Closest Points (ICP) findet man die übereinstimmenden Punkte mehrerer Scans und kann diese zu einer einzelnen Punktwolke zusammensetzen. Namen möglicher Geräte sind BLK360 oder RieglVZ400
- d) Time-of-Flight (TOF) nutzt Spezialkameras wie eine Kinect oder Google-Tango. Es werden ebenfalls Bilder mit eingebauter Tiefenkarte über die Distanz zur Kamera erfasst. Die Zusammensetzung der Einzelbilder erfolgt anschließend z.B mit SLAM (Registrierung). Der günstige Preis hat eine geringe Genauigkeit wegen Messrauschen zum Nachteil.

Das zu erstellende Dateiformat (siehe unten, 1.1) ist hauptsächlich frei wählbar. Aktuelle Dateien im e57-Format bauen auf einem XML-Baum auf und können somit einige Informationen übersichtlich enthalten. Den üblichen Koordinaten wird eine Intensität und ein Farbwert hinzugefügt.

Punktwolkenformate(üblich):

Format	Archivierung	Aktualität
e57	XML	offener Standard, aktuell
LAS	binär	offener Standard, aber veraltet
PLY	ASCII, binär	offener Standard
PTS	ASCII	existiert, sehr unhandlich
RCS	binär	proprietär, von Autodesk(Recap)

Tab. 1.1: Formate der Dateninformation einer Punktwolke, alle untereinander konvertierbar

1.1.2 Intensität

Ist die Darstellung allerdings nicht allzu exzellent, trifft man die Einstellung, während einer Aufnahme mehr Punkte speichern zu lassen, denn so bildet man sich eine höhere *Genauigkeit*. Dieses Vorhaben führt allerdings auch schnell zu einer hohen Datenmenge und einer millionenhohen Anzahl an Punkten bei einem Scan von ganzen Gebäuden. Je höher die Menge der Punkte, umso niedriger kann man die Fehlerquote/Abweichung einstufen. Die Genauigkeit steigt wenn die Anzahl der Punkte einer Punktwolke(Flächenbasiert, Volumenbasiert) zunimmt.

2. PUNKTWOLKEN-VERARBEITUNG

Zu Anfang sollte die komplette/riesige Menge an gesammelten Punkten geglättet (sortiert und gefiltert) werden, um durch das Entfernen der Ausreißer den *Rauschanteil* zu verringern.[2]

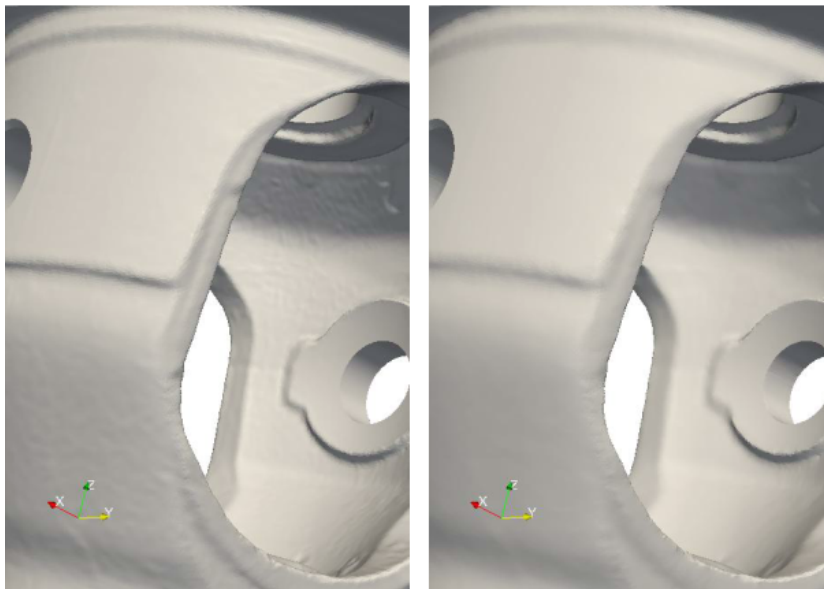


Abb. 2.1: Gussbauteil: Links:Original, Rechts: Geglättet

Eine Datenaufbereitung aus verschiedenen Messungen in einem gemeinsamen Koordinatensystem, gibt die Möglichkeit einzelne Flächenstücke zu generieren. Das Erzeugen des gesamten CAD - Modells folgt.[3]

