Hochschule für Angewandte Wissenschaften Hamburg
Hamburg University of Applied Sciences

Bachelorarbeit

Joschka Schulz

Entwicklung eines 3D Scanners basierend auf einem Laser-Distanzsensor

Fakultät Technik und Informatik
Studiendepartment Informatik

Faculty of Engineering and Computer Science
Department of Computer Science

Joschka Schulz

Entwicklung eines 3D Scanners basierend auf einem Laser-Distanzsensor

Bachelorarbeit eingereicht im Rahmen der Bachelorprüfung

im Studiengang Bachelor of Science Angewandte Informatik
am Department Informatik
der Fakultät Technik und Informatik
der Hochschule für Angewandte Wissenschaften Hamburg

Betreuender Prüfer: Prof. Dr. Philipp Jenke
Zweitgutachter: Prof. Dr. Ulrike Steffens

Eingereicht am: 21. April 2015

Joschka Schulz

Thema der Arbeit
Entwicklung eines 3D Scanners basierend auf einem Laser-Distanzsensor

Stichworte
3D Scanner, Laser-Distanzsensor, TinkerForge

Kurzzusammenfassung
Dieses Dokument handelt von einem Konzept und Entwurf zur Umsetzung eines 3D Scanners anhand eines Laser-Distanzsensor. Der Scanner wird mittels TinkerForge Bausteinen gesteuert die über eine Javaanwendung gesteuert werden können.
Joschka Schulz

Title of the paper
Developing a 3D scanner based on a laser distance sensor

Keywords
3D Scanner, laser distance sensor, TinkerForge

Abstract
This document is about a concept and design of an implemention of an 3D scanner working with an laser sensor. The scanner is controlled by an TinkerForge Stack which is controlled by an Java application.

Inhaltsverzeichnis

1. Einleitung

Durch die Entwicklung eines 3D-Scanner mit Laser-Distanzsensor wird die Möglichkeit geschaffen eine kombination aus Hard- und Software zu erstellen, die Forschung und weitere Arbeiten verwendet können. Die Möglichkeit reale Objekte als Punktwolke aufnehmen zu können, und diese dann am Computer bearbeitbar oder untersuchbar zu machen, bietet eine Menge Spielraum für Anwendungsgebiete. So können beispielsweise bei Ärzten Körper von Patienten mit 3D-Scannern aufgenommen werden, um anschließend eine maßgefertigte Stütze für Orthopädie und Prothetik anzufertigen (3D, 2015).

1.1. Problemstellung

Es soll ein physikalisches Objekt in eine Punktwolke überführt werden, um diese dann digital darzustellen oder diese zu bearbeiten. Dies soll durch einen selbstentwickelten 3D-Scanner geschehen. Des weiteren soll der 3D Scanner in der Lage sein, sich möglichst selbstständig zu justieren, und leichte Scanfehler durch Algorithmen in der Software korrigieren. Der dabei entstehende Scanner soll keine schon fertig implementierte Software benutzten und es soll speziell für diesen Scanner eine neue Software geschrieben werden, die in der Lage ist, die Kommunikation der Hardware als auch die Fehlerkorrekturen ausführen zu können.

1.2. Ziel der Arbeit

Ziel der Arbeit ist es, durch den Scanner und der eigens implementierten Software ein Objekt aufzunehmen. Exemplarisch wird in diesem Fall eine Gummiente eingescannt, um diese als Punktwolke abspeichern zu können. Dabei gilt es einen Algorithmus zu implementieren, der dafür sorgt Ungleichmäßigkeiten beim Scannen zu korrigieren, und das Objekt in seiner richtigen Form abzubilden. Weiter ist es auch Ziel der Arbeit die Grundkonfiguration und Justierung des Scanners zu erstellen, um diesen benutzbar für weitere Bachelorarbeiten zu machen.

1.3. Ähnliche Lösungsansätze

Da die Entwicklung eines 3D-Scanners nicht unbedingt ein neues Thema ist, kann man sich auf dem aktuellen Markt umschauen und wird verschiedene andere Lösungsansätze finden. Die Lösungsansätze sind klar in zwei sehr große Kategorien zu unterteilen. Die eine Kategorie sind die optischen Scanner und die andere Kategorie sind die physischen 3D Scanner.

1.3.1. Optische 3D Scanner

Unter den optischen 3D Scannern werden jene Scanner eingeordnet, welche das Objekt nicht berühren müssen, um ein Abbild von diesem zu machen. Das bedeutet grade sehr zerbrechliche Objekte, wie zum Beispiel Ausgrabungen in der Archäologie, können aufgenommen werden ohne dass das Objekt berührt werden muss und dadurch Schaden nimmt.

Probleme bei dieser Art von Scannern gibt es, wenn es darum geht, reflektierende Oberflächen, wie zum Beispiel poliertes Metall, aufzunehmen. Das Licht wird an dieser Stelle reflektiert und der Sensor oder das Gerät, welches das Licht wieder aufnimmt, ist nicht in der Lage, die Entfernung richtig aufzunehmen.

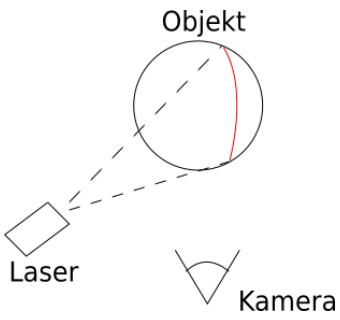

Abbildung 1.1.: Laserlinien Verfahren

Im Bereich der optischen Scanner gibt es viele Herangehensweisen. Ein Verfahren in diesem Gebiet wäre die Projektion einer Line mittels eines Lasers auf dem Objekt. Diese Linie wird anschließend durch eine Kamera aufgenommen und über eine Software aus dem Bild herausgerechnet. Durch die bekannte Rotation des Objektes ist es nun möglich, die Silhouette aus dem Objekt zu rechnen und so ein drei Dimensionales Abbild zu erzeugen, zu sehen auf Abbildung 1.1.

Ein ähnliches Verfahren zu dem, bei dem der Laser eine Linie auf das Objekt projeziert, ist das Projezieren eines ganzen Lichtgitters. Durch dieses auf das Objekt geworfene Muster ist

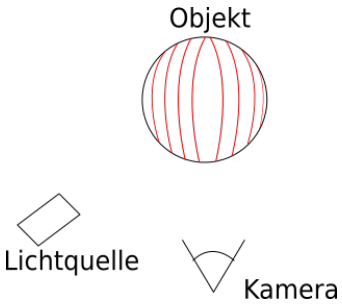

Abbildung 1.2.: Strukturiertes Licht Verfahren

es auch in diesem Verfahren mögich aus der Ausleuchtung des Objektes ein 3D Modell zu erzeugen, zu sehen auf Abbildung 1.2.

Ein klarer Vorteil in diesen beiden Verfahren ist, dass das Bild, das ohnehin gemacht wird, um das Objekt zu berechnen, im späteren Verlauf auch benutzt werden kann, um Farbwerte an den Objekten zu bestimmen. So können für die einzelnen Punkte in der Punktwolke Farben festgelegt werden, um bei der späteren Abbildung das Objekt möglichst Farbecht wieder zu geben. Verwendet wird das strukturierte Licht Verfahren bei den Scannern von David (DAVID, 2015).

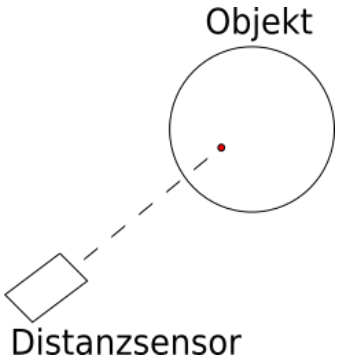

Abbildung 1.3.: Punktuelle Aufnahme

Der in dieser Arbeit entfworfene 3D Scanner gehört zum Bereich der optischen 3D-Scanner, da er mithilfe seines Laser-Distanzsensors die Punkte vom Objekt entgegen nimmt. Durch dieses Verfahren kann zwar erst einmal nicht die Farbe bestimmt werden, aber dafür sehr genaue Muster die beispielsweise in das Objekt graviert sind oder hervorstehen. Bei diesem Verfahren werden die Punkte einzeln nach und nach aufgenommen, wie es in Abbildung 1.3 zu sehen ist.

1.3.2. Taktile 3D Scanner

Unter den taktilen Messverfahren versteht man Scanner, welche zum Beispiel einen messenden Taster, die Oberfläche von Objekten abtasten. Dabei kann unterschieden werden zwischen den messenden Tastern, die an der Oberfläche entlang geführt werden und den schaltenden Messtastern, die nach jeder Messung einmal abgesetzt werden müssen (xpertgate GmbH & Co. KG, 2015). Mithilfe eines Koordinatenmessgerät werden ausgewählte Punkte an einem Objekt abgetastet und die Koordinaten in x-, y- und z-Werte digital aufgenommen. Durch dieses Verfahren ist es möglich auch an Stellen zu Messen, welche für das optische Messverfahren schwer erreichbar sind oder, wenn es nötig ist, sehr genaue Messungen vorzunehmen.

Da in diesem Verfahren die Messgenauigkeit besser ist als die des optischen Verfahrens, ist es für die Industrie in einigen Bereichen noch unerlässlich, bestimmte Fertigungsteile oder aber das Werkzeug zum Produzieren dieser Teile, auf die taktile Messung zu verzichten und Optische Messungen einzusetzten (Vierling, 2013, S.13-14).

1.4. Motivation

Die Motivation dieser Arbeit liegt darin durch den Bau eines 3D-Scanners anhand von einem Laser-Distanzsensor Objekte welche auf den Scanner gelegt werden, durch eine in dieser Bachelorarbeit entwickelten Software, mittels einer Punktwolke aufzunehmen. Diese aufgenommene Punktwolke kann für spätere Anschauungen wieder hergestellt werden. In diesem entwickelten Scanvorgang können verschieden große Objekte aufgenommen werden, solange diese nicht über den Drehteller herausragen.

1.4.1. Bau eines 3D-Scanners

Da nicht auf einen vorhandenen 3D-Scanner mit Laser-Distanzsensor zurückgegriffen wird, wird innerhalb dieser Arbeit ein eigener Scanner mit einem Laser-Distanzsensor konstruiert. Dieser Scanner wird anhand von einem rotierenden Teller und einem vertikal bewegbaren Distanzsensor umgesetzt, um das Objekt aus jedem Winkel aufnehmen zu können. Durch solch einen Aufbau muss sich der Scanner immer an die zu scannende Stelle bewegen und kann dann einen Punkt für die zu erstellende Punktwolke aufnehmen.

1.4.2. Implementation einer Software für den 3D-Scanner

Da die Hardware bei dieser Bachelorarbeit nur den kleineren Teil darstellt, muss weiterhin noch eine Software implementiert werden, mit der es möglich ist die Hardware anzusprechen. So

wird die Software mittels zwei USB Anschlüssen in der Lage sein, die Positionen des Drehtellers und des Laser-Distanzsensor einzustellen. Durch diese Einstellungsoptionen wird die Software in der Lage sein, kleine Kalibrierungen vorzunehmen, welche durch angebrachte Bauteile entstehen können, die nicht korrekt ausgerichtet wurden.

Des weiteren muss eine Kommunikation geschaffen werden, um mit den Laser-Distanzsensor zu kommunieren. Dieses muss über eine serielle Schnittstelle geschehen und bedarf es diese Kommunikation weitestgehenst zu abstrahieren, damit es auch möglich ist über eine Javaklasse diesen anzusprechen.

1.4.3. Basis für andere Arbeiten und Projekte

Die Hardware und Software kann in zukünftigen Bachelorarbeiten wieder verwendet oder auch erweitert werden. So kann beispielsweise der Scanvorgang, welcher zur Zeit noch keine Löcher erkennen kann, weitgehend erweitert werden, so dass das Objekt aus verschiedenen gescannten Objekten zusammen gesetzt werden kann. So können viele Bachelorarbeiten im Bereich des 3D Scanners von den in dieser Arbeit geschaffenen Grundlagen profitieren.

1.5. Themenabgrenzung

Nicht Ziel dieser Bachelorarbeit ist es, den Scanner dahingehend zu erweitern, dass dieser in der Lage ist die Farbe des zu scannenden Objektes zu erkennen um diese dann auf die Punktwolke zu projizieren. Des weiteren ist es auch nicht Teil der Arbeit Löcher in Objekten zu finden und diese zurück zu rechnen. Es wird lediglich die Grundsteuerung und Kalibrierung der Hardware behandelt.

