Feodor-Lynen-Gymnasium
Planegg

Abiturjahrgang 2016

S E M I N A R A R B E I T

Rahmenthema des Wissenschaftspropädeutischen Seminars:

Physik der Superhelden

Leitfach: *Physik*

Thema der Arbeit:

3D-Wahrnehmung von Daredevil, aktuelle Techniken zur 3D-Vermessung und Bau eines 3D-Scanners

Verfasser/in:
Felix Wimbauer

Kursleiter/in:
Ralph Zierke

Abgabetermin:

10. November 2015

Bewertung	Note	Notenstufe in Worten	Punkte		Punkte
schriftliche Arbeit				x 3	
Abschlusspräsentation				x 1	
				Summe:	
				Gesamtleistung nach § 61 (7) GSO = Summe:2 (gerundet)	

Datum und Unterschrift der Kursleiterin bzw. des Kursleiters

Inhaltsverzeichnis

1. Einleitung

Matt Murdock alias „Daredevil“ ist eine Superhelden-Figur aus der gleichnamigen Comic-Serie des Marvel-Verlags. Daredevil ist ein eher unkonventioneller Superheld. Als der angehende Richter einen Mann vor einem entgegenkommenden Lastwagen rettet, kommt er dabei mit dessen radioaktiver Fracht in Kontakt. Matt verliert sein Augenlicht, doch seine übrigen Sinne werden ins Übermenschliche gesteigert und er kann die Umgebung räumlich, d.h. in drei Dimensionen, durch einen Radarsinn wahrnehmen [1]. Im Zentrum dieser Arbeit stehen der Vergleich dieser 3D-Wahrnehmung mit den heute üblichen Techniken des 3D-Scannens und der Bau eines 3D-Scanners.

Zunächst soll hierzu in Kapitel 2 Daredevils Fähigkeit zur dreidimensionalen Wahrnehmung näher untersucht werden.

Kapitel 3 wird sich dann mit den aktuellen technischen Verfahren zur 3D-Wahrnehmung befassen, auch genannt 3D-Scanning, und diese jeweils kurz vorstellen. Hierbei kann man die 3D-Scanner zunächst grob in 3D-Scanner mit und ohne Kontakt zum Objekt einteilen. Bei den kontaktlosen Scannern wird wiederum zwischen der Time-of-flight-Methode und verschiedenen trigonometrischen Verfahren unterschieden. In diesem Kapitel wird dann auch die Frage gestellt, welches technische Verfahren zum 3D-Scannen Daredevils Fähigkeit zur 3D-Wahrnehmung am ehesten entspricht.

Den praktischen Hauptteil der Arbeit in Kapitel 4 bildet die eigenständige Entwicklung eines 3D-Scanners. Die Mechanik des Scanners wird dabei aus Fischertechnik aufgebaut. Die Steuerung und Signalauswertung des Scanners erfolgt über eine Arduino-Plattform, die in C programmiert wird. Zusätzlich wird die Hauptsteuerung über den PC in Java programmiert. Bei dem 3D-Scanner handelt es sich um einen Time-of-flight Scanner. Die Messeinrichtung lässt sich sowohl um die y-, als auch um die z-Achse drehen. Im Betrieb tastet der Scanner den Raum ab und sendet die Entfernungsdaten mit den zugehörigen Rotationswinkeln an einen verbundenen Computer. Dieser verarbeitet dann die Messwerte und erstellt eine dreidimensionale Punktwolke des Raumes. In der Arbeit werden auch zwei unterschiedliche Sensoren (Ultraschall, LIDAR) getestet und auf ihre Eignung für den Scanner überprüft. Einen wichtigen Teil der Arbeit nimmt die graphische Darstellung und Aufarbeitung der Punktwolke ein. Auch eine einfache Form der Fehlerkorrektur wird implementiert. Ein Fazit mit einem Ausblick schließt die Arbeit ab.

2. Daredevils Radar-Sinn

Im Jahr 1964 riefen der legendäre Comic-Autor Stan Lee und der Zeichner Bill Everett die Figur „Daredevil“ ins Leben. Die Serie hatte jedoch erst dank des neuen Zeichenstils von Frank Miller in der 1980er Jahren Erfolg.

Daredevil heißt mit „bürgerlichem“ Namen Matt Murdock. Als Kind war Matt ein fleißiger und guter Schüler, wurde allerdings deshalb auch von den anderen Kindern gehänselt. Um Selbstvertrauen zu gewinnen und auch um sich zu wehren, begann Matt seine Fitness zu trainieren.