Fügt man einer Punktwolke die Geometriedaten des Gegenstandes hinzu, lässt sich diese in ein Computer Aided Design (CAD) oder Building Information Modelling (BIM)(mit höherem Informationsgehalt) schreiben.

$$\text{Punktwolke} \Rightarrow \begin{cases} \text{Computer Aided Design (CAD)} \\ \text{Building Information Modelling (BIM)} \end{cases}$$

Ein CAD und das BIM gehören ebenfalls zu den Formaten eines virtuellen Abbild, sind jedoch viel informationsreicher. Sie können beispielsweise vor einem Gebäudebau erstellt werden, um diesen zu führen. Ein BIM-Manager erstellt solche Baupläne für ein Unternehmen, so wird eine vorherige Kostenabschätzung über semantische Informationen der Baumaterialien von Bauteilen durchgeführt. Möglichst viele Einzelheiten werden hinzugefügt, um während eines Baus keine Fragen zu haben und damit vor allem keine Fehler auftreten. [4]

Vorher erwähnte *Genauigkeit*, beeinflusst durch Scan-Intensität, wird auch als Level of Detail (LOD) im Sinne des Fertigstellungsgrades definiert. Ein Grad definiert den notwendigen Informationsgehalt und Detailierungsgrad [5].

Die Level of Detail (LOD):

1. LOD 100 (konzeptionell)
2. LOD 200 (ungefähre Geometrie)
3. LOD 300 (genaue Geometrie)
4. LOD 350 (Zwischenstufe, eingeführt für die Darstellung von Schnittstellen zw. Disziplinen)
5. LOD 400 (Ausführung)
6. LOD 500 (Bestandsdokumentation)

2.1. BEARBEITUNG MIT SOFTWARE

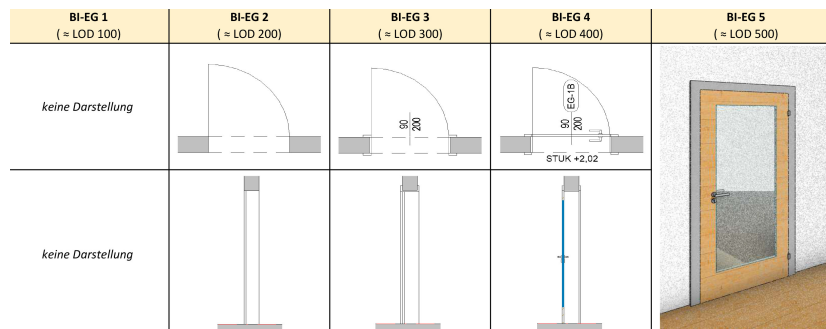


Abb. 2.2: Level-of-Detail

2.1 Bearbeitung mit Software

Es gibt Programme auf deren Ebene man beispielsweise mit Tabelle_1.1-Dateien (Punktwolken -Scans) arbeiten kann. Die zu Anfang gespeicherte Menge an Punkten sollte nicht unbedingt komplett direkt zur Darstellung genutzt werden. Eine Filterung, Sortierung und Ausrichtung sorgt erst einmal dafür, dass die Punktwolken problemlos geladen, bzw. in einem dreidimensionalen Raum wieder dargestellt werden können. Bild 1.1 zeigt eine in MeshLab geladene Punktwolke. Dafür werden die Koordinaten aus der Datei benötigt. Auch Informationen, wie einzelne Punkt-Farben, können hinzugefügt werden und auf diese Weise das Bild des Gegenstands näher an die Realität führen.

2.1.1 Triangulierung

Über die Punkte eines Gegenstandes werden kleine Dreiecksebenen gebildet, so dass diese nachher wie eine große Ebene/Form rekonstruktiv wirken (Abb. 2.3) [3]. Dieser Vorgang nimmt jedoch viel Zeit in Anspruch.