1.6. Struktur der Arbeit

Die Arbeit ist wie folgt gegliedert. Im Kapitel 2 wird auf die Idee vom 3D Scanner eingegangen. Anschließend wird erklärt, wie der 3D-Scanner für diese Arbeit entworfen ist und was für Bestandteile er hat. Nach dem Kapitel geht es dann zu der Software, die speziell für diesen Scanner entworfen wurde. Abschließend werden die Fehler, die während der Arbeit aufgetaucht und dokumentiert worden sind, genauer erklärt.

2. 3D Scanner

Wie in der Einleitung schon erwähnt, gibt es verschiedene Kategorien von 3D-Scannern. In dieser Bachelorarbeit finden der Entwurf und die Umsetzung eines 3D-Scanner mit Hilfe eines Laser-Distanzsensors statt.

2.1. Die Idee

Die Idee des 3D Scanners besteht darin, es zu ermöglichen, ein Objekt ohne händischen Eingriff von allen Seiten aufzunehmen. Am besten geschieht dies durch einen drehbaren Untergrund und einen in der horizontalen bewegbaren Distanzsensor. Durch diese Kombination ist es möglich, Linie für Linie die Umrisse des Objektes aufzunehmen und so die Oberfläche als Punktwolke darzustellen.

Wichtig bei dieser Idee ist es, dass der Laser-Distanzsensor bei jeder Messung immer nur einen einzelnen Entfernungswert zurückliefert. Der Vorteil von diesem Vorgehen wird die hohe Genauigkeit sein, der Nachteil wird die langsame Abtastung sämtlicher benötigten Punkte sein.

Die Ansteuerung des Scanners wird über einen TinkerForge Stack geschehen. Dies sind kleine Modular erweiterbare Bausteine, die zu einem für die Funktion angepassten Stack zusammen gesetzt werden können. TinkerForge bietet von Haus aus Module für die Kommunikation im eigenen Stack und zur Kommunikation mit Steppermotoren.

2.2. Einschränkungen von 3D Scannern

Durch die Idee des rotierenden Tellers wird es Probleme geben um beispielsweise Löcher, wie in einem Henkel einer Tasse, und die Ober- und Unterseite des Objektes aufzunehmen. Des Weiteren entstehen die Einschränkungen, die alle optischen Scanner besitzen, das bedeutet, dass es unmöglich oder nur schwer möglich sein wird, ein Objekt mit einer spiegelnden Oberfläche aufzunehmen. Dies kann den Laser irritieren und es können keine Messungen vorgenommen werden kann.

Weiterhin kann es dazu kommen, dass bei der Messung an bestimmten Stellen mehr Punkte verfügbar sind als an anderen Stellen. Dies entsteht dadurch, dass durch die Drehscheibe die Kanten des Objektes, welche weiter außerhalb liegen, mehr Entfernung zurücklegen als der Mittelpunkt des Objektes. Somit wird die Auflösung des Objektes je geringer je mehr Volumen diese besitzen.

2.3. Anwendungsgebiete

Durch die Seltenheit von 3D Scanner welche nur über einen einzelnen Distanzsensor verfügen wird sich in dieser Bachelorarbeit auch auf Themengebiete gestützt in denen Laser-Distanzsensoren zur Bildauswertung benutzt werden.

2.3.1. Raumvermessung

Ein ähnliches Verfahren wird auch bei den Aufnehmen von Höhlen und Inneneinrichtungen benutzt. Dabei wird mit einem Laser-Distanzsensor die Entfernung zu den Wänden, dem Bodens und der Decke gemessen. Durch dieses Verfahren lässt sich im späteren Verlauf auswerten, wie die Beschaffenheit des aktuellen Raumes ist.

2.3.2. Industrie

In der Industrie werden 3D Scanner dafür benutzt um getätigte Arbeiten zu kontrollieren. So kann zum Beispiel nachdem ein Objekt geschweißt wurde kontrolliert werden, ob die Schweißnaht ordnungsmäßig angebracht wurde und keine Lücken aufweißt. Auch können Gegenstände auf Verformungen kontrolliert werden. Mit diesen Verfahren ist es der Industrie möglich, Fehler durch Verformungen der Gegenstände oder Optimierungs-Anpassungen heraus zu finden. beispielsweise wird solch ein Scanverfahren bei Crash-Tests durchgeführt (gom, 2015).

2.3.3. Archäologische Messungen

In der Archäologie wird es immer wichtiger Funde digital aufnehmen zu können und so schnell abrufbar oder aber auch teilbar mit anderen Archäologen zu machen. Dies zeigt sich durch Unternehmen wie Explius die speziel für Archäologen einen 3D Scanner entwickelt haben (Explius, 2015).

2.3.4. Aufnahme von Objekten in eine virtuelle Realität

Durch das Aufnehmen von Objekten können diese wieder in virtuellen Realitäten verwendet werden. So könnten beispielsweise realitätsnahe Wohnungen oder andere Räume erstellt werden.

2.3.5. Kopieren von vorhanden Gegenständen

Es wäre denkbar die vom 3D-Scanner aufgenommen Objekte in ein für einen von 3D-Drucker verstandenes Format zu bringen und die eingescannten Objekte anschließend wieder auszudrucken. Durch diesen Vorgang wäre es Möglich eine Kopie von einem 3 Dimensionalen Gegenstand zu erstellen. Dies kann auch dazu verwendet werden 3D Scanner zu evaluieren, indem das spätere Resultat mit dem echten zu vergleichen wird. Als Beispiel kann dies in der Bachelorarbeit von Francis Engelmann an der Rheinisch-Westfälische Technische Hochschule Aachen gesehen werden (Engelmann, 2011, S.45-46).

3. Der Aufbau des 3D Scanner

In diesem Kapitel der Bachelorarbeit wird beschrieben wie genau der Scanner, welcher innerhalb der Bachelorarbeit konstruiert wird, aufgebaut wird. Dabei wird kurz auf die Idee und den Entwurf eingegangen und anschließend werden einmal die einzelnen Komponenten der Hardware erklärt.

3.1. Idee zum Aufbau

Die Idee des Aufbaus des 3D-Scanners ist es, die Basis von einem 3D-Drucker zu benutzten und diesen zweckgemäß umzubauen. An den Schlitten der normalerweise die Drüse für den 3D Druck befestigt, wird der Laser-Distanzsensor befestigt und anstelle der Wärmeplatte für den 3D Druck soll es eine Drehscheibe geben, auf der später die Objekte platziert werden können.

Durch den nach oben und unten bewegbaren Laser-Distanzsensor und die Drehscheibe ist es möglich die Oberfläche des kompletten Objektes in einem 360° Winkel aufzunehmen.

3.2. Entwurf des Scanners

In dem Entwurf der Hardware für den 3D-Scanner wurde entschiedenen, das Gerüst des 3D-Druckers K8200 umfunktionieren, sodass es den Anforderungen des Scanners genügt. Der Hersteller Velleman betont selber bei dem Gerüst, das es gut modifizierbar ist (Velleman, 2015). Das Ursprüngliche Gestell des 3D-Druckers bot viele Funktionen welche ohnehin gebrauchen würden um den Bau des Scanners zu realisieren. So wurde die Wärmplatte durch einen Drehteller ausgetauscht und anstelle einer Drüse für den 3D-Druck wurde ein Laser-Distanzsensor montiert. Dieser war nun durch einen Schlitten hoch und runter fahrbar. Die beiden Servomotoren wurden weiterhin an dem Gestellt beibehalten um den Schlitten und den Drehteller bewegen zu können.

Durch eine kleine Halterung wurden anschließend an dem Gestell eine kleine Halterung befestigt um die Bauteile von TinkerForge anbringen zu können. Diese Bauteile beginnen unten mit dem Masterbrick und zwei Stepperbricks die oben auf die Platine herausgesteckt

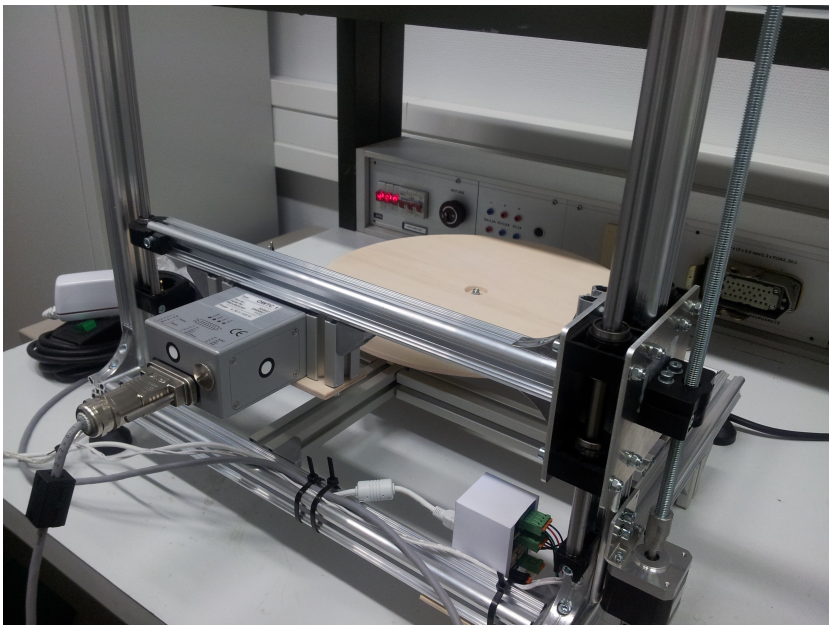

Abbildung 3.1.: Foto vom Aufbau des Scanners

wurden. Durch ein Stromkabel wird der TinkerForge Stack mit ausreichend Strom versorgt um die Platinen und die beiden Steppenmotoren mit versorgen zu können. Die Motoren sind direkt mit den Platinen verbunden und haben keine weitere Verbindung. Auf der Abbildung 3.1 ist ein Foto zu sehen, auf dem der Entwurf im Labor umgesetzt wurde.

3.2.1. Allgmein

Der Scanner sollte einen festen Rahmen mit einem hoch und runter fahrbaren Schlitten besitzen. Außerdem sollte es möglich sein eine Drehscheibe zu montieren, um den zu scannenden Gegenstand rotieren zu können. Das Grundgerüst eines üblichen 3D-Druckers kommt daher gut in Frage. Es bietet schon einen verstellbaren Schlitten und ist auch weiterhin schon mit zwei Servo Motoren bestückt die später verwendet werden können um den Schlitten hoch und runter zu fahren und die Drehscheibe mit dem daraufstehenden Objekt zu bewegen.

Der bewegliche Schlitten sollte dann mit den ausgewählten Laser-Distanzsensor bestückt werden. Damit besteht die Möglichkeit das Objekt in der Vertikalen komplett zu scannen. Die zusätzliche Rotation des Drehtellers ermöglicht es nun, das Objekt von jeder Seite mit den Sensor zu erreichen.

3.2.2. Der Schlitten

Der Schlitten stellt den Teil des 3D-Scanners dar, welcher eingesetzt wird um den Laser-Distanzsensor hoch und herunter zu bewegen. Er sollte durch eine Gewindestange frei in der Höhe positionierbar sein. Damit die spätere Ansteuerung des Scanners besser bestimmbar ist, wird als Motor ein Steppenmotor mit einer geringen Auflösung eingesetzt. Durch die einzelnen Schritte des Motors kann in der späteren Anwendung eine Startposition bestimmt werden und immer wieder zu dieser zurück gekehrt werden. Außerdem lässt sich an den Schritten die aktuelle Höhe des Distanzsensors messen.

3.2.3. Die Drehscheibe

Die Drehscheibe ist ein großer Teller des 3D-Scanners, auf welchen das Objekt platziert wird, dass gescannt werden soll. Dabei ist darauf zu achten, dass das Objekt so mittig wie möglich auf der Scheibe positioniert wird um möglichst genau später wieder diesen Mittelpunkt zu finden. Die Drehscheibe wird durch einen Gummiriemen und einen Schrittmotor gedreht. Dieser Gummiriemen kann durch mehrere Zähne in Halterungen unterhalb der Drehscheibe greifen um so eine möglichst hohe Auflösung zu erhalten.

3.2.4. Der Laser-Distanzsensor

Bei dem Laser-Distanzsensor handelt es sich um ein OWTC 1 von der Firma Welotec, welcher dazu konzipiert ist auch noch in weiteren Entfernungen, bis 150 Meter, sehr genaue Messungen vorzunehmen (Welotec, a). Dabei ist zu beachten, dass durch die hohe Genauigkeit der Scanner nur auf einer bestimmt festgelegten Hertzzahl laufen kann und dadurch den Aufbau in der Geschwindigkeit der Punktaufnahme beschränkt.