Als Matt eines Tages einen älteren Mann vor einem Truck rettet, wird er angefahren und mit einer radioaktiven Masse übergossen. Er verliert zwar sein Augenlicht, doch seine übrigen Sinne werden deutlich geschärft und er erhält die Fähigkeit, die Umgebung dreidimensional durch einen Radar-Sinn wahrzunehmen.

In der darauffolgenden Zeit lernt Matt von einem Ninja Meister, seine Fähigkeiten richtig einzusetzen und wird in den Martial Arts Kampfkünsten geschult. Diese setzt er zum ersten Mal gegen die Bande eines Straßengangsters ein, nachdem diese seinen Vater ermordet hat.

Daredevil besitzt also keine wirklich „übernatürlichen“ Fähigkeiten wie z.B. das Fliegen von Superman, mit Ausnahme des Radar-Sinns [1].

Wie wird der Radar-Sinn nun im Detail im Comic dargestellt?

Abbildung 1: Erklärung des Radar-Sinns in Daredevil #1 von Mark Waid [3]

In Daredevil #1 vom Autor Mark Waid erklärt Matt Murdock seinen Radar-Sinn so (siehe Abbildung 1):

Begleiter: „Du hast gesagt es wäre so wie ein Echolot.“
Matt Murdock: „So als ob mein Gehirn die Entfernung der Umgebung durchgehend in alle Richtungen erfasst. Es fühlt sich wie abtasten an.“

Dies erklärt allerdings nur sehr wenig die „naturwissenschaftlichen Grundlagen“ des Radar-Sinns. Die Formulierung legt nahe, dass der 3D-Sinn von Daredevil ähnlich wie bei einem Echolot oder einer Fledermaus über eine „Time-of-Flight“-Messung, das heißt über die Laufzeitmessung eines ausgesendeten Signals, das von einem Gegenstand im Raum reflektiert wird, gemessen wird (Ein 3D-Scanner basierend auf einer Time-of-Flight-Messung wird genau in Kapitel 3.2.1 erklärt.).

Offen bleibt allerdings, ob es sich nun wirklich um ein Radarsignal oder eher ein Schall- bzw. Ultraschall-Signal wie bei einem Echolot handelt. Auch bleibt offen, wie präzise er so die Umgebung wahrnehmen kann.

Zieht man ein Radargerät oder ein Echolot als Vergleichsmaßstab heran, könnten folgende Genauigkeiten in der Ortsauflösung erreicht werden: Bei Radargeräten mit sehr hohen Frequenzen von ca. 8 Ghz könnte Daredevil die Umgebung auf wenige Zentimeter genau wahrnehmen. Mit Ultraschall wäre eine Präzision auf einige Millimeter möglich. Das Beispiel der Fledermaus zeigt zudem, dass Ultraschall zu Orientierungszwecken im Raum grundsätzlich gut geeignet ist [2].

Für beide Verfahren gilt jedoch eine beschränkte Reichweite.

Da der Radarsinn in unterschiedlichen Comics allerdings auch unterschiedlich dargestellt wird, insbesondere was die Präzision angeht, lässt auch dieser Punkt keine genaueren Schlüsse bezüglich der Beschaffenheit des Radar-Sinns zu (siehe Abbildung 2 und 3).

Abbildung 2: Radar-Panel von Paolo Rivera (Daredevil #2) [3]

Abbildung 3: Radar-Panel von Marcos Martín (Daredevil #4) [3]

Hinweise darauf, um welche Art von Wellen es sich handelt, gibt es nochmals am Ende der Hauptgeschichte [3].

Dabei wird Daredevil von Captain America mit „Chaff“ besprüht, was seinen Radar-Sinn einschränkt.

Bei „Chaff“ handelt es sich um eine Wolke aus dünnem Aluminium, Plastik und mit Metall versehenem Fiberglas. Im zweiten Weltkrieg entwickelt, führt „Chaff“ zu Messfehlern bei Radaren [4].

Selbst wenn dies auch ein Echolot ein wenig beeinträchtigen würde, spricht diese Darstellung doch für die Verwendung elektromagnetischer Wellen.

Zudem spricht Mark Waid in einem Interview davon, dass es sich beim Radar-Sinn um eine Form von Radarwellen handelt, die sich in alle Richtungen ausbreiten.