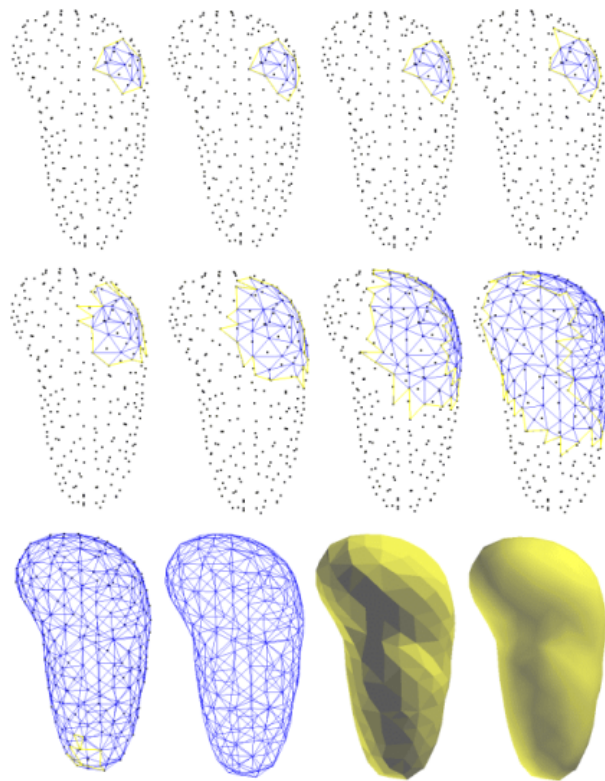


Abb. 2.3: Unten links sind es ca. 500 erzeugte Dreiecke, mit anschließender Schattierung

Die meisten Programme benötigen allerdings, für die komplette Darstellung, viel Arbeitsspeicher aufgrund der großen Anzahl von Punkten und besitzen somit eine schwer zu bedienende Oberfläche.

2.1.2 Programm - Revit

Die Software „Revit“ ist zwar kostenpflichtig, jedoch auch als Studentenversion vorhanden. Diese lädt zum Vorteil nur nötige Punkte, die angezeigt werden (nicht-sichtbare Punkte zum Beispiel nicht). Somit wird weniger Arbeitsspeicher gebraucht, was bei der Arbeit mit den Punktwolken hilft.

2.1. BEARBEITUNG MIT SOFTWARE

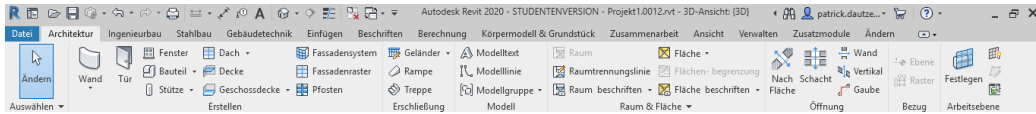
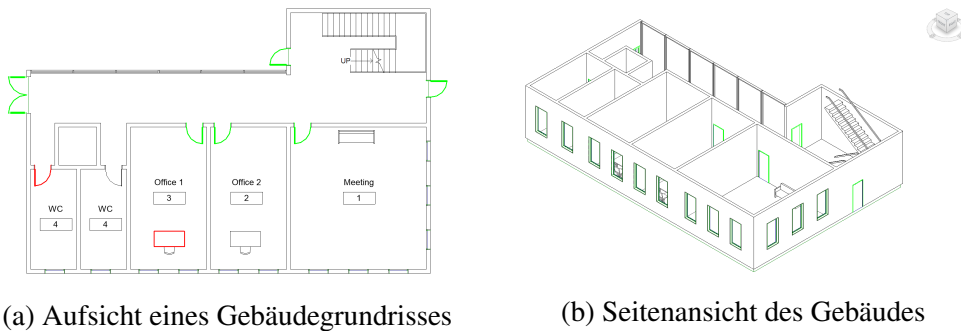


Abb. 2.4: Erste Handlungen unter Revit, wie das Einfügen einer Wand(links)

Mit einem Programm wie Revit lässt sich durch legen eigener Wände(Abb. 2.4), per Hand selbst ein parametrisches Abbild eines Gebäudes erstellen. Wand für Wand wird von Hand eingefügt und größengerecht aneinander positioniert. Anschließend können Gebäudeabschnitte definiert, und weitere Stockwerke aufgesetzt werden. Je nachdem wie weit man in das LOD gehen möchte oder zu gehen hat(siehe oben), kann man auch noch komplexe CAD-Bauteile, wie einzelne Türen oder Fenster, hinzufügen. Es können Exports, wie in die Industry Foundation Classes (IFC), durchgeführt werden. So können andere Programme mit angelegten Informationen arbeiten.