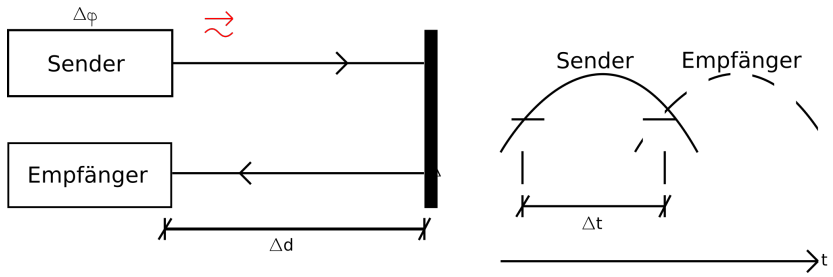

Abbildung 3.2.: Funktionsweise der Phasenverschiebung

Der in dieser Arbeit verwendete Laser-Distanzsensor nutzt das Verfahren der Phasenverschiebung. Bei der Phasenverschiebung wird das vom Objekt reflektierte Licht aufgenommen

und anschließend über die Lichtintensität ermittelt wie weit das Objekt entfernt liegt. Dies wird grafisch in der Abbildung 3.2 gezeigt. Die Phasenverschiebung eignet sich gut für den Gebrauch im 3D Scan, da eine sehr genaue Auflösung auch über mehrere Meter erreicht werden kann. So kann beispielsweise der OWTC 1 auf bis zu 150 Meter im Bereich von 100 micro Meter messen (Vierling, 2013, S.20-21).

3.2.5. Der Sockel

Der Sockel sitzt in der Mitte der Drehscheibe. Er wird dafür benutzt Objekte komplett aufnehmen zu können, da der Laser-Distanzsensor nicht bis an die untere Kante der Drehscheibe gefahren werden kann. Dazu könnte der Sockel noch dazu verwendet werden den Scanner zu kalibrieren, da die Größe des Sockels im Vorfeld schon bekannt ist.

3.2.6. Der TinkerForge Stack

Der TinkerForge Stack ist ein kleiner Turm aus Platinen, die in der vertikalen zusammen gesteckt werden können. Durch den Entwurf des Scanners werden ein Masterbrick (TinkerForge, 2015c), ein Temperatursensor und zwei Stepperbricks (TinkerForge, 2015d) benötigt um alles zu steuern. Der Masterbrick ist dabei die zentrale Stelle an der auch die Verbindung zu den anderen drei Modulen hergestellt wird. Der Temperatursensor wurde lediglich zur Überprüfung der Module angebracht und die beiden Stepper Bricks sind jeweils mit einem Stepper Motor verbunden und können somit die Höhe des Schlittens und den Grad des Drehtellers steuern (TinkerForge, 2015e).

3.3. Einschränkungen im Entwurf

Bei den Einschränkungen im Entwurf ist davon auszugehen das es nicht möglich ist Objekte von oben oder unten zu Scannen ohne sie manuell in die entsprechende Lage zu bringen. Dies bedeutet, dass zum Beispiel bei einer Tasse nicht aufgenommen werden könnte, wie dick der Rand der Tasse ist. Weiterhin gibt es Probleme durch den Bewegungsablauf des Scanners bei beispielsweise eines Henkels an einer Tasse. Es kann nicht festgestellt werden ob dieser geschlossen oder geöffnet ist, da der Scanner lediglich den Rand der Tasse erkennt. Somit ist der Scanner sehr darauf beschränkt Objekte aufzunehmen welche einen geschlossenen Körper haben. In Versuchen dieser Bachelorarbeit wurde deswegen dafür eine Gummiente genommen.

4. Die Software

Da im Rahmen dieser Bachelorarbeit nicht auf eine schon vorhandene Software zurückgegriffen werden soll und für den Aufbau des Scanners eine eigene Ansteuerung implementiert werden soll, wird eine speziell auf die Bauteile zugeschnittene Software entworfen und implementiert. Dabei sollte diese Möglichkeiten bieten um weitere Bachelorarbeiten innerhalb dieses Themas entwerfen und umzusetzen zu können.

4.1. Die Idee

Die Idee der Software ist es, die Möglichkeit durch eine leichte Benutzeroberfläche die Benutzung des Scanners zu gestatten. Die Software sollte sich nahtlos in das Computergrafik Framework integrieren und darüber eine Darstellung für den Benutzer anbieten. Des weiteren muss die Software die Möglichkeit bieten den Scanner zu justieren und gegebenenfalls Abweichungen bei der Messung zu korrigieren.

Weiterhin sollte die Software einen Modularen Aufbau haben. Dies wäre zum Beispiel für den Gebrauch von einem anderen Laser-Distanzsensor von Vorteil. Dieser könnte direkt an den Schlitten angebracht werden und es muss nur gegen ein Interface entwickelt werden, ohne den Rest der Software anzufassen. Auch bietet der modulare Aufbau für andere Bachelorarbeiten einen guten Anschluss um eigene Objekte zu integrieren.

Um die Motoren über die Software anzusteuern bietet der Hersteller der TinkerForge Bricks eine passende Java API (TinkerForge, 2015b). Der Laser-Distanzsensor wiederum muss über eine serielle Schnittstelle angesprochen werden (Welotec, b) und eine Drittanbieter API ist notwendig um dies aus dem Java Quellcode zu erreichen.

4.2. Vorgehensweise

In diesem Abschnitt wird Schritt für Schritt der Weg erklärt, der gegangen wurde um die Software für den Scanner zu entwickeln.

4.2.1. Überblick über TinkerForge gewinnen

Zum Start der Entwicklung der Software muss sich erst einmal ein Überblick über die Bausteine von TinkerForge verschafft werden. Die Kommunikation erfolgt dabei über den Masterbrick auf welchen die anderen Bausteine gesteckt sind. Jeder Baustein hat eine eindeutige Brick ID und kann anschließend mit dieser ID über den Masterbrick angesprochen werden. In dem Fall dieser Bachelorarbeit werden so in der Software ein Masterbrick und zwei Steppermotorbricks registriert.

Die Brick ID kann dann innerhalb der von TinkerForge zur Verfügung gestellten API nun verwendet werden um den richtigen und vor allem immer den gleichen Stepperbrick für eine Aufgabe zu bestimmen. So kann es nicht vorkommen das nach einem Neustart des kompletten Systems, der falsche Stepperbrick für die Aufgabe angesprochen wird.

4.2.2. Serielle Kommunikation abbilden

Die Schnittstelle zum Laser-Distanzsensor ist nur über serielle Kommunikation möglich. Damit diese noch auf neuen Systemen ohne seriellen Port funktioniert, wurde ein USB zu seriellen Port Kabel zwischen den System und den 3D-Scanner zu gehängt. Die Software benutzt anschließend die Bibliothek jssc um einfache Strings an die Serielle Schnittstelle zu geben (JSSC, 2015). Nun mussten durch die Dokumentation des OWTC1 Lasers die passende Befehle für die Kommunikation herausgesucht werden und diese durch leichtere Methodenaufrufe abstrahiert werden, da die Form der Aufrufe nicht konsistent in einer gleichen Folge existiert.

4.2.3. Kalibrieren des Sensors

Nachdem die ersten Teile der Ansteuerungssoftware fertig waren, mussten Möglichkeiten gefunden werden mit denen der Distanzsensor kalibriert werden konnte. Dazu wurden genaue Punkte auf einer Gummiente mit dem Laser markiert und dann die Steps aufgenommen die nötig waren um diese Ente genau 360° zu drehen. Für die kalibrierung des Schlittens in der Vertikalen wurde dieser über eine Skala mit Zentimetereinteilung gefahren. So ist es möglich geworden Methoden zu entwerfen die nicht mehr wie zuvor auf Steps vom Steppenmotor beruhen sondern um die Drehscheibe in einen bestimmten Winkel zu drehen und den Schlitten Millimeter genau hoch und runter zu bewegen.

4.2.4. Punkte aufnehmen

Die Punkte werden über die Kommunikation der seriellen Schnittstelle aufgenommen. Im ersten Versuch wurde der Laser-Distanzsensor an die richtige Stelle gefahren und dann nach

dem aktuellen Entfernungswert gefragt. Dies führte dazu, dass die Motoren ziemlich ausgelastet waren da sie immer nur sehr kurze Entfernungen zurück legten. Im nächsten Schritt wurde der Schlitten bis zum maximal angegeben Punkt bewegt und dazwischen alle Punkte alle n-Schritte aufgenommen. Diese Methode führte zu deutlich verkürzten Aufnahmen der Objekte und ein kontinuierlicher Ablauf beim Aufnehmen eines Objektes.

4.3. Entwurf der Softwarearchitektur

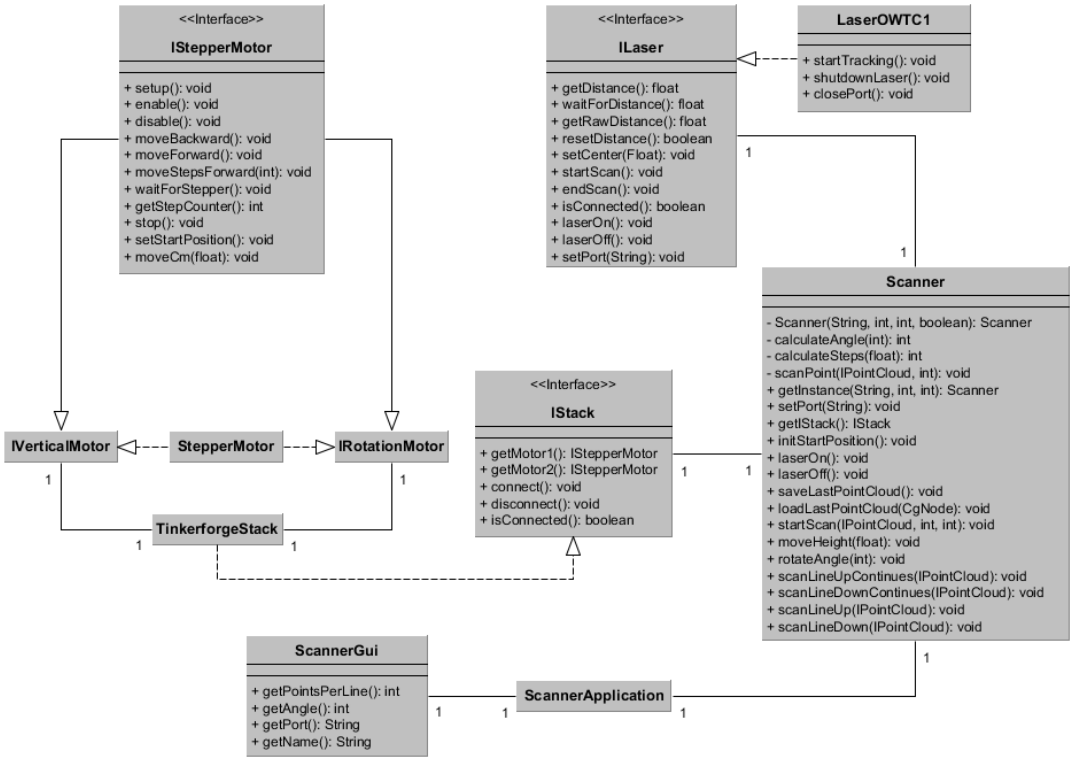

Abbildung 4.1.: Diagram der Softwarearchitektur

In diesem Punkt der Arbeit wird die Architektur in einem an UML angelehnten Klassendiagram dargestellt und in Abbildung 4.1 dargestellt.

Die Software Architektur ist vom *Scanner* aus aufgebaut. An dieser Stelle fließen alle Elemente die benötigt werden um die Benutzeroberfläche anzuzeigen, die Motoren zu bedienen oder aber um den Laser anzusprechen zusammen. So werden der Benutzeroberfläche in der *Scanne-*

rApplication alle benötigten Befehle gegeben um den Ablauf eines kompletten Scanvorgangs durchzuführen.

Die *ScannerApplication* stellt dabei die Einbindung des Computergrafik Framesworks da und besitzt wie auf dem Diagram zu sehen selber keine weiteren Methoden. Damit die Anwendung bedient werden kann, bindet die *ScannerApplication* eine *ScannerGui* ein, die dafür sorgt, dass alle nötigen Oberflächenbefehle richtig benutzt werden können.

Weiterhin beinhaltet der Scanner ein *IStack* hinter dem der *TinkerForgeStack* mit den beiden Steppermotoren steht. Der *TinkerForgeStack* und der *StepperMotor* wurden im Vorfeld von Prof. Dr. Philipp Jenke entworfen und konnten in dieser Form für die Bachelorarbeit weiter verwendet werden, ohne dass die Klassen mit Methoden erweitert werden mussten. Das Interface *IStepperMotor* bildet einen Motor ab, der nichts zwangsläufig über einen TinkerForge Baustein angesprochen werden muss, solange er alle Methoden vom Interface selber implementiert hat.

