

## TODOs

1: Glossar .....	5
2: Bild? .....	5
3: Bild? .....	5
4: Akronyme .....	6
5: Mehr Überblick .....	7
6: Voxel statt Würfel verwenden? .....	8
7: Ablauf als Bild .....	8
8: Background Option für weißen Hintergrund für Bilder .....	9
9: Mehr Baumeigenschaften .....	10
10: Baumeigenschaften + ? → Segmente .....	10
11: Segmente + Eigenschaften + ? → Klassifizierung? .....	10
12: Meshing .....	11
13: Oben/Unten Teilung in 2 Segmente für Debug .....	11
14: Orthogonal? .....	13
15: Bedienung/Interface .....	13
16: Referenzen .....	13

# Masterarbeit

**Berechnung charakteristischen Eigenschaften  
von botanischen Bäumen mithilfe von 3D-Punkt-  
wolken.**

**Name:** Anton Wetzel

E-Mail: anton.wetzel@tu-ilmenau.de

Matrikelnummer: 60451

Studiengang: Informatik Master

**Betreuer:** Tristan Nauber

E-Mail: tristan.nauber@tu-ilmenau.de

**Professor:** Prof. Dr.-Ing. Patrick Mäder

E-Mail: patrick.maeder@tu-ilmenau.de

Fachgebiet: Data-intensive Systems and Visualizati-  
on Group

Datum: 21.11.2023



**TECHNISCHE UNIVERSITÄT  
ILMENAU**

# Inhaltsverzeichnis

1. Glossar .....	5
1.1. Punktwolken .....	5
1.2. Scanner .....	5
1.3. Datenstrukturen .....	5
1.4. Akronyme .....	5

## I. Überblick

1. Punktwolke .....	7
2. Daten .....	7
3. Stand der Technik .....	7

## II. Berechnung

1. Ablauf .....	8
2. Separierung in Bäume .....	8
2.1. Diskretisieren .....	8
2.2. Segmente bestimmen .....	8
2.3. Segmentieren .....	8
3. Baumeigenschaften .....	9
3.1. Krümmung .....	9
3.2. Punkthöhe .....	9
3.3. Varianz in Scheibe .....	9
4. Segmentierung von einem Baum .....	10
5. Eigenschaften für Visualisierung .....	10
5.1. Normale .....	10
5.2. Punktgröße .....	10
6. Baumart .....	10

## III. Meshing

## IV. Visualisierung

1. Technik .....	11
2. Punkt .....	11
3. Dynamische Eigenschaft .....	11
4. Subpunktwolken (Bäume) .....	11
4.1. Selektion (Raycast) .....	11
4.2. Anzeige .....	12
5. Eye Dome .....	12
6. LOD Octree .....	12
6.1. Detailstufe .....	12
6.2. Kostenbudget? .....	12
7. Kamera/Projektion .....	13
7.1. Kontroller .....	13
7.1.1. Orbital .....	13

7.1.2. First person .....	13
7.2. Projektion .....	13
7.2.1. Perspektive .....	13
7.2.2. Orthogonal? .....	13
8. Bedienung/Interface .....	13
Bibliographie .....	14

# 1. Glossar

Todo: Glossar

## 1.1. Punktwolken

**Koordinatensystem** ist eine Menge von Achsen, mit den eine Position genau beschrieben werden kann. Im Normalfall werden kartesische Koordinaten verwendet, welche so orientiert sind, dass die x-Achse nach rechts, die y-Achse nach oben und die z-Achse nach hinten zeigt.

**Punkt** ist eine dreidimensionale Position, welcher zusätzlichen Informationen zugeordnet werden können.

**Punktwolke** ist eine Menge von Punkten. Für alle Punkte sind die gleichen zusätzlichen Informationen vorhanden.

**Normale** ist ein normalisierter dreidimensionaler Vektor, welcher die Orientierung einer Oberfläche von einem Objekt angibt. Der Vektor ist dabei orthogonal zur Oberfläche, kann aber in das Objekt oder heraus gerichtet sein.