(a) Aufsicht eines Gebäudegrundrisses

(b) Seitenansicht des Gebäudes

Abb. 2.5: Gebäudegrundriss mit Revit erstellt aus Wänden,Fenstern und Türen

Solch einem aufgestellten Gebäude fehlen nur noch die Geometriedaten, um als Bauplan zur Verfügung gestellt zu werden.

2.1. BEARBEITUNG MIT SOFTWARE

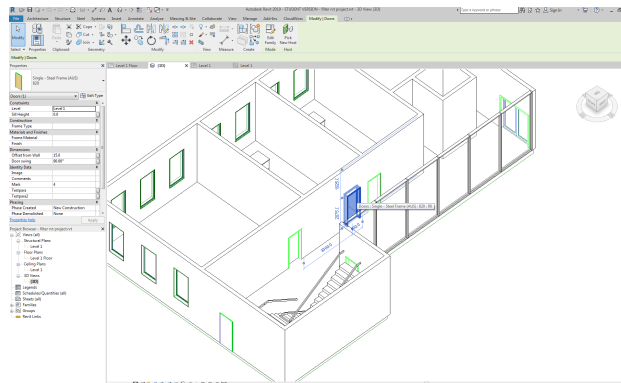


Abb. 2.6: Eine ausgewählte Tür in einem Gebäudeplan im rvt-Format und dessen Informationen über Breite, Höhe und andere Abstände

Punktwolken werden allerdings nach dem Einfügen in Revit keine weiteren Möglichkeiten gegeben, als sich verschieben/drehen zu lassen. Sie können nicht mit erstellten Gebäudemodellen verglichen werden oder wenigstens eine Ausrichtung automatisiert durchführen. Dazu benötigt man wiederum eigens hinzugefügte Erweiterungen:

Eigene Funktionen/Plugins(Abb. 2.7) lassen sich, nach Entwicklung in vorzugsweise C-Sharp(C#), einbinden und ausführen, um mit geladenen Punktwolken/Objekten und den Erstellungen zu agieren.

2.2 Eigene Handlungen

Da das Programm diese Option nicht bietet, gehört die Einblendung eines Koordinatensystems zu den selbstgeschriebenen Plugins. Mit einer Ausgabe der generellen Anzahl an enthaltenen Wänden, verschaffen die Achsen einen weiteren Überblick. Beide Funktionen sind in Abbildung 2.7 als Eingefügt zu sehen.

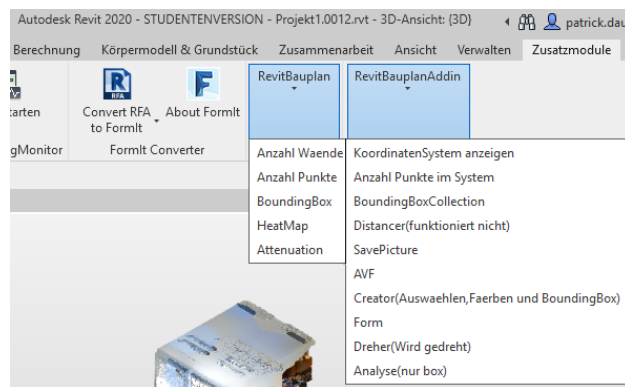


Abb. 2.7: Eigene Plugins hinzugefügt in angelegten Ordnern RevitBauplan/RevitBauplanAddin unter Zusatzmodule

2.2.1 Erstellung -> Creator

Liniendarstellung (*Creator.Create3DModelLine* im Anhang zu sehen) von Punkt zu Punkt wird erstmals zur Erstellung eines Koordinatensystems gebraucht. Generell werden auch später Linien durch den Creator erzeugt.

```
class KoordinatenSystem : IExternalCommand
{
    public Result Execute(ExternalCommandData commandData, ref string message, ElementSet elements){
        Main.reloadStatics(commandData);
        XYZ ursprung = new XYZ(0, 0, 0);
        Creator.Create3DModelLine(ursprung, new XYZ(300, 0, 0)); //X-Achse
        Creator.Create3DModelLine(ursprung, new XYZ(0, 300, 0)); //Y-Achse
        Creator.Create3DModelLine(ursprung, new XYZ(0, 0, 300)); //Z-Achse
        return Result.Succeeded;
    }
}
```