Der letzte Baustein vom *Scanner* ist der Laser. Dieser wird über das Interface *ILaser* implementiert. Dieses Interface bietet alle Funktionen die der Scanner benötigt um die aktuelle Entfernung des Scanners zu bekommen. Die Klasse *LaserOWTC1* ist eine genaue Implementierung des *ILaser* Interfaces für den Laser vom Bautyp OWTC1, so wie er in dieser Bachelorarbeit verbaut wurde.

Der *Scanner* selber wird über ein Singleton Pattern implementiert und bietet aus diesem Grund keinen konkreten Konstruktor der von außen erreichbar ist. Es ist wichtig, dass nur ein *Scanner* innerhalb des Versuchsaufbaus existiert. Ein zweiter *Scanner* könnte dafür sorgen, dass während eines laufenden Scans bestimmte Einstellungswerte wie zum Beispiel die der Geschwindigkeit des Schlittens verstellt werden und somit ein kompletter Scanvorgang kein verwendbares Ergebnis liefert.

4.3.1. OpenGL Framework

Innerhalb des Computergrafik Framework wird JOGL verwendet (JOGL, 2015). Dieses OpenGL Framework bietet eine sehr dichte Schnittstelle zu OpenGL. Es wird in diesem Framework auf viele Vereinfachungen verzichtet um eine möglichst genaue Abbildung aller Funktionen die OpenGL zur Verfügung stellt zu bieten. Innerhalb dieser Bachelorarbeit können so auf einer niedrigen Ebene die Punktwolken abgebildet werden.

4.3.2. Serieller Port Framework

Um mit dem Laser-Distanzsensor zu kommunizieren, ohne das auf eine Hardware nahe Sprache wie C zurückgegriffen werden muss, wurde sich in dieser Arbeit dafür entschieden die Java

Bibliothek Java Simple Serial Connector zu benutzten (JSSC, 2015). Die Bibliothek stellt über ein natives Interface eine Schnittstelle für die serielle Schnittstelle über System Bibliotheken zur Verfügung und bietet so eine leichte Verwendung innerhalb einer Java Anwendung.

4.3.3. Computergrafik Framework

Schon in den ersten Schritten der Bachelorarbeit musste darüber nachgedacht werden wie die Software am besten in das Computergrafik Framework integriert werden kann und was es für vorhandene Funktionen bietet, für welche die Integration des Scanners von Nutzen sein könnte. Schnell stellte sich dabei heraus, dass einfache Menüs ohne großen Aufwand integriert werden konnten. Dafür musste lediglich eine abstrakte Klasse geerbt werden und das Menü als Menü in der Software registriert werden. Da die komplette Oberfläche des Frameworks in Swing geschrieben wurde, musste auch für das neu erstellte Menü Swing benutzt werden.

Weiterhin werden nützliche Funktionen zur Verfügung gestellt um beispielsweise eine Punktwolke darzustellen. Dazu muss einfach nur eine neue leere Punktwolke erstellt werden und anschließend nach und nach Punkte zu dieser Wolke hinzugefügt werden. Dies traf sich mit der Funktionsweise des 3D Scanners recht gut und konnte so verwendet werden, dass der aktuell gescannte Punkt direkt angezeigt wurde.

Des Weiteren kann das Framework dafür verwendet werden um schnell Punkte und andere Werte in eine Datei zu schreiben. So kann in kürzester Zeit ein Speicherformat für die Punktwolken erstellt werden was auch wieder von der Software geladen werden kann. Alles was dafür nötig ist, ist das Format über die Klasse AsciiPointFormat zu konfigurieren.

4.3.4. Modularer Aufbau

Innerhalb der Software für den 3D Scanner, gibt es die Möglichkeiten einzelne Komponenten auszutauschen und durch andere Komponenten zu ersetzten. So gibt es für die Distanzsensoren und für die Motoren ein Interface gegen welches implementiert werden kann um Bausteine schnell auszutauschen. Des Weiteren gibt es auch schon andere Module für diese Interfaces um die Hardware zu simulieren. Dies ermöglicht leichte Tests durchzuführen ohne auf die physische Präsenz des Scanners angewiesen zu sein.

Die Simulations Module sind auf das gröbste reduziert und bewegen lediglich den simulierten Distanzsensor hoch und herunter und drehen das Objekt um 360°. Der simulierte Distanzsensor erzeugt jedes mal die selbe Entfernung um so ein zylinderähnliches Punktwolken-Abbild zu erstellen. In der Zeit der Entwicklung der Software war es leichter, durch die Simulation des Scanners Scans durchzuführen, als jedesmal einen sehr lange dauernden Scan durchzuführen.

4.4. Implementation der Software

In diesem Teil der Bachelorarbeit wird darauf eingegangen wie die Software umgesetzt wurde. Dabei ist die Reihenfolge der Sektionen sinnvoll gegliedert von den Anfängen der Schnittstellen bis zur eigentlichen Umsetzung der Software für den Scanner.

4.4.1. Schnittstelle zum Laser-Distanzsensor

Um eine Schnittstelle zum Laser-Distanzsensor zu implementieren, ist es nötig eine Verbindung über die serielle Schnittstelle herzustellen. Damit dieses an möglichst vielen Maschinen möglich ist wurde vom seriellen Kabel auf den Computer ein Seriell auf USB Adapter benutzt. Nach anfänglichen Problemen konnte die Verbindung hergestellt werden. Der Hersteller des OWTC1 hat eine ausführliche Dokumentation in Deutsch und in Englisch über alle möglichen Befehle herausgegeben. Bei den Befehlen werden über die Simple Serial Java Connecter Bibliothek Zeichen in Form von Java Strings an die Schnittstelle des Lasers gegeben. Dieser schickt anschließend eine Nachricht zurück an die Software aus der abzulesen ist, ob der Befehl erfolgreich oder missglückt ist.

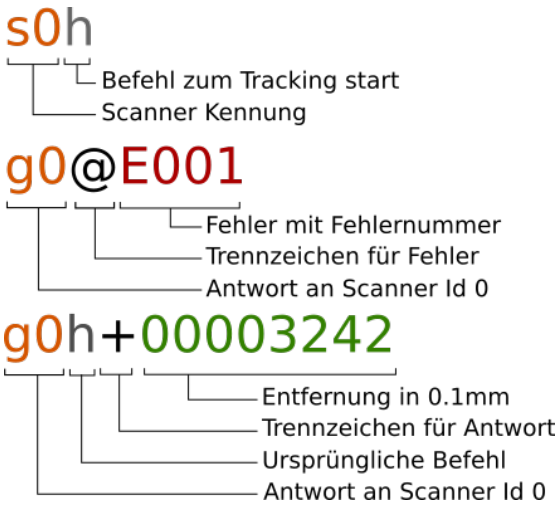

Abbildung 4.2.: Schema der Seriellen Funktionen

Da die Befehle keine einheitliche Form haben, zu erkennen auf der Abbildung 4.2, wurde für die Schnittstelle des Sensors mit genau diesem Laser eine Klasse entworfen, die die Kommunikation mit dem Laser vereinfacht. So kann beispielsweise auf feste Konstanten zurückgegriffen werden, welche einen ersichtlicheren Namen als die Buchstabenfolge trägt.

Um auf Antworten von den Anfragen des Scanners zu reagieren, wird über die Simple Serial Java Connecter Bibliothek ein Event Listener auf die serielle Schnittstelle gelegt. Dieser Listener wird immer genau dann ausgelöst wenn über die Schnittstelle ein Signal zum Computer geschickt wird. Innerhalb des Listeners werden die ankommenden Bytefolgen, welche als Strings interpretiert werden, auf die Antworten vom Scanner abgeglichen. Eine Antwort an die Software beginnt jeweils mit einem kleinen g gefolgt von der Identifikationsnummer des Laser-Distanzsensors. Dadurch das nur einer verbaut ist, wird immer die Identifikationsnummer 0 zurückgegeben. Gefolgt von dieser Kombination wird erneut das Befehlszeichen gesendet und anschließend als Trennzeichen ein Plussymbol. In der Abbildung 4.2 ist in der dritten Kombination die Antwort eines Trackingbefehls zu sehen. In genau diesem Beispiel wird hinter dem Pluszeichen die Entfernung in 0.1mm zurückgegeben.

Die Schnittstelle zum Laser-Distanzsensor bietet dazu auch noch weitere Funktionen wie das Ein- und Ausschalten des Lasers ohne eine Messung zu starten. Dies ist eine sinnvolle Methode um den Laser-Distanzsensor manuell auf einen bestimmten Punkt auszurichten. Es kann dazu verwendet werden um beim Einbau in den Schlitten den Laser auch wirklich auf den Mittelpunkt zu justieren. Wichtig zu wissen ist, dass keine Messung gestartet werden kann solange der Laser-Distanzsensor den Laser ohne Messung aktiviert hat und noch keinen Befehl zum Beenden empfangen hat.

4.4.2. Schnittstelle zum TinkerForge Stack

Die Schnittstelle zum TinkerForge Stack wird über eine USB Verbindung realisiert. Anders als beim Laser-Distanzsensor musste für diese Schnittstelle nicht extra ein Treiber herausgesucht werden. Die API der Bausteine ist so aufgebaut, dass jedes Brick eigene Funktionen zum Bedienen hat. Da der Scanner ein Masterbrick und zwei Stepper Bricks verwendet, war es nur nötig die Befehle der Stepperbricks zu benutzten. Die Verbindung von dem Masterbrick, an dem auch der Rechner angeschlossen ist, zu den beiden Stepperbricks funktioniert vollkommen automatisch nachdem die eindeutigen IDs der Stepperbricks registriert sind und eine Verbindung zum Masterbrick hergestellt wurde.

Des Weiteren bietet der Stepperbrick die Möglichkeit bestimmte Konfigurationen vor zu nehmen. So kann das Modul mit schon vorhandener Konfiguration initialisiert werden. Unter der Konfiguration gibt es die Möglichkeiten die Beschleunig, die Bremsträgheit und die maximale Geschwindigkeit des Motors zu konfigurieren. Weiterhin kann angegeben werden in was für Schritten gezählt werden soll. Dabei ist es möglich ein Schritt zu halbieren, vierteln oder zu achteln.

Nun konnten die beiden Motoren des Scanners speziell an die Software von TinkerForge angepasst werden. Im ersten Entwurf der Software wurde der Motor für den Schlitten immer nur soweit bewegt wie für die nächste Messung nötig war. Dies wurde beim zweiten Entwurf geändert zu einem kontinuierlichen Ablauf im Scan Verhalten.

Ansteuerung des Schlitten

Der Schlitten wird über einen der beiden Stepperbricks angesteuert. Innerhalb der Software ist es möglich über eine eindeutige ID genau diesen Stepper zu konfigurieren oder ihn zu bewegen. Verwaltet wird die Klasse des Motors über die TinkerForge Klasse, welche die Verbindung abstrahiert und den Zugriff auf beide Motoren bereitstellt. Zur leichteren Kontrolle über den Schlitten gibt es neben den normalen Schritt Bewegungen des Stepper Motors auch eine Methode um den Schlitten eine gewisse Millimeter Anzahl nach oben oder unten zu bewegen. Durch diese Art ist es leichter Konfigurationen und Tests vor zu nehmen. Die am Schlitten eingesetzten Stepper Motoren haben eine Auflösung von 1,8 Grad pro Schritt, dies ist entnehmbar von dem Aufkleber auf den Stepper Motor oder aus dem Datenblatt des K8200 Gerüstes (Velleman, 2013).

Ansteuerung der Drehscheibe

Genau so, wie auch der Schlitten vom TinkerForge Stack verwaltet wird, wird auch der Motor der Drehscheibe über diesen Stack und einer eindeutigen ID verwaltet. Auch gibt es bei der Drehscheibe die Möglichkeit den Stepper um eine gewisse Anzahl an Steps zu bewegen, was dazu führt das der Drehteller sich dementsprechend dreht. Um leichter mit den Drehteller arbeiten zu können gibt es eine Methode die dafür sorgt das der Drehteller sich um einen gewissen Grad dreht. Die Mathematische Berechnungen an dieser Stelle werden in Grad angegeben. Der Motor der an der Drehscheibe eingesetzt wird, ist identisch mit dem des Schlittens, und hat somit die Auflösung von 1,8 Grad für einen Schritt (Velleman, 2013).