Im selben Interview sagt Waid, dass Daredevil durch diese Fähigkeit nur einen vagen, silhouettenhaften Eindruck seiner Umgebung bekommt [5].

Zusammenfassend lässt sich sagen, dass die Darstellungen im Comic keinen eindeutigen Hinweis zum tatsächlichen naturwissenschaftlichen Mechanismus, der dem Radar-Sinn zu Grunde liegen, geben, am ehesten handelt es sich jedoch um ein Time-of-flight Verfahren mit elektromagnetischen Wellen.

3. Verfahren zum 3D-Scannen

Im folgenden Abschnitt sollen verschiedene Techniken des 3D-Scannens dargestellt werden, die der 3D-Wahrnehmung von Daredevil unterschiedlich nahe kommen. In der technischen Praxis hängt die Frage, welche 3D-Scan-Technik am besten zum Einsatz kommt, stark vom jeweiligen Anwendungsfall (z.B. Erfassen der dreidimensionalen Form eines Objekt oder Erfassen eines Raums), der erforderlichen Genauigkeit und der angestrebten Scann-Geschwindigkeit ab. Ergebnis des Scanvorgangs ist unabhängig von der verwendeten Technik in der Regel eine so genannte Punktwolke, die die Oberflächenkoordinaten des gescannten Objekts oder Raums, ggf. ergänzt um Oberflächeninformationen (z.B. Farbe, Textur), angibt.

Die unterschiedlichen Methoden lassen sich zunächst in 2 Gruppen aufteilen: Scanner die beim Scanvorgang das Objekt berühren (Contact-Scanner) und Scanner die das Objekt beim Scannen nicht berühren (Non-Contact-Scanner) [6].

3.1 Contact-Scanner

Bei Contact-Scannern wird das zu scannende Objekt im wahrsten Sinne des Wortes abgetastet. Dabei bewegt sich ein Berührungssensor an einem flexiblen Arm über das Objekt. Der flexible Arm ist ebenfalls mit Sensoren ausgestattet, so dass die 3D-Position des Berührungssensors ausgelesen werden kann. Das Objekt selbst ist auf einem Tisch oder Gestell fixiert. Der flexible Arm mit dem Berührungssensor kann automatisch oder per Hand bewegt werden. Mit Contact-Scannern kann eine sehr hohe Genauigkeit erreicht werde, die Abtastgeschwindigkeit ist jedoch in der Regel deutlich langsamer als bei Non-Contact-Scannern. Typische Anwendungsfälle von Contact-Scannern sind die Erfassung von Bauteilen für ein CAD-Modell im Industriebereich oder die Erstellung eines Datenmodells einer Figur für einen Animationsfilm [8].

Abbildung 4: Contact Scanner [7]

3.2 Non-Contact-Scanner

Unter den Non-Contact Scannern werden hier nur sogenannte aktive Scanner betrachtet. Dabei emittiert der Scanner eine Form von Strahlung, die vom zu scannenden Objekt reflektiert wird. Aus den Eigenschaften der zurückgeworfenen Strahlung können dann die x, y und z Koordinate des Objektpunktes ermittelt werden. Als Strahlung können Licht, Schall oder auch Röntgenstrahlung genutzt werden. Damit dürften aktive Non-Contact-Scanner auch der von Daredevil verwendeten Technik am nächsten kommen.

Im Folgenden sollen nun zwei Arten von aktiven Non-Contact-Scan-Techniken vorgestellt werden, die Flug-Zeit-Methode und die Triangulationsmethode.

3.2.1 Flug-Zeit-Methode

Bei der Flug-Zeit-Methode misst der Scanner die Zeit, die zwischen dem Aussenden eines Strahlungspulses und der Ankunftszeit des reflektierten Strahlungspulses vergeht. Am verbreitetsten sind gepulste Laser als Strahlungsquelle (LIDAR (Light Detection and Ranging)-Scanner), sonstige elektromagnetische Wellen (z.B. Radiowellen im Radar), oder auch Schallwellen (Sonar).