Abb. 2.8: Anzeigen der Achsen eines einfachen dreidimensionalen Koordinatensystems

2.2.2 Rotation -> Dreher

Drehen um eine Achse mit bestimmtem Winkel (Grad, Bogenmaß) dient dem weiteren Ausrichten gegenüber anderer Flächen und Objekte, um für höchstmögliche Übereinstimmung zweier Elemente zu sorgen.

```
public static bool Rotate(Element element, Line axis, double winkel)
{
    bool rotated = false;

    // Rotate the element via its location curve.
    LocationCurve curve = element.Location as LocationCurve;
    if (null != curve)
    {
        rotated = curve.Rotate(axis, winkel);
    }

    return rotated;
}
```

Abb. 2.9: Zur Rotation eines Elements im System

2.2.3 Punkteverteilung -> BoundingBox/-Collection

Da sich mit einer Programmvariablen PointCollection nicht mehr als 999.999 Punkte erfassen lassen, muss die BoundingBox, welche eine „Box“ um die komplette Punktwolke bildet, gesplittet werden, falls diese aus mehr besteht. Für kleinere Abteilungen kann dies wieder zutreffen, von daher wird es möglicherweise weiter durchgeführt und bildet einen Oktalbaum(Octree). Erst bei einem Punkteinhalt < 1 Mio lässt sich in einer PointCollection damit arbeiten. In der linken/oberen Ecke (Abb. 2.10) musste aufgrund einer hohen Punktzahl eine Aufteilung mehr durchgeführt werden. Auf diese Weise lässt sich auch die wahre Summe der Punkte, Abschnitt für Abschnitt addiert, über eine Methode countPoints (Anhang), ermitteln.

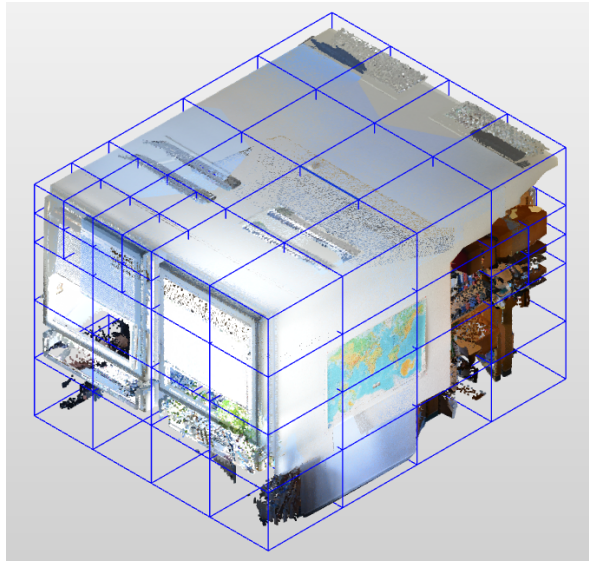


Abb. 2.10: Plugin zur Darstellung der Punktverteilung mit Punktzahl ≤ 900.000 pro Box

Das Anlegen einer Liste der eingeschränkten PointCollections gewährt, durch eine Programm-Schleife den Durchlauf der einzelnen Punkte, einen Zugriff auf die Eigenschaften eines jeden Punktes. Außerdem kann direkt auf spezielle kleinere Bereiche der Punktwolke über den Listenindex zugegriffen werden.

2.2.4 Heat-Map

Über Projektion/Fallen der Punkte auf die Wand berechnet man die Distanz und markiert die Wand an dieser Stelle in einer Farbe des Distanzwertes. Die Wand sollte wieder grob/genau je nach Punktzahl mitsamt Fenstern (Abb. 2.11) abgebildet werden. Deformationen und Fehler bei Scans sollten abhängig von Höhe der Scan-Genauigkeit und dem LOD erkennbar sein.



Abb. 2.11: Grobe/kleine Fenster-Heatmap

2.2.5 Vergleich

Vorhandene Eigenschaften stehen für weitere Ermittlungen durch Berechnungen zur Verfügung. Mit den Werten kann ein Bauplan-Vergleich durchgeführt werden. Nach grober Ausrichtung zwischen erzeugtem Programmabbild/Ebenen und Punktwolken-Scan besteht bestmögliche/wahrscheinliche Parallelität.