Konfiguration der Motoren

Die Motoren werden durch sogenannte Stepperbricks gesteuert. Dabei handelt es sich um Module aus der TinkerForge Familie, die die Steuerung der Steppermotoren einfach für den Endanwender halten. TinkerForge bietet verschiedene Möglichkeiten die Motoren zu konfigurieren wobei ein Step immer die gleiche Rotation im Motor bleibt. So bedeutet dies, dass selbst wenn in ein Achtel Schritten gezählt wird, wird der vollkommene Schritt bei der Hardware erst dann erreicht wenn dieser um Acht Schritte weiter gezählt wird. Die wichtigsten Einstellungen

bei dem 3D-Scanner sind die Beschleunigung, maximale Geschwindigkeit und Bremsträgheit. Diese Werte sollten nicht zu gering eingestellt werden, da ansonsten der Motor lautstarke Geräusche von sich gibt. Wenn die Werte wiederum zu hoch konfiguriert wurden gibt es Probleme da der Laser nicht schnell genug die Entfernung von dem Objekt aufnehmen kann und somit die komplette Punktwolke verzogen und nicht mehr brauchbar wird. Ein ideales Maß an Beschleunigung und Geschwindigkeit muss getestet werden, aktuell sind die Werte 500 für maximale Geschwindigkeit und Bremsträgheit und 1000 für Beschleunigung konfiguriert, so das der Scanner möglichst schnell ein Scan ausführen kann, ohne dass die Resultatpunktwolke unter der Geschwindigkeit leidet. Da die Algorithmen zum kontinuierlichen Scannen auf den Werten der Steps basieren, ist es egal wie die Werte eingestellt sind solange die Hardware damit klar kommt, was viel Möglichkeiten zum Spielen lässt.

Außerdem kann bei den Stepperbrick auch noch auf Optionen wie ein sofortiges Stoppen und die Einstellung vorgenommen werden, ob der Stepper volle, halbe, viertel oder Achtelschritte ausführt. Dieser Wert sollte nicht von ganzen Schritten umgestellt werden, da ansonsten das Ergebnis verfälscht werden könnte und ein Step nicht mehr die Wertigkeit von einem Step besitzt.

4.4.3. Finden des Nullpunktes

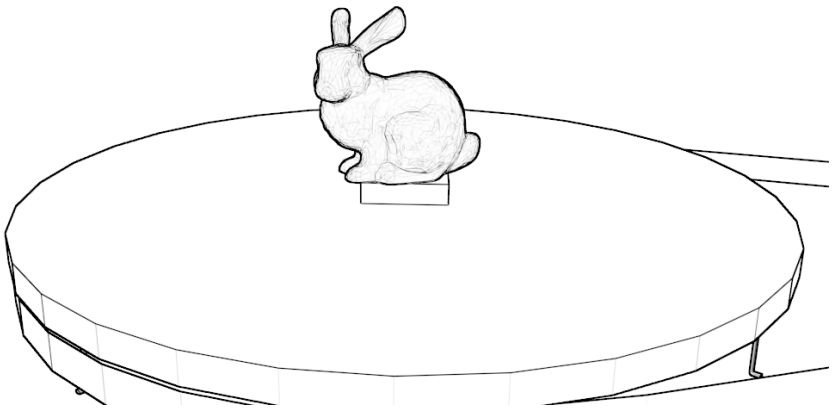

Abbildung 4.3.: Sockel mit Objekt

Da der 3D Scanner bei einem kompletten Neustart nicht die aktuelle Position der Steppermotoren kennt, müssen diese auf den eigentlichen Nullpunkt neu justiert werden. Wie auf der Abbildung 4.3 zu sehen ist, gibt es auf dem Drehteller einen Sockel auf dem das spätere Objekt

platziert wird um es zu Scannen. Dieser Sockel wird benötigt, da der Schlitten nicht bis zu der untersten Kante des Drehtellers fahren kann. Um den Scanner nun auf den Nullpunkt zu justieren darf kein Objekt auf dem Sockel stehen sobald der Initialisierungsvorgang gestartet wurde.

Der Scanner versucht bei jedem Start einen Nullpunkt zu finden, indem er den Schlitten so lange abwärts bewegt, bis der Scanner ein Objekt im gültigen Scannbereich gefunden hat. Dieses Objekt sollte bei einer erfolgreichen Justierung der Sockel auf dem Drehteller sein. Sobald der Sockel gefunden wurde, wird die gefunden Stelle als Nullpunkt markiert indem die Schrittzähler der Motoren zurückgesetzt werden auf Null.

4.4.4. Kontinuierliches Scannen

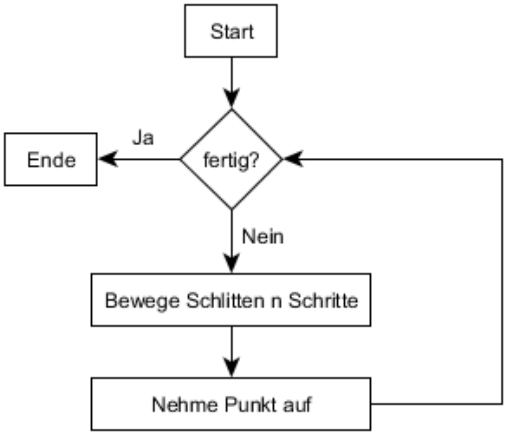

Abbildung 4.4.: Altes Verfahren um Punkte aufzunehmen

Bevor das kontinuierliche Scannen implementiert wurde, war es notwenig den Laser-Distanzsensor an die Position zu bewegen an dem er den nächsten Punkt aufnehmen sollte. Wie es in Abbildung 4.4 zu sehen ist.

Durch die Erweiterung des kontinuierlichen Scannen war es möglich, die Zeit zu verkürzen die der Scanner benötigt um ein komplettes Objekt aufzunehmen. Der Ablauf wie dieser umgesetzt werden sollte, wird in der Abbildung 4.5 als Diagram abgebildet. Implementiert wurde es dabei wie folgt, der Steppenmotor wurde am Anfang auf die Grundposition gefahren und der Schrittzähler wurde anschließend auf die Zahl Null gesetzt. Von nun an konnte dem Schlitten der Befehl gegeben werden das er, zum Beispiel zu 10cm Höhe, fahren soll. Während der Schlitten auf dem Weg ist werden die Steps vom Steppenmotor kontinuierlich hoch gezählt.

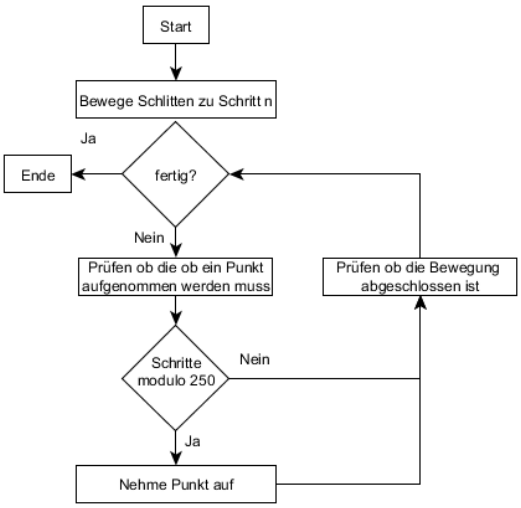

Abbildung 4.5.: Neues Verfahren um Punkte aufzunehmen

Nun wird jedes mal, wenn der Wert vom Steppenmotor durch eine vorgegebene Zahl teilbar ist ein Punkt aufgenommen. Dieses Verfahren macht es sehr einfach die Genauigkeit eines Scans einzustellen. Alles was dafür getan werden muss ist die Zahl durch die die Schritte geteilt werden sollen auf einen gewünschten Wert zu konfigurieren. Dabei gilt, je kleiner der Wert ist desto mehr Aufnahmen der Punkte werden auf der gleichen Strecke gemacht.

Auch wenn dieser Vorgang den Scanvorgang sehr beschleunigt bedarf es sehr viel Kalibrierung zwischen der Geschwindigkeit des Steppenmotors und der des Laser-Distanzsensors. Aus dem Versuch ging hervor, dass der Distanzsensor ungeeignet ist um sehr schnelle Wiederholraten beim Abtasten des Gegenstandes zu erzielen und so war trotz neuem Ansatz die Geschwindigkeit ungefähr bei einem Drittel von dem was die Steppenmotoren schaffen würden, da der Abtastvorgang sehr lange dauert.

4.4.5. Anpassen der vorhandenen GUI

Bei der Arbeit war durch das Computergrafik Framework schon eine Benutzeroberfläche vorgegeben. Diese konnte nicht wie gegeben verwendet werden, sondern bedurfte es ein eigenes Menü mit Java Swing zu implementieren, zu sehen auf der Abbildung 4.6. Über diese Benutzeroberfläche ist es nun möglich die Verbindung und Justierung des Scanners zu starten. Durch den Startbutton bewegt sich der Schlitten solange nach unten bis er im scanbaren Bereich eine Entfernung gefunden hat. Der erste Punkt der im Regelfall gefunden werden

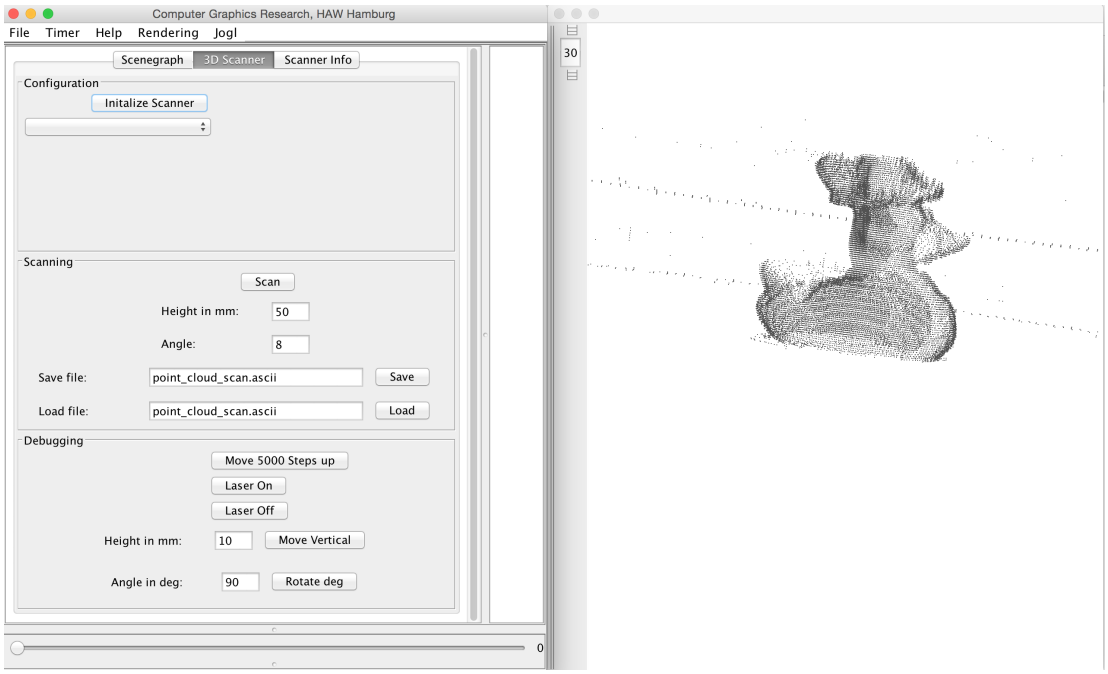

Abbildung 4.6.: Screenshot der Benutzeroberfläche

sollte, ist der Sockel auf dem im späteren Verlauf das Objekt positioniert wird. Diesen Punkt benutzt der Scanner anschließend als Nullpunkt.

Nachdem der Scanner die Position gefunden hat, ist es möglich die Auflösung und maximale Höhe des Objektes einzugeben. Diese Werte sorgen sehr stark dafür wie lange der spätere Vorgang in Anspruch nehmen wird.

Sollte der Scan komplett abgeschlossen sein, gibt es eine Möglichkeit diese Punktwolke innerhalb einer Datei zu speichern und auch im späteren Verlauf durch eine Öffnen-Funktion im Menü wieder herzustellen.

4.4.6. Erstellung eines Speicherformates

Bei der Erstellung eines Speicherformates geht es um ein Format welches durch die Scannersoftware in ein Textformat ausgegeben werden kann und für spätere Zwecke wiederverwendet werden kann. So gehen keine Informationen die während eines Scanvorganges gesammelt werden verloren.

Punktwolkenformat

Nachdem ein Scan erfolgreich gelaufen ist, ist es wünschenswert die dabei entstandene Punktwolke irgendwie ablegen zu können anstelle jedesmal wieder einen neuen Scan anzufertigen. Um dieses Problem zu lösen bietet die Software die Möglichkeit die Punktwolke in einem eigenen Format zu speichern. Das speichern benutzt ein einfaches Format, welches schnell wieder eingelesen werden kann.