## 1.2. Scanner

Arial ...

Terrestrial ...

... ...

## 1.3. Datenstrukturen

**Tree** ist eine Datenstruktur bestehend aus Knoten, welche wiederum Kinderknoten haben können. Die Knoten selber können weitere Informationen enthalten.

**Octree** ist ein Tree, bei dem ein Knoten acht Kinderknoten haben kann. Mit einem Octree kann ein dreidimensionaler Würfel aufgeteilt werden. Jeder Knoten gehört zu einem Würfel, welcher gleichmäßig mit den Kinderknoten weiter unterteilt wird.

Todo: Bild?

**Quadtree** ist ein Tree, bei dem ein Knoten vier Kinderknoten haben kann. Statt eines dreidimensionalen Würfels bei einem Octree, kann ein Quadtree ein zweidimensionales Quadrat unterteilen.

Todo: Bild?

**Leaf** ist ein Knoten, welcher keine weiteren Kinderknoten hat. Für Punktwolken gehört jeder Punkt zu genau einem Leaf.

**Branch** ist ein Knoten, welcher weitere Kinderknoten hat.

**Root** ist der erste Knoten im Tree, alle anderen Knoten sind direkte oder indirekte Kinderknoten vom Root.

## 1.4. Akronyme

Todo: Akronym

# I. Überblick

## 1. Punktwolke

- Menge von Punkten
- mindestens Position

## 2. Daten

- Waldstücke
- Deutschland
- terrestrial und arial
- zusätzlich manuelle Datenbestimmung
- nur Position bekannt

Todo: Mehr Überblick

## 3. Stand der Technik

Punktwolken können mit unterschiedlichen Lidar Scanverfahren aufgenommen werden. Aufnahmen vom Boden oder aus der Luft bieten dabei verschiedene Vor- und Nachteile [1]. Bei einem Scan von Boden aus, kann nur eine kleinere Fläche abgetastet werden, dafür mit erhöhter Genauigkeit, um einzelne Bäume genau zu analysieren [2]. Aus der Luft können größere Flächen erfasst werden, wodurch Waldstücke aufgenommen werden können, aber die Datenmenge pro Baum ist geringer [3].

Nach der Datenerfassung können relevante Informationen aus den Punkten bestimmt werden, dazu gehört eine Segmentierung in einzelne Bäume [4] und die Berechnung von Baumhöhe oder Kronenhöhe [3].

Ein häufiges Format für Lidar-Daten ist das LAS Dateiformat [5]. Bei diesem werden die Messpunkte mit den bekannten Punkteigenschaften gespeichert. Je nach Messtechnologie können unterschiedliche Daten bei unterschiedlichen Punktwolken bekannt sein, aber die Position der Punkte ist immer gegeben. Aufgrund der großen Datenmengen werden LAS Dateien häufig im komprimierten LASzip Format [6] gespeichert. Die Kompression ist Verlustfrei und ermöglicht eine Kompressionsrate zwischen 5 und 15 je nach Eingabedaten.

*LASTools* [7] ist eine Sammlung von Software für die allgemeine Verarbeitung von LAS Dateien. Dazu gehört die Umwandlung in andere Dateiformate, Analyse der Daten und Visualisierung der Punkte. Durch den allgemeinen Fokus ist die Software nicht für die Verarbeitung von Waldteilen ausgelegt, wodurch Funktionalitäten wie Berechnungen von Baumeigenschaften mit zugehöriger Visualisierung nicht gegeben sind.

## II. Berechnung

Todo: Voxel statt Würfel verwenden?