3. AUSBLICK UND ZUKUNFT

Letztendlich müssen Messungen nicht mehr von Hand (per Zollstock) durchgeführt werden, vor allem bei schwierigem Zugriff auf die Außenseite eines Gebäudes. Ein genauer Punktwolken-Scan mit seinen Abrundungen(Filterung, Sortierung) übergibt auf Abfrage eine Menge an Werten. Aus diesen Messungen können schnell Berechnungen für Unbekannte über das System ausgeführt werden.

An einem Scan können ebenso, beliebig viele Änderungen vorgenommen werden, ohne Einfluss auf die Realität und schnellere/einfachere Bearbeitung, da diese maschinell und virtuell verlaufen.

Es besteht eine Ausarbeitung der virtuellen Darstellung, um für höhere Genauigkeit zu sorgen und immer näher an die Realität zu rücken.

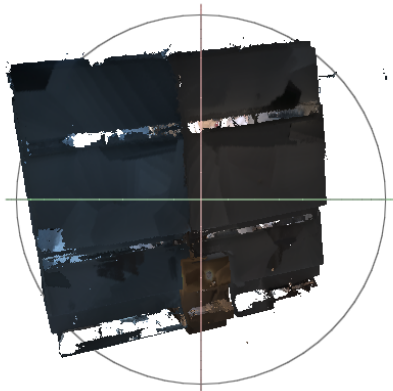
Man giert wahrhaftig danach, die Erstellung des Computer Aided Design (CAD) (siehe Kap.2) aus einer übergebenen Punktwolke zu automatisieren, um von handlicher Arbeit wegzutreten.

3.1 Eigene Plugins

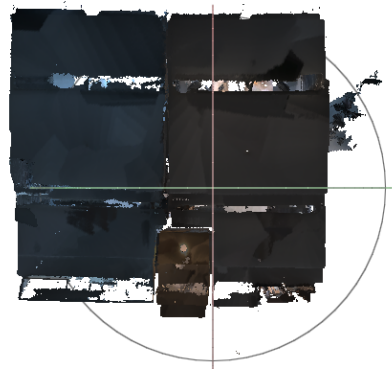
Das Schreiben und anschließende Einfügen der Erweiterungen lässt einem freie Möglichkeiten, eigene Vorhaben zu planen und durchzuführen.

Wie bereits erwähnt, ist keine automatische Ausrichtung der Punktwolke in Revit vorhanden. Dies würde ich von daher gerne als Thema für meine Bachelorarbeit nutzen.

Die BoundingBox-Collection verschafft einen Zugriff über alle Punkte. Eine Heat-Map zeigt die Punkteverteilung. Mit einer Art *Übereinstimmungs-Algorithmus* und minimalen Verschiebungen/Drehungen lässt sich möglicherweise eine automatisierte Punktwolken-Ausrichtung bestimmen.