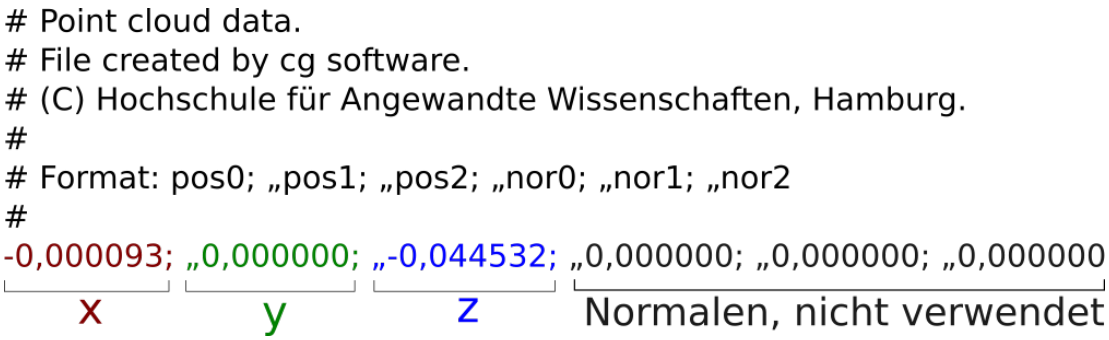

Abbildung 4.7.: Kopf und erste Datenzeile im Punktwolkenformat

Das Format, welches in Abbildung 4.7 grafisch dargestellt wird, in der die Punktwolke gespeichert wird, beginnt mit einem Kopfbereich. Dieser Kopfbereich wird durch Rauten markiert. Diese Rauten stellen sicher, dass alles was hinter der Raute kommt ein Kommentar ist. So kann beim späteren einlesen des Formates schnell erkannt werden wo der Kopfbereich ist und somit diese Zeilen ignoriert werden. Der Kopfbereich hält Informationen darüber mit welchen Program die Datei erstellt wurde und wer der Urheber des Scans ist. Des Weiteren wird auch in einer Kurzform ein Beispiel für das Format angegeben.

Das Format fängt mit einem kleinen Header an, in denen Informationen zum Autoren des Scans oder auch zum eigentlichen Objekt gespeichert werden können. Anschließend kommt der Teil des gescannten Objektes. Für jeden aufgenommenen Punkt gibt es eine einzelne Zeile in dieser die x, y und z Koordinaten beschrieben sind. Der Nullpunkt des Koordinatensystems ist der Mittelpunkt der Drehscheibe und die Position an dem der Scanner die erste Entfernung gefunden hat. Wenn der Ablauf der Kalibrierung erfolgreich ist, sollte dies der Sockel sein auf dem das Objekt steht. Weiterhin beinhaltet das Format auch eine Angabe für Normalen. Diese wird aber durch den ASCII Point Writer aus dem Computergrafik Framework geschrieben und wird beim Speichern eines 3D Scans nicht verwendet.

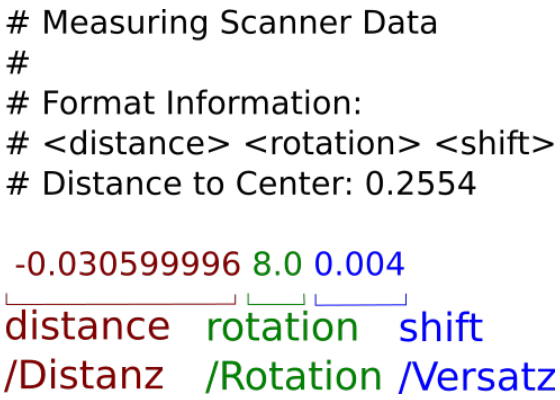

Abbildung 4.8.: Kopf und erste Datenzeile im Fehlerkorrekturformat

Fehlerkorrekturformat

Neben dem normalen Format um eine Punktwolke aufzunehmen gibt es auch ein weiteres Format welches zur kalibrierung gescannter Daten dient. Das Format besteht wie auch das Punktwolkenformat aus drei Werten pro Scan, zu sehen in der Abbildung 4.8, nur sind es dieses Mal nicht die Werte x, y und z sondern die Entfernung, die Rotation und die Versetzung. Aus diesen Werten kann ein bekanntes Objekt zurück gerechnet werden um dann die entsprechende Form zu rekonstruieren. Wenn dies geschehen ist, kann anhand der Abweichungen jedes Punktes bestimmt werden in welchem Umfang das gescannte Objekt von den eigentlich Objekt abweicht und in was für einer Form es eine Streckung oder Stauchungen gibt. Dieses Format könnte auch weiter dazu benutzt werden um festzustellen wie genau der Laser montiert wurde und gegebenfalls diesen noch besser ausrichten.

Objekt kalibrierung

Bei der Objekt-Kalibrierung ging es darum, Fehler in dem gescannten Objekt zu finden und diese anschließend zu korrigieren. Nachdem die ersten Objekte durch den Scanner aufgenommen wurden, stellte sich heraus, dass durch Fehler in der Justierung des Laser-Distanzsensors Objekte nicht über die Mitte der Drehscheibe aufgenommen wurden. Das führte dazu, dass Stauchungen und Streckungen in der Punktwolke entstanden.

Um Fehler wie diese zu vermeiden wurde zuerst einmal eine Dose aufgenommen. Bei dieser Dose war der Durchmesser bekannt. Da diese Dose einen Zylinder als Punktwolke zurück gegeben hat und der Durchmesser dieser Punktwolke bekannt war, war es nun möglich durch Zurückrechnen heraus zu bekommen an welchen Stellen diese Fehler eintraten. Zusätzlich zum Scan wurde nun ein Algorithmus, der durch Variablen anpassbar ist, implementiert. Durch

diesen Algorithmus ist es möglich falsch justierte Hardware in einem bestimmten Rahmen zu tolerieren und die falsch aufgenommenen Punkte in ihre ursprüngliche Position zurück zu berechnen.

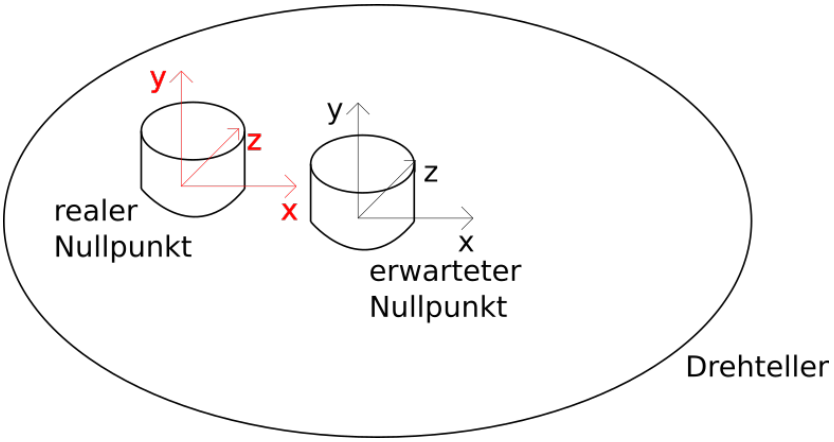

Abbildung 4.9.: Kalibrierung eines Zylinder

Der Algorithmus berechnet nun beide Koordinaten Systeme zu einem Welt Koordinatensystem zusammen (Jenke, 2014b). Grafisch ist das Problem auf der Abbildung 4.9 dargestellt. Der Nullpunkt des realen Objektes liegt nicht auf dem erwarteten Nullpunkt der Drehscheibe. Da mit einem zylinderförmigen Objekt keine Rotationsprobleme auftreten können und somit leicht der Nullpunkt gefunden werden kann, wurde eine Dose eingescannt mit der dann die Abweichungsvariablen richtig justiert werden konnten.

4.5. Evaluation

Nachdem die Software implementiert wurde, ist es möglich gescannte Objekte mit realen Objekten abzugleichen. Bei dem Vergleich des Scans kann schnell festgestellt werden, dass die erwarteten Oberseiten des Objektes nicht aufgenommen werden konnte, oder nur mit einer niedrigen Auflösung an Punkten aufgenommen werden konnte.

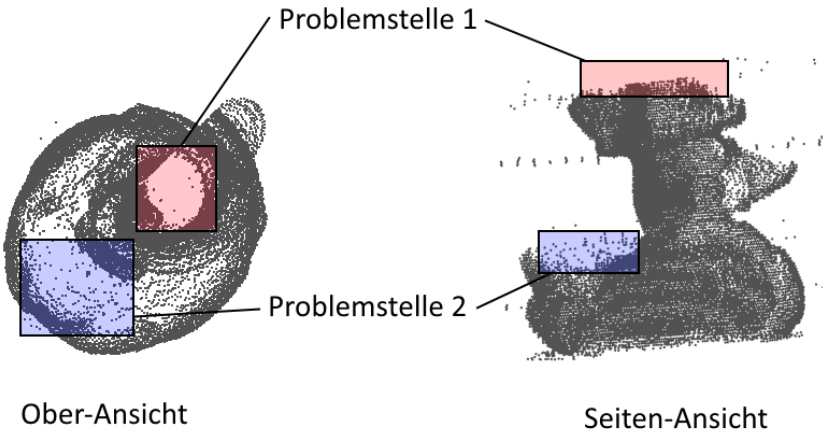

Abbildung 4.10.: Auswertung hochaufgelöster Gummiente

Um zu schauen ob die vorhergesagten Probleme wirklich auftreten, wurde zum Testen eine Gummiente gescannt bei welcher bestimmte Punkte genauer betrachtet werden können. Zu sehen sind diese Punkte auf der Abbildung 4.10 rot und blau markiert. Die markierten Punkte wurden nach ihrer Lage an dem Objekt ausgewählt. So konnte vor dem Scan vorhergesagt werden, dass die Oberseite des Schnabels der Ente und der Rücken Problemzonen für den Scan darstellen.

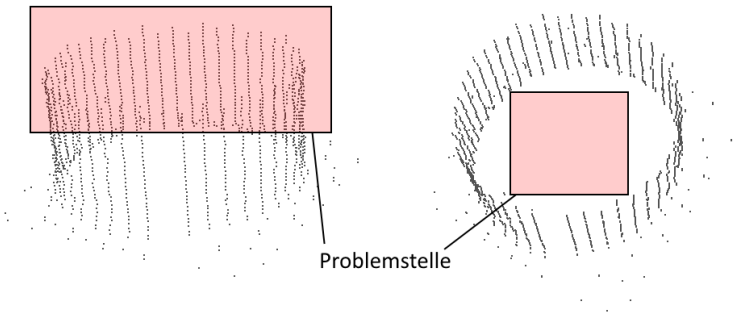

Abbildung 4.11.: Scan einer Dose

Um zu bestätigen, dass die Lage des Scanners das Problem darstellt wurde ein zweites Objekt nach den Kriterien des ersten Objektes untersucht. So konnte auch bei dem zweiten Objekt,

wobei es sich um den Scan einer Dose handelte, wieder die Oberseite als Problem markiert werden, da dieser Bereich vom Laser-Distanzsensor nicht erreicht werden kann. Zu sehen ist der Bereich, wie bei der Gummiente auch, auf der Abbildung 4.11 rot markiert.

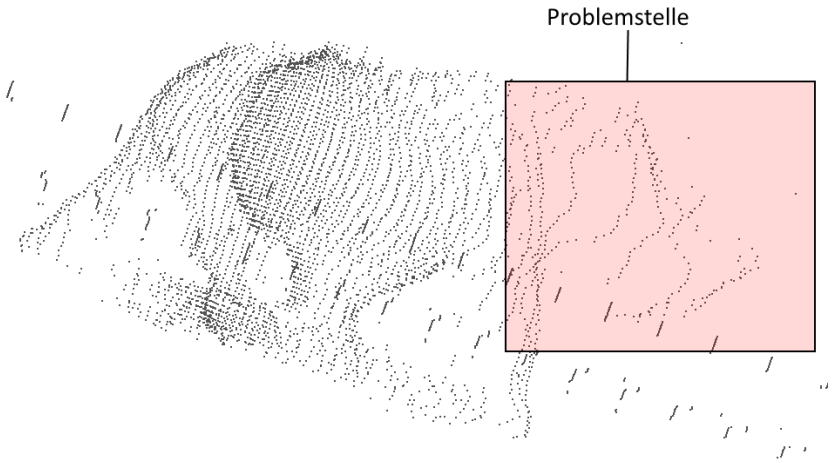

Abbildung 4.12.: Scan einer Nashornstatue

Ein anderes Problem ist die veränderte Auflösung vom Mittelpunkt bis zum Rand des Drehtellers. Zu erwarten ist, dass am Rand des Drehtellers bei gleicher Rotation die Punkte einen weiteren Abstand von einander bekommen. Durch den Scan einer kleinen Nashornstatue konnte dies bestätigt werden. So ist klar zu sehen, dass der Kopf des Nashorns dichter am Rand des Drehtellers ist, als der Körper der Statue. Auf der Abbildung 4.12 ist dieses Verhalten im rot makierten Bereich zu erkennen, wo der Kopf weit auseinadner liegende Punkte aufweißt, liegen die Punkte am Körper dicht bei einander.