### 1. Ablauf

Todo: Ablauf als Bild

1. Diskretisieren
  - in 5cm große Würfel unterteilen
  - kann vollständig im Hauptspeicher sein
2. Segmente bestimmen
  - Top Down
  - Quadtree
  - nearest mit ...m max distance
3. Segmentieren (nochmal)
  - je nach Würfel zum Segment ordnen
4. Segmente analysieren
  - Beimeigenschaften
  - Punktgigenschaften
5. Segmente abspeichern
6. gemeinsamer Octree erstellen
  - LOD für Visualisierung

## 2. Separierung in Bäume

### 2.1. Diskretisieren

1. Punkte laden
2. Position diskretisieren (5cm) und zugehöriger Würfel berechnen
3. alle Würfel, welche Punkte enthalten abspeichern

### 2.2. Segmente bestimmen

1. Boden Bestimmen
  - tiefster Würfel in ...m × ...m
2. von oben nach unten in Scheiben segmentieren
  - für jeden Würfel den nächsten bereits Segmentieren Würfel mit Maximaldistanz ...m bestimmen
  - wenn kein Würfel gefunden, neues Segment anfangen
  - wenn Würfel gefunden gleiches Segment verwenden

### 2.3. Segmentieren

1. Punkt nochmal laden
2. Berechnen zu welchen Würfel der Punkt gehört
3. Segment vom Würfel zum Punkt zuordnen



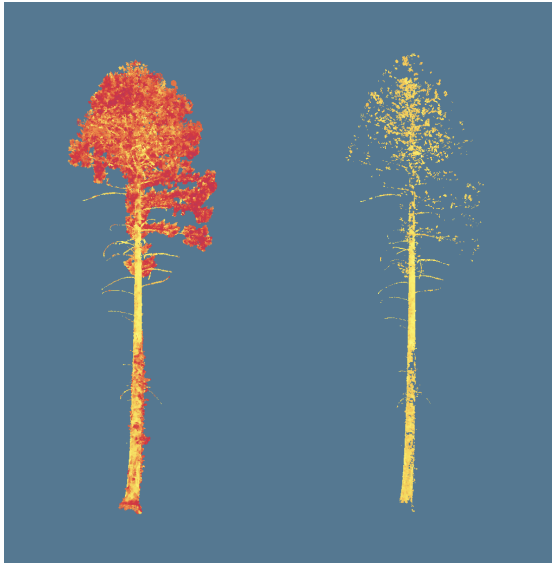
4. Punkte im gleichem Segment zusammenfassten zu einer Punktwolke

### 3. Baumeigenschaften

Alle Punkte in einem Segment (Baum) sind verfügbar

#### 3.1. Krümmung

1. Hauptkomponentenanalyse
  - $\lambda_i$  mit  $i \in \mathbb{N}_0^2$  und  $\lambda_i > \lambda_j$  wenn  $i > j$
2.  $c = \frac{3\lambda_2}{\lambda_0 + \lambda_1 + \lambda_2}$ 
  - $c \in [0, 1]$



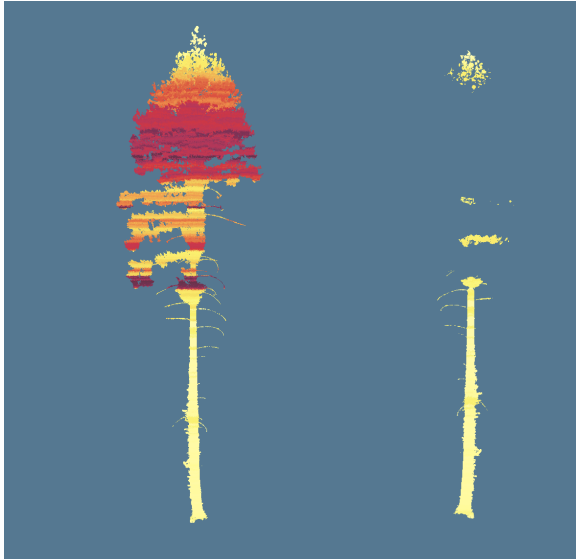
Todo: Background Option für weißen Hintergrund für Bilder