Die Entfernung l des Gegenstandes von der Strahlungsquelle lässt sich wie folgt ermitteln:

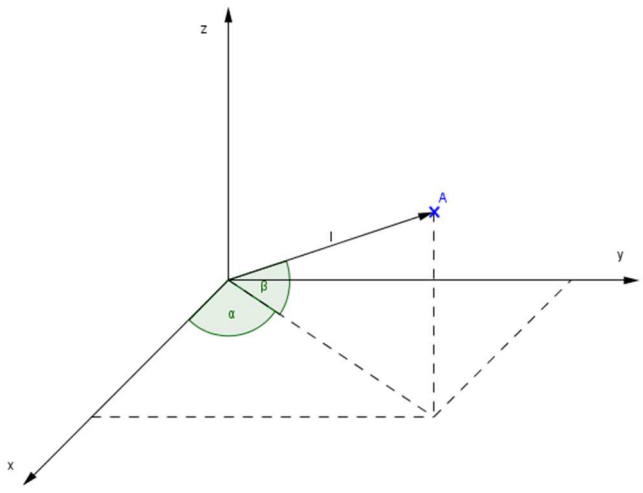
$$v = \frac{2\,l}{\Delta t}$$

$$l = \frac{\Delta t \cdot v}{2}$$

Die Größe v gibt dabei die Strahlungsgeschwindigkeit an. Im Falle eines Lasers als Strahlungsquelle handelt es sich um die Lichtgeschwindigkeit c . Die Größe Δt beschreibt die Zeit des Strahlungspulses von der Strahlungsquelle und wieder zurück.

Die x, y und z Koordinate können dann aus der Ausrichtung, d.h. dem horizontalen und vertikalen Winkel der Strahlungsquelle und der Entfernung l abgeleitet werden.

Die 3d-Koordinate wird dabei folgendermaßen berechnet (siehe Abbildung 5):



$\boldsymbol{l}$ entspricht der durch das Messgerät gemessenen Entfernung,

$\boldsymbol{\alpha}$ und $\boldsymbol{\beta}$ der Ausrichtung des Messgeräts.

$$A = (x_A|y_A|z_A)$$

$$x_a = \cos\alpha \cdot \cos\beta \cdot l$$

$$y_A = \sin\alpha \cdot \cos\beta \cdot l$$

$$z_A = \sin\beta \cdot l$$

Abbildung 5: Berechnung einer 3d-Koordinate anhand einer Seitenlänge und zwei Winkeln (mit Geogebra [22] erstellt)

Das Objekt bzw. der Raum wird nun abgescannt, indem die Ausrichtung der Strahlungsquelle horizontal und vertikal abgeändert wird. Die Winkeländerung kann durch Bewegung der kompletten Strahlungsquelle oder auch durch Bewegung eines Spiegels erfolgen. Abbildung 6 zeigt als Beispiel den Laserscanner C10 der Firma Leica.

Typische Anwendungsfälle von Laserscannern mit Flug-Zeit-Methoden sind Vermessung von Räumen in Gebäuden oder Vermessungsaufgaben im Freien. Möglich sind große Reichweiten (bis ca. 300 m), bei beschränkter Messgenauigkeit (im Millimeter- bzw. Zentimeter-Bereich) und relativ hohe Messgeschwindigkeiten (ca. 10.000 Messpunkte/s).

Der im praktischen Teil aufgebaute Laserscanner wird die Flug-Zeit-Methode nutzen, wobei der Scan-Kopf sowohl horizontal als auch vertikal gedreht wird.

Abbildung 6: Leica Laserscanner C10 [9]

3.2.2 Triangulations-Methode

Bei der Triangulationsmethode besteht der Scanner-Kopf aus einer Strahlungsquelle, in der Regel einem Laser, und einem seitlich versetzten Sensor, üblicherweise einem CCD-Sensor, wie er z.B. auch in Digitalkameras verbaut wird (siehe Abbildung 7). Die Entfernung des Objekts kann dabei aus dem Abstand zwischen Laser und Sensor sowie dem Winkel des auf den Sensor einfallenden reflektierten Lichts ermittelt werden. Praktisch wird der Winkel über eine Optik als Auslenkung auf dem CCD-Sensor gemessen.

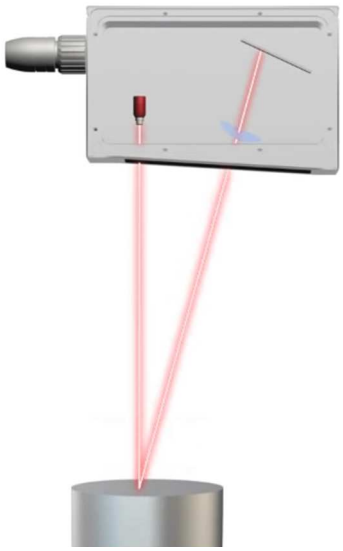


Abbildung 7: Aufbau eines Laserabstandsmessgeräts [10]

Ziel ist es, die Entfernung $\boldsymbol{x}$ vom Laser zum Objekt zu berechnen (siehe Abbildung 8).