5. Aufgetretene Probleme

Während der Bachelorarbeit sind zahlreiche Probleme aufgetreten. Diese Probleme werden in diesem Kapitel der Bachelorarbeit festgehalten um auch Bachelorarbeiten welche auf dieser aufbauen die Möglichkeit zu geben schnell Lösungen für die aufgetretenen Probleme zu finden.

5.1. Hardware Probleme

Bei den Hardware Problemen geht es ausschließlich um Probleme welche aufgetreten sind im Zusammenhang mit der Hardware. Diese Probleme können nicht durch die Software oder verbesserte Algorithmen ausgebessert werden und mussten physikalisch korrigiert werden.

5.1.1. Umsetzung von Motor auf Teller

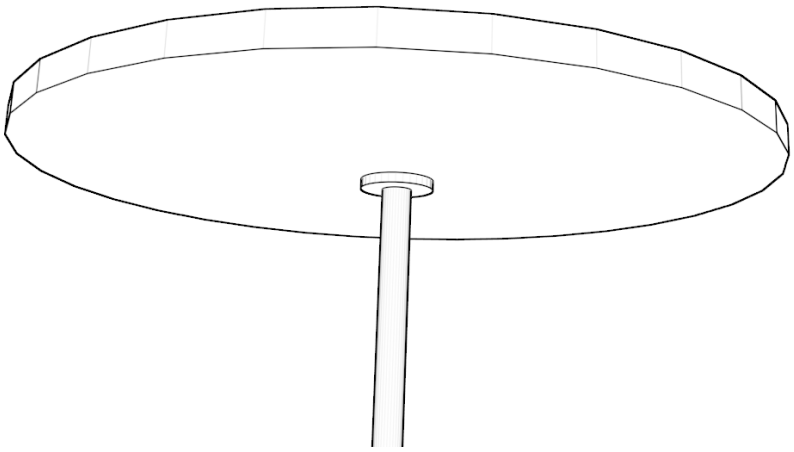

Abbildung 5.1.: Entwurf 1 und 2

Nachdem der Scanner das erste Mal im Betrieb genommen wurde, konnten gleich einige Probleme identifiziert werden. Eines dieser Probleme war der Drehteller, zu sehen in der Abbildung 5.1. Im ersten Entwurf des Scanners schien die Umsetzung vom Motor auf die

Drehscheibe nicht zu stimmen und die Scheibe war mit einem Gewinde am Gehäuse angebracht. Dies sorgte dafür, dass sich der Teller mit jedem Scanvorgang etwas weiter nach oben gedreht hatte und später dann ein Stück nach unten gesprungen ist. Dieses Problem hätte zum einen dafür gesorgt, dass die Werte vom Laser in der vertikalen verfälscht worden wären aber auch dass der Drehteller sich ungleichmäßig bewegte und hin und wieder einen kleinen Sprung gemacht hatte. Weiterhin sorgte die geringe Umsetzung dafür, dass die Drehung eher stockend als flüssig war. Alle diese aufgetretenen Probleme waren über die Software nicht sichtbar, da der Stepperbrick trotz stehenden Motor weiter gezählt hat. Ein Scannen mit dieser Umsetzung war schlichtweg unmöglich.

Im zweiten Entwurf wurde das Gewinde durch eine Röhre ersetzt in welcher sich der Drehteller frei bewegen konnte. Diese Änderung hat leider nur geringfügige Änderungen bewirkt wodurch dann das Problem aber klar in der Umsetzung von dem kleinen Schrittmotor auf den großen Drehteller identifiziert werden konnte.

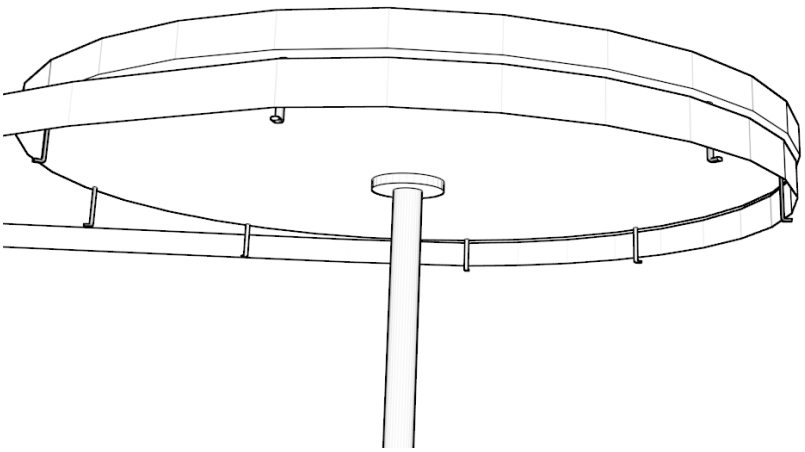

Abbildung 5.2.: Entwurf 3

Im dritten und letztendlich letzten Entwurf wurden die Führungen für den gezähnten Gummiriemen am äußeren Rand des Drehtellers befestigt. Dies geschieht indem an den Drehteller kleine Haken angebracht wurden, wie es auch in der Abbildung 5.2 zu erkennen ist. Da nun die Umsetzung auf den Rand des Drehtellers liegt, gab es keine weiteren Probleme mit den genauen Bewegungen in kleinen Schritten.

5.1.2. Höhe des Sockels

Eigentlich war es vorhergesehen, dass der Sockel die Höhe für den Nullpunkt bestimmt. Da dieser aber etwas zu niedrig ist, ist es für den Laser-Distanzsensor unmöglich bis an den Sockel zu kommen. Dieses Problem ist bei den ersten Versuchen der Justierung auf den Nullpunkt aufgefallen.

Sollte der Sockel nicht erhöht werden, gibt es Problem mit dem Schlitten. Dieser fährt solange herunter bis er einen gültigen Sachwert ermittelt. Wenn der Sockel seine normale Höhe hat, fährt sich der Schlitten an der unteren Kante fest und die Justierung muss abgebrochen werden, da der Motor überdreht.

Im Testfall der Bachelorarbeit war es kein Problem den Sockel etwas zu erhöhen damit alles ohne Probleme funktioniert. In der Zukunft sollte dieser etwas verlängert werden, damit es bei der Justierung keine Probleme gibt.

5.1.3. Motor vom Schlitten hängt

Ein sehr häufig aufgetretenes Problem war, dass der Schlitten sich aufhängt während er sich hoch oder runter bewegt. Dieses Verhalten ist sehr ungünstig, da der Stepperbrick von TinkerForge das Feststecken nicht wahrnehmen kann und so ein verfälschtes Bild entstehen würde. Die angewandte Lösung um dieses Problem zu beseitigen war die Gewindestange mit etwas Öl zu schmieren. Andere Lösungen für eine langfristige Lösung wäre es den Steppenmotor noch fester am Gehäuse zu montieren oder aber eine zweite Gewindestange an der anderen Seite des Schlittens zu montieren. Dieses zweite Gewinde müsste anschließend mit dem ersten verbunden werden und würde den Schlitten an beiden Seiten geichzeitig anheben.

5.1.4. Versatz des Lasers

Nachdem die ersten kompletten Scans erfolgt sind wurde schnell klar, dass die Objekte an einem Punkt sehr verstaucht sind und an einem anderen sehr weit gedehnt. Die Ursache für dieses Problem war die Justierung des Laser-Distanzsensors auf den Mittelpunkt der Drehscheibe. Der Laser verfehlte das Ziel um ein paar Millimeter was dazu führte das der eigentlich gedachte Nullpunkt in der Mitte der Drehscheibe etwas neben den eigentlichen Nullpunkt lag.

Das Problem wurde beseitigt durch ein neues Speicherformat aus dem genauere Scannerwerte ausgelesen werden konnten und mit welchem der Scan anschließend zurückgerechnet werden konnte. Nun existiert ein Algorithmus der den Versatz des Lasers ausgleicht. Der Grund warum nicht einfach der Distanzsensor auf den Mittelpunkt ausgerichtet wurde liegt in der

Tatsache, dass der Distanzsensor leicht ausgetauscht werden kann und so nicht bei jedem Ausbau ein sehr genaues Justieren durchgeführt werden muss.

5.2. Software Probleme

Bei den Software Problemen handelt es sich ausschließlich um Probleme die mit der Software oder im Zusammenhang mit Geräten welche über die Software kommunizieren vorgekommen sind. Die in diesem Unterkapitel aufgezählten Probleme sind nur über die Software zu korrigieren oder Systemeinstellungen von entsprechenden Betriebssystemen.

5.2.1. Treiberprobleme unter Windows 8

Ein großes Problem mit der Software war es, für den seriellen Adapter, den passenden Treiber unter Windows 8 zu finden. Zwar wurde vom Hersteller genau angegeben um was für einen Chip es sich handeln sollte, doch war dies vermutlich nicht der der angegeben war. Nach langem Recherchieren wurde dann klar das die Treiberinstallation ein Problem hatte, der darin bestand dass sobald einmal ein neuerer Treiber unter Windows 8 installiert worden war, war es selbst durch eine Deinstallation nicht möglich den älteren Treiber wieder auszuwählen. Um dieses Problem zu beheben musste in der Liste der Treiber für den seriellen Chip manuell der Eintrag für den Treiber vom Jahr 2008 ausgewählt werden.

5.2.2. Abstürze im Computergrafik Framework

Dieses selten auftretende Problem hat am Anfang viele Probleme bereitet. Da ein Scan von einem etwas größeren Objekt viel Zeit in Anspruch genommen hat, waren zuerst die Einstellungen sehr gering eingestellt zum Testen. Als dann größere Objekte gescannt wurden, mit deutlich mehr Punkten, ist teilweise die Software mit einem nicht nachvollziehbaren Fehler von OpenGL mit einem Backtrace in die Binaries abgestürzt. Es stellte sich heraus, dass es einen Fehler im Computergrafik Framework gab bei denen es ab einer gewissen Menge an Punkten in einer Punktwolke ausgelöst wurde. Durch dieses Verhalten schien an einer Stelle in einen nicht reservierten Speicherbereich geschrieben worden zu sein was dazu führte, dass die komplette Anwendung abstürtzte und den Benutzer noch nicht einmal die Möglichkeit ließ seinen Scan zu speichern.

Durch eine Änderung im Computergrafik Framework wurde dieser Fehler anschließend behoben und sollte in der Zukunft nicht erneut auftreten.

5.2.3. Abstürzende TinkerForge Firmware

Durch langes Scannen ist der Fehler aufgetreten, dass teilweise der komplette TinkerForge Stack ausgefallen ist. Dieser war anschließend nur durch manuelles Strom entziehen und wieder hinzufügen in der Lage neuzustarten. Die erste Vermutung dieses Problems war es, dass der TinkerForge Stack zu heiß lief und sich selber abschaltete. Nach dem Einbau eines Hitzesensors konnte dieses Problem schnell ausgeschlossen werden. Das Problem wurde anschließend in einer veralteten Firmware gefunden. Diese Firmware hatte einen Bug, welches den kompletten Stack zufällig zum Absturz bringen konnte. Nachdem alle Bricks vom TinkerForgestack mit der aktuellsten Firmware geflasht wurden, ist das Problem mit den abstürzenden Bauteilen nicht erneut aufgetreten.

5.2.4. Lärm bei zu niedrigen Motoreinstellungen

Durch eine Fehlkonfiguration wurden die Steppenmotoren dazu gebracht in einer sehr niedrigen Geschwindigkeit zu fahren. Dieses Problem ist nicht wieder aufgetreten nachdem die Motor Konfiguration auf einere höhere Geschwindigkeit gestellt wurde. Weiterhin ist hier als Problem zu erkennen, dass während eines Scanvorgangs nicht die von TinkerForge mitgelieferte Software Brick Manager geöffnet werden darf um Werte aus den Bricks abzulesen. Es ist möglich die Konfiguration des 3D-Scanners so sehr zu verstellen, dass die Scannersoftware komplett neugestartet werden muss um die Grundkonfiguration erneut zu laden.

5.2.5. Länge eines Scanvorgangs

Bei dem ersten Software Entwurf wurde der Schlitten und die Drehscheibe auf einen genauen Punkt bewegt um anschließend vom Laser-Distanzsensor den aktuellen Wert der Position abzufragen. Dieses Verfahren wurde für jeden einzelnen Punkt durchgeführt. Dadurch, dass der Motor an eine Position fahren, stoppen, scannen, weiterfahren musste, ging viel Zeit durch das Beschleunigen und Abbremsen des Motors verloren. Auch bestand die Messung aus vielen kleinen Untermessungen um einen Mittelpunkt festzustellen.