(a) 3D-Punktwolke unausgerichtet



(b) 3D-Punktwolke ausgerichtet

Abb. 3.1: Punktwolken-Ausrichtung

ANHANG

```
//Punkte der Punktwolke pci in Box min bis max
public static long countPoints(PointCloudInstance pci, XYZ min, XYZ max)
{
    double rund = 0.001;

    List<Plane> planes = new List<Plane>();
    planes.Add(Plane.CreateByNormalAndOrigin(XYZ.BasisX, min));
    planes.Add(Plane.CreateByNormalAndOrigin(-XYZ.BasisX, max));
    planes.Add(Plane.CreateByNormalAndOrigin(XYZ.BasisY, min));
    planes.Add(Plane.CreateByNormalAndOrigin(-XYZ.BasisY, max));
    planes.Add(Plane.CreateByNormalAndOrigin(XYZ.BasisZ, min));
    planes.Add(Plane.CreateByNormalAndOrigin(-XYZ.BasisZ, max));
    PointCloudFilter filter = PointCloudFilterFactory.CreateMultiPlaneFilter(planes);

    long ps = 0;
    ps = pci.GetPoints(filter, 0.0001, 999999).Count();
    if (ps <= 900000)
    {
        return ps;
    }

    //Box in 8 neue Boxen gesplittet da mehr als 900000 enthalten sind
    XYZ mid = new XYZ((max.X + min.X) / 2, (max.Y + min.Y) / 2, (max.Z + min.Z) / 2);
    return (countPoints(pci, min, (min + max) / 2)
        + countPoints(pci, new XYZ(min.X, mid.Y + rund, min.Z), new XYZ(mid.X + rund, max.Y, mid.Z + rund))
        + countPoints(pci, new XYZ(mid.X + rund, min.Y, min.Z), new XYZ(max.X, mid.Y + rund, mid.Z + rund))
        + countPoints(pci, new XYZ(mid.X + rund, mid.Y + rund, min.Z), new XYZ(max.X, max.Y, mid.Z + rund))
        + countPoints(pci, new XYZ(min.X, min.Y, mid.Z + rund), new XYZ(mid.X + rund, mid.Y + rund, max.Z))
        + countPoints(pci, new XYZ(mid.X + rund, min.Y, mid.Z + rund), new XYZ(max.X, mid.Y + rund, max.Z))
        + countPoints(pci, new XYZ(min.X, mid.Y + rund, mid.Z + rund), new XYZ(mid.X + rund, max.Y, max.Z))
        + countPoints(pci, new XYZ(mid.X + rund, mid.Y + rund, mid.Z + rund), max));
}
```

Abb. 2: C# - Plugin unter Zusatzmodule/RevitBauplanAddin zur Ausgabe der Punktzahl

3.1. EIGENE PLUGINS

```
//einfache Linie
public static void Create3DModelLine(XYZ p, XYZ q)
{
    //Document doc = pUiapp.ActiveUIDocument.Document;

    using (Transaction tr = new Transaction(Main.m_doc, "Create3DModelLine"))
    {
        tr.Start();
        try
        {
            if (p.IsAlmostEqualTo(q))
            {
                throw new System.ArgumentException("Expected two different points.");
            }
            Line line = Line.CreateBound(p, q);
            if (line == null)
            {
                throw new Exception("Geometry line creation failed.");
            }
            ModelCurve mCurve = Main.m_doc.Create.NewModelCurve(line, NewSketchPlanePassLine(line));

            tr.Commit();
        }
        catch (ExternalApplicationException ex)
        {
            TaskDialog.Show("Titel", ex.Source + Environment.NewLine + ex.StackTrace + Environment.NewLine + ex.Message);
            // tr.Rollback();
        }
    }
}
```

Abb. 3: C# - Code zur Erstellung einer Linie im 3D-System

ABKÜRZUNGEN

ASCII	American Standard Code for Information Interchange
BIM	Building Information Modelling
CAD	Computer Aided Design
GIA	Geodätische Institut und Lehrstuhl für Bauinformatik & Geoinformationssysteme
ICP	Iterative Closest Points
IFC	Industry Foundation Classes
LOD	Level of Detail
MLS	Mobiles Laserscanning
MMS	Mobile-Mapping-System
SFM	Structure from Motion
SLAM	Simultaneous Localization and Mapping
STL	Standard Triangulation Language
TLS	Terrestrisches Laserscanning
TOF	Time-of-Flight
XML	Extensible Markup Language

LITERATURVERZEICHNIS

- [1] Andreas Breitbarth. *Entwicklung von Methoden zur optischen 3D-Vermessung in Bewegung*. Jena, friedrich-schiller-universität jena, diss., 2015, Thüringer Universitäts- und Landesbibliothek Jena, Jena, 2015.
- [2] Udo Jankowski. Direktkonvertierung von punktwolken aus scan-daten in finite element modelle: Abschlussbericht ; laufzeit des vorhabens: 01.11.08 bis 31.10.10, 2011.
- [3] Wilhelm Wilke. *Segmentierung und Approximation großer Punktwolken*. Darmstadt, techn. univ., diss., 2000.
- [4] Trimble Inc. Bim-workflow in der gebäudetechnik. <https://constructible.trimble.de/blogs/bim-workflow-in-der-geb%C3%A4udetechnik/>. Accessed: 2019-12-03.
- [5] Build Informed GmbH. Lods - der fertigstellungsgrad | build informed. <https://www.buildinformed.com/fertigstellungsgrad-lod/>. Accessed: 2019-11-21.