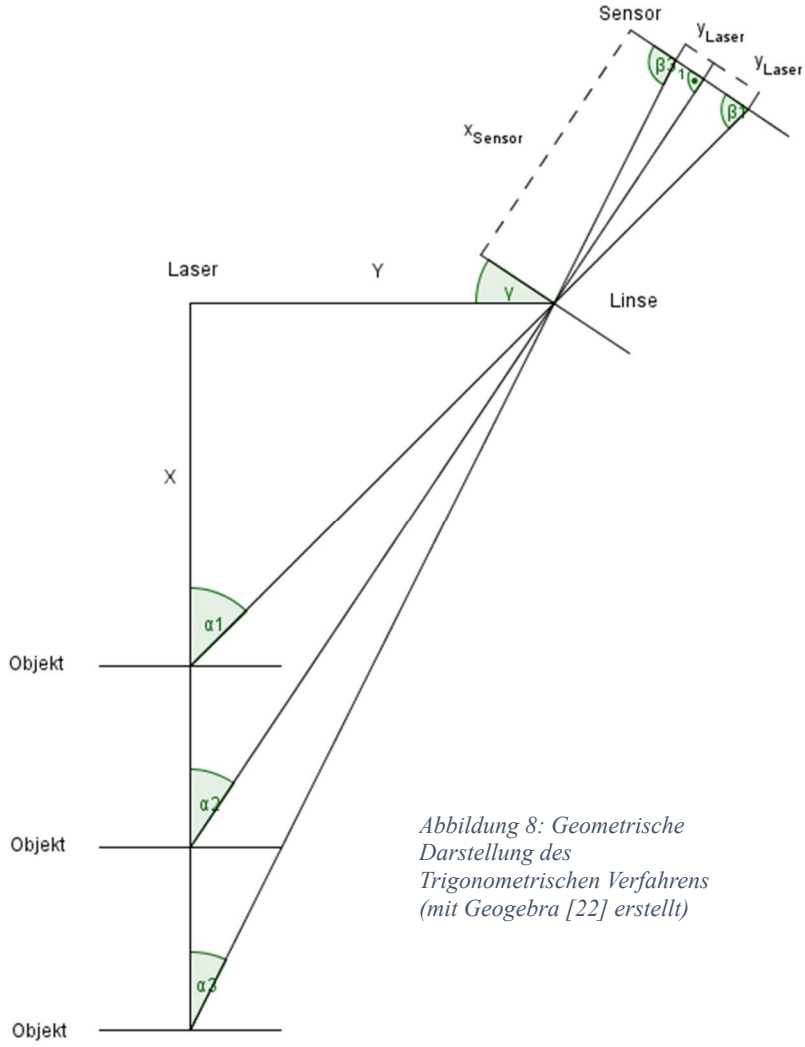


Abbildung 8: Geometrische Darstellung des Trigonometrischen Verfahrens (mit Geogebra [22] erstellt)

Dabei ist die Entfernung $\boldsymbol{y}$ zwischen dem Laser und der Linse, der Neigungswinkel $\boldsymbol{\gamma}$ des Sensors, der Abstand $\boldsymbol{x_{Sensor}}$ zwischen der Linse und dem Sensor, sowie der Abstand $\boldsymbol{y_{Laser}}$ des Auftreffpunktes des Lasers auf dem Sensor vom Mittelpunkt des Sensors (links = negativ) bekannt.

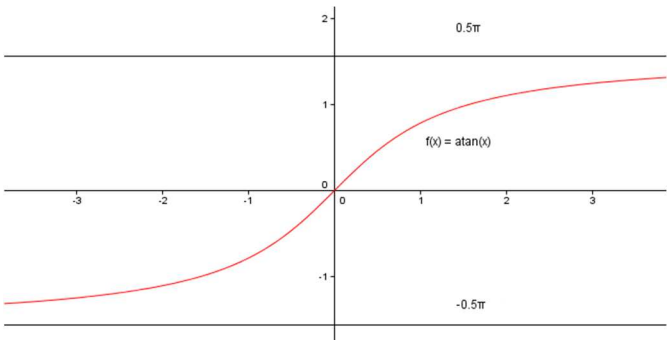


Abbildung 9: Graph der atan(x) Funktion (mit Geogebra [21] erstellt)

Nun muss man zunächst den Auftreffwinkel $\boldsymbol{\beta}$ des Laserstrahls berechnen.