Das Problem wurde im zweiten Entwurf dadurch gelöst, dass der Scanner auf einen durchgängigen Trackingmodus gestellt wurde und somit immer den letzten gemessenen Punkt abrufbar machte. Der Schlitten wurden nun nicht mehr von Punkt zu Punkt bewegt, sondern von dem Nullpunkt bis zur maximalen Höhe und wieder herunter. Jeder Punkt der dazwischen lag konnte nun genau an einer gewissen Schrittstelle aufgenommen werden. Es stellte sich weiterhin dabei heraus, dass es auch nicht schlimm wäre wenn der eine oder andere Punkt

dadurch verloren geht. Erst bei größeren Löchern ist ein Nachrechnen wie das Loch beschaffen ist schwer.

5.2.6. Dokumentation des Laser-Distanzsensor

Ein großes Problem machte während der Implementation der Kommunikation zum Distanzsensor die Dokumentation des Herstellers. Schnell wurde klar das eine spezielle Lösung dafür entwickelt werden musste, da die Befehle und Fehlercodes selten ein gleiches Muster aufwiesen und somit ein einheitliches Ansprechen der verschiedenen Funktionen unmöglich war.

Stattdessen mussten die einzelnen Funktionen des Laser-Distanzsensors gekapselt werden und durch ein einfaches Java Interface ansprechbar gemacht werden. Somit muss der Entwickler, der die Funktionen benutzten möchte sich nicht darum kümmern welche und wie viele Byte zum Scanner geschickt werden sollen sondern einfach nur die Funktionen aufrufen.

6. Schluss

In diesem Teil der Arbeit wird der praktische Teil noch einmal zusammengefasst und es wird ein Ausblick gegeben, auf mögliche Erweiterungen die an dem Scanner durchgeführt werden können.

6.1. Zusammenfassung und Fazit

Die Problemstellung, welche sich daraus ergibt einen 3D-Scanner mittels eines Laser-Distanzsensor zu entwickeln, wurde am Anfang dieser Arbeit erläutert. In Anbetracht dieses Problems wurde im Labor der HAW Hamburg ein 3D-Scanner mittels eines Laser-Distanzsensors und eine Software die es möglich macht, den neu konstruierten Scanner zu Steuern entwickelt. Nachdem die Vorgehensweise erläutert wurde, wurde die schrittweise Implementierung erläutert.

Eine Herausforderung bestand darin, die genauen Messungen am Objekt zu ermöglichen. So mussten Schnittstellen, zur Hardware, geschaffen werden mit denen es möglich ist diese genauen Messungen durchzuführen. Eine zweite Herausforderung bestand darin, das aufgenommene Objekt nach einem Scanvorgang als Punktwolke richtig darzustellen. Da die korrekte Vorgehensweise eines Scanvorgangs nur Sinn ergibt, wenn diese direkt an der Hardware getestet werden kann, musste ein Großteil dieser Arbeit im Labor entwickelt und getestet werden. Durch mehrere Hardware Bestellungen innerhalb der Bachelorarbeit, war es nicht möglich durchgängig am Scanner zu arbeiten und es konnten nur Implementationen vorgenommen werden die das Testen an der Hardware nicht benötigten, wodurch das Vorankommen erschwert wurde.

Um zu testen, ob der in dieser Arbeit entwickelte 3D Scanner funktioniert, wurde die Oberfläche einer Gummiente und einer Dose aufgenommen. An dem Abbild der Gummiente sind sehr schnell Stauchungen oder Streckungen zu sehen, sollte etwas beim Scanvorgang nicht funktioniert haben. Die Dose wurde dafür benutzt um zu sehen, in welche Richtung die Stauchung und Streckung einsetzten um einen Algorithmus zu entwickeln der diese Probleme wieder ausgleicht.

6.2. Ausblick

In diesem Kapitel der Bachelorarbeit werden Zukunftsvisionen vorgestellt in der die Hardware oder die Software noch zu erweitern wäre um bessere Resultate erzielen zu können. Wichtig in dieser Bachelorarbeit war es einen Grundstein zu legen auf denen weitere Arbeiten und Projekte aufbauen können.

6.2.1. Aufnehmen von Farben

Da der Laser-Distanzsensor kein Verständnis von Farbe hat und immer nur einen Punkt in einer gewissen Entfernung messen kann, wäre eine denkbare Erweiterung für den Scanner einen Kameraaufsatz zu entwickeln. Durch diese Kamera könnten dann auch die Farben von dem Objekt und den grade gescannten oder als nächstes kommenden Punkt aufgenommen werden. Diese so erhaltenen Farbwerte würden dann jedem Punkt zugewiesen werden um dann später auch ein farbiges Abbild des Objektes anzeigen zu können.

6.2.2. Alle Seiten des Objektes scannen

Da es bekannte Probleme bei dem Scan von der Ober und Unterseite gibt, wäre eine mögliche Erweiterung dem Scanner beizubringen auch die Ober- und Unterseite des Objektes auf zu nehmen. Wenn dies dann geschehen ist kann aus den verschiedenen Scans der verschiedenen Seiten wieder ein einzelnes Objekt erstellt werden. Das Resultat wäre dann ein geschlossenes Objekt in denen auch Löcher erkannt werden könnten.

6.2.3. Druck eines gescannten Objektes

Nachdem das Objekt vollständig aufgenommen wurde befindet es sich im Format einer Punktwolke. Wenn diese Punktwolke nun auf ein Meshnetz welches geschlossen ist berechnet werden kann, ist es möglich die Form das Objektes an einen 3D Drucker zu geben um ein Abbild des eingescannten Objektes auszudrucken. In weiteren Schritten wäre es sogar möglich ein einzelnes Gerät zu entwerfen was erst die Punkte von einem Objekt aufnimmt und dieses dann intern an einen 3D Drucker geben kann um es während des Scanvorgang noch zu drucken.

A. Anleitung zur Benutzung des Scanners

In diesem Anhang wird auf die Benutzung des Scanners eingegangen.

A.1. Einrichten der Scanumgebung

Bevor mit dem Scannen gestartet werden kann, ist es wichtig das die nötige Scanumgebung korrekt eingerichtet wurde. Für den Scanner werden neben den 3d scanner git-Repository auch noch das cg git-Repository benötigt. Eine ausführliche Beschreibung, woher die Repos zu bekommen sind, findet sich in der Dokumentation des Computergrafik Research Dokument (Jenke, 2014a).

Nachdem die Softwareumgebung eingerichtet wurde, kann die Hardware vorbereitet werden. Dafür existieren zwei USB Stecker, einer für die TinkerForge Verbindung und der andere für den Seriellen Adapter zum Laserdistanzsensor.

A.2. Debugging Werkzeuge

Um zu kontrollieren ob die Verbindung zu der Hardware korrekt aufgebaut werden kann, liefern die Hersteller Software, mit der es möglich ist eine Verbindung zur Hardware herzustellen. Zum einen handelt es sich dabei um den BrickViewer (TinkerForge, 2015a) von TinkerForge. Dieser gibt die Möglichkeit Konfigurationen an den Bricks auszuführen. Auch können aktuelle Werte wie etwa die Geschwindigkeit des Motors ausgelesen werden.

Die andere Software ist eine grafische Benutzeroberfläche für den Laser-Distanzsensor. Mit dieser Oberfläche können Messungen durchgeführt werden ohne die Befehle für den Scanner zu kennen. Innerhalb der Bachelorarbeit ist es vorgekommen, dass die Software abgestürzt ist und der Laser weiterhin lief. Dies verhinderte einen Neustart der Software. Durch dieses Tool war es anschließend möglich den Laser auszuschalten und die Software ordnungsgemäß neuzustarten.

A.3. Aufnehmen eines Objektes

Sobald die Hardware ordnungsgemäß läuft, kann mit dem Scannen gestartet werden. Dafür darf kein Objekt auf dem Scanner platziert werden. Sollte der Fehler mit dem zu niedrigen Sockel noch bestehen, muss dieser ein Stück erhöht werden. Danach wird durch die Scanner Software der passende serielle Port ausgewählt und auf İnitialize Scanner"geklickt. Der Scanner sollte nun solange nach unten fahren, bis er den Sockel erreicht hat. Dies ist der neue Nullpunkt für den Scanner.

Anschließend kann auf den Sockel das zu Scannende Objekt platziert werden und die Einstellungen für die Auflösung gewählt werden. Es sollte auch eine Höhe festgelegt werden, da es für den Scanner nicht möglich ist die Höhe des Objektes selbstständig zu erkennen. Nach einem druck auf SScanßollte der Scanner nun von alleine starten und das komplette Objekt aufnehmen.

A.4. Speichern und Laden

Sobald der Scanner einen vollständigen Scan abgeschlossen hat, ist es möglich den letzten Scan in einer Datei zu sichern. Dafür muss ein Name in das SSpeichern-Feldëingegeben werden und durch den Button SSave"bestätigt werden.

Es ist auch möglich ein Objekt was gespeichert wurde wieder zu laden. Dazu muss der Name der Datei die eingelesen werden soll in das "Laden-Feld"geschrieben werden und auf "Load"geklickt werden. Die Punktwolken die wieder eingelesen werden, werden dem root Element im Renderbaum zugeordnet, was es ermöglicht auch mehrere Scans gleichzeitig anzuzeigen.

Abbildungsverzeichnis

Literaturverzeichnis

[JSSC 2015] : *java-simple-serial-connector*. April 2015. – URL https://code.google.com/p/java-simple-serial-connector/

[3D 2015] 3D, Artec: *Maßgefertigte Stützen für Orthopädie und Prothetik*. April 2015. – URL http://www.artec3d.com/de/case_studies/Ma%C3%9Fgefertigte+St%C3%BCtzen+f%C3%BCr+Orthop%C3%A4die+und+Prothetik_3125

[DAVID 2015] DAVID: *DAVID 3D SCANNER*. April 2015. – URL http://www.david-3d.com/de

[Engelmann 2011] Engelmann, Francis: *FabScan Affordable 3D Laser Scanning of Physical Objects*. September 2011. – URL http://hci.rwth-aachen.de/materials/publications/engelmann2011a.pdf

[Explius 2015] Explius: *Explius 3D Scanner [Archäologie]*. April 2015. – URL http://www.explius.de/de/index.php/explius-3d-scanner/explius-3d-scanner-archaeologie/

[xpertgate GmbH & Co. KG 2015] GmbH & Co. KG xpertgate: *Lexikon: Koordinatenmessmaschinen*. 4 2015. – URL http://www.xpertgate.de/produkte/Koordinatenmessmaschinen.html

[gom 2015] gom: *ARGUS - Optische Formänderungsanalyse*. April 2015. – URL http://www.gom.com/de/messsysteme/blechumformung.html

[Jenke 2014a] Jenke, Prof. Dr. P.: *CgResearch - Architekturdokumentation, Komponenten und Konzepte*. Hochschule für Angewandte Wissenschaften (HAW), Hamburg (Veranst.), Dezember 2014

[Jenke 2014b] Jenke, Prof. Dr. P.: *Kalibrierung des 3D Scanners*. Hochschule für Angewandte Wissenschaften (HAW), Hamburg (Veranst.), 2014

[JOGL 2015] JOGL: *JOGL Homepage*. http://jogamp.org/jogl/www/. April 2015. – URL http://jogamp.org/jogl/www/

[TinkerForge 2015a] TinkerForge: *Brick Viewer (brickv)*. April 2015. – URL http://www.tinkerforge.com/de/doc/Software/Brickv.html#brickv

[TinkerForge 2015b] TinkerForge: *Java - API Bindungs*. April 2015. – URL http://www.tinkerforge.com/de/doc/Software/API_Bindings_Java.html

[TinkerForge 2015c] TinkerForge: *Master Brick*. April 2015. – URL http://www.tinkerforge.com/de/doc/Hardware/Bricks/Master_Brick.html#master-brick

[TinkerForge 2015d] TinkerForge: *Stepper Brick*. April 2015. – URL http://www.tinkerforge.com/de/doc/Hardware/Bricks/Stepper_Brick.html#stepper-brick

[TinkerForge 2015e] TinkerForge: *Was ist Tinkerforge?* April 2015. – URL http://www.tinkerforge.com/de/home/what_is_tinkerforge/

[Velleman 2013] Velleman: *Datasheet*. Velleman (Veranst.), November 2013

[Velleman 2015] Velleman: *K8200*. April 2015. – URL http://www.k8200.eu/specs/

[Vierling 2013] Vierling, Florian: *Practical 3D | 3D-Praxis (3D-Scanner)*. 2013

[Welotec a] Welotec: *OWTBC Datasheet*. Welotec (Veranst.)

[Welotec b] Welotec: *Technical Manual OWTC / OWTB V2*. Welotec (Veranst.)

Hiermit versichere ich, dass ich die vorliegende Arbeit ohne fremde Hilfe selbständig verfasst und nur die angegebenen Hilfsmittel benutzt habe.

Hamburg, 21. April 2015 Joschka Schulz