#### 3.2. Punkthöhe

1.  $h = \frac{p_y - y_{\min}}{y_{\max} - y_{\min}}$ 
  - $h \in [0, 1]$

#### 3.3. Varianz in Scheibe

1. 5 cm Scheiben
2. geometrischen Schwerpunkt berechnen
3. Varianz  $v$  berechnen
4.  $x = \frac{v_i}{v_{\max}}$ 
  - $x \in [0, 1]$



Todo: Mehr Baumeigenschaften

## 4. Segmentierung von einem Baum

Todo: Baumeigenschaften + ? → Segmente

## 5. Eigenschaften für Visualisierung

### 5.1. Normale

1. Hauptkomponentenanalyse
2. Eigenvektor für  $\lambda_2$

### 5.2. Punktgröße

1. Durchschnittliche Abstand zu umliegenden Punkten
2. Ausgleichsfaktor?

## 6. Baumart

Todo: Segmente + Eigenschaften + ? → Klassifizierung?

- out of scope?
- neural?

## III. Meshing

Todo: Meshing

## IV. Visualisierung

### 1. Technik

- Rust
- WebGPU (wgpu)
- native Window (website?)
- LAS/LAZ

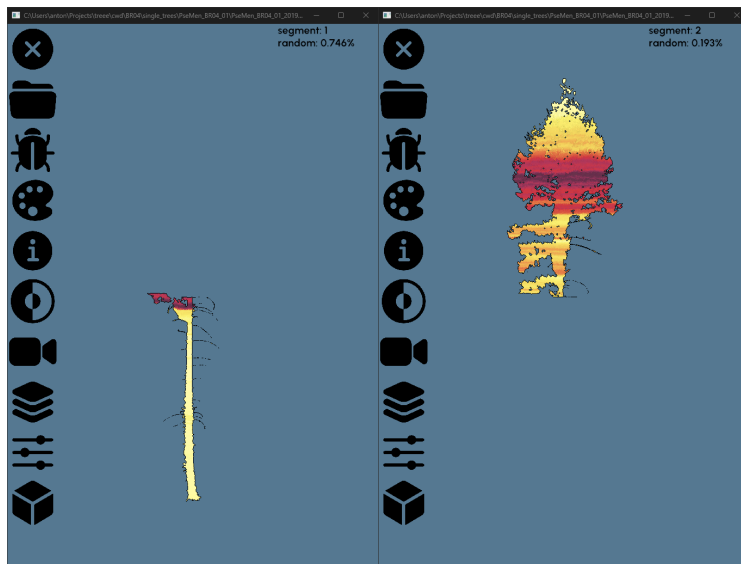
### 2. Punkt

- Instancing
- quad rect
- Ausdehnung mit Normale
- Discard mit Distanz für Kreis (Kreisfläche)

### 3. Dynamische Eigenschaft

- eigenschaften als 32 bit unsigned integer
- look up table für farbe basierend auf eigenschaftswert

### 4. Subpunktvolken (Bäume)



Note: Oben/Unten Teilung in 2 Segmente für Debug

#### 4.1. Selektion (Raycast)

- von root bis leaf
- bestimme intersection mit knoten

- wenn leaf mit intersection gefunden
  - lade Segmente, welche im Leaf liegen
  - bestimme Abstand Ray-Punkt
  - wenn kleiner als Radius hit
  - Segment mit Punkt mit geringstem Abstand Ergebnis

## 4.2. Anzeige

- segmente separate abgespeichert
  - keine LOD Stufen, nur Originalpunkte
  - maxbuffer size?

## 5. Eye Dome

1. Post processing
2. depth image
3. anliegender Pixel mit maximalem Abstand
  1.  $(-1, 0), (0, -1), (1, 0), (0, 1)$
4. Parameter  $m$
5.  $x = \frac{\text{maximaler abstand}}{m}$
6. auf  $[0, 1]$  beschränken
7. Parameter color?
8. Pixel mit color und  $x$  als  $\alpha$  überlagern

## 6. LOD Octree

1. (Octree begriffe in English)
2. Octree mit maximaler Blattgröße  $1 \ll 15?$  (32k)
3. Blätter mit mehr Punkten werden in 8 Kinderknoten geteilt
  - Punkte auf Kinder verteilen
4. non Leaf Knoten wird LOD aus Kindern berechnet
  1. Punkte kombinieren
  2. Für Eigenschaften wert von einem Punkt übernehmen
5. rekursiv von Kindern bis zum Root
6. beim rendern für entferne Punkte nur Lod Stufe verwenden
  1. je näher so genauere LOD Stufe

### 6.1. Detailstufe

1. Grid
  - Größe abhängig von Leafgröße, wird gröber für größere Blätter
2. Kombination von Punkten
  - Größe als Fläche addieren
  - Normale Durchschnitt
  - Position durchschnitt
  - Eigenschaften?

### 6.2. Kostenbudget?

- Anpassung der Genauigkeit
  - Verringerung des Aufwands
- Iteratives anpassen an das Budget?

## 7. Kamera/Projektion

### 7.1. Kontroller

- bewegt Kamera
- kann gewechselt werden, ohne die Kameraposition zu ändern

#### 7.1.1. Orbital

- rotieren um einem Punkt im Raum
- Kamera fokussiert zum Punkt
- Entfernung der Kamera zum Punkt variabel
- Punkt entlang der horizontalen Ebene bewegbar
- To-do: Oben-Unten Bewegung

#### 7.1.2. First person

- rotieren um die Kamera Position
- Bewegung zur momentanen Blickrichtung
- Bewegungsgeschwindigkeit variabel
- To-do: Oben-Unten Bewegung

## 7.2. Projektion

### 7.2.1. Perspektive

- Projektion mit Field of View Kegel

### 7.2.2. Orthogonal?

Todo: Orthogonal?

## 8. Bedienung/Interface

Todo: Bedienung/Interface

Todo: Referenzen

## Bibliographie

- [1] J. J. Donager, A. J. Sánchez Meador, und R. C. Blackburn, „Adjudicating perspectives on forest structure: how do airborne, terrestrial, and mobile lidar-derived estimates compare?“, *Remote Sensing*, Nr. 12, S. 2297, 2021.
- [2] M. Disney, „Terrestrial LiDAR: a three-dimensional revolution in how we look at trees“, *New Phytologist*, Nr. 4, S. 1736–1741, 2019, doi: <https://doi.org/10.1111/nph.15517>.
- [3] T. Suzuki, S. Shiozawa, A. Yamaba, und Y. Amano, „Forest Data Collection by UAV Lidar-Based 3D Mapping: Segmentation of Individual Tree Information from 3D Point Clouds“, *International Journal of Automation Technology*, S. 313–323, 2021, doi: [10.20965/ijat.2021.p0313](https://doi.org/10.20965/ijat.2021.p0313).
- [4] A. Burt, M. Disney, und K. Calders, „Extracting individual trees from lidar point clouds using treeseg“, *Methods in Ecology and Evolution*, Nr. 3, S. 438–445, 2019, doi: <https://doi.org/10.1111/2041-210X.13121>.
- [5] L. Graham, „The LAS 1.4 specification“, *Photogrammetric engineering and remote sensing*, Nr. 2, S. 93–102, 2012.
- [6] M. Isenburg, „LASzip: lossless compression of LiDAR data“, *Photogrammetric engineering and remote sensing*, Nr. 2, S. 209–217, 2013.
- [7] C. Hug, P. Krzystek, und W. Fuchs, „Advanced Lidar Data Processing with Lastools“, 2004. Verfügbar unter: <https://api.semanticscholar.org/CorpusID:14167994>