Hierbei ist es jedoch nötig, eine Fallunterscheidung anzuwenden, da man nur den Innenwinkel berechnen kann.

Der Innenwinkel entspricht nur bei positiven Werten von $\boldsymbol{y_{Laser}}$ dem gesuchten Winkel $\boldsymbol{\beta}$ (siehe Abbildung 9).

Im Falle von $\boldsymbol{y_{Laser}} = 0$ ist die Rechnung nicht lösbar, da $\boldsymbol{y_{Laser}}$ der Nenner ist. Der Winkel hat den Wert 90°.
Im Falle von $\boldsymbol{y_{Laser}} < 0$ entspricht $\boldsymbol{\beta}$ dann aber nicht mehr dem Innenwinkel des Dreiecks, sondern dem Außenwinkel. Diesen kann man durch $\boldsymbol{180° - Innenwinkel}$ berechnen. Der Innenwinkel ist dabei vom Betrag her gleich dem Winkel bei $\boldsymbol{-y_{Laser}}$.

Daraus ergibt sich folgende Fallunterscheidung für die Berechnung von $\boldsymbol{\beta}$.

$$\beta = \begin{cases} 180° - \tan^{-1}\left(\frac{x_{Sensor}}{-y_{Laser}}\right), & y_{Laser} < 0 \\ 90°, & y_{Laser} = 0 \\ \tan^{-1}\left(\frac{x_{Sensor}}{y_{Laser}}\right), & y_{Laser} > 0 \end{cases}$$

Anschließend kann man mit Hilfe des Neigungswinkels $\boldsymbol{\gamma}$ des Sensors $\boldsymbol{\alpha}$ berechnen:

$$\alpha = 90° - (\beta - \gamma)$$

Schließlich ist es möglich, über Triangulation die Entfernung $\boldsymbol{x}$ zu berechnen:

$$x = \frac{y}{\tan \alpha}$$

Die Abtastung des Objekts erfolgt nun dadurch, dass der Scanner-Kopf horizontal und vertikal geschwenkt oder das Objekt rotiert wird.

Die Messgeschwindigkeit der Triangulations-Methode kann dadurch deutlich erhöht werden, dass nicht nur einzelne Punkte auf ein Objekt projiziert werden, sondern Streifen. Man spricht dann von 3D-Laser-Scannen. Je nach Form des Objekts weicht dann der auf den CCD-Sensor abgebildete Streifen von einer Linie ab, wodurch wiederum der Abstand der einzelnen Punkte auf diesem Lichtstreifen mit Hilfe der Triangulationsmethode ermittelt werden kann. Indem der Laserstreifen über das Objekt geführt wird, gewöhnlich durch Drehen des Objekts, kann so die Gesamtform des Objekts bestimmt werden.

Die Triangulations-Methode ist typischerweise für kürzere Entfernungen geeignet (wenige Meter), erreicht jedoch deutlich genauere Messergebnisse (Größenordnung Mikrometer) im Vergleich zur Flug-Zeit-Methode [11].

4. Bau eines 3D-Scanners

„Inspiriert“ von der 3D-Wahrnehmung von Daredevil soll in diesem Kapitel beschrieben werden, wie ein 3D-Scanner mit einfachen Mitteln selbst entwickelt, aufgebaut und programmiert werden kann. Es wird gezeigt, dass so recht beachtliche Messergebnisse erzielt werden können.

Der aufgebaute Scanner soll nach der im vorherigen Kapitel beschriebenen Flugzeit-Methode arbeiten. Das Kernstück des 3D-Scanners bildet ein Flug-Zeit-Abstandssensor, der mechanisch um zwei Achsen gedreht wird. Im ersten Schritt des Projektes wurden zwei Sensoren auf ihre Eignung für den Scanner getestet, nämlich ein Ultraschallsensor und ein LIDAR-Sensor.

4.1 Vergleich zweier Abstandssensoren Ultraschallsensor HC-SR04 vs. LIDAR-Sensor LIDAR-Lite

Der Abstandssensor ist das wichtigste Bauteil des Scanners. Als mögliche Kandidaten für den Abstandssensor wurden zwei im Handel erhältliche und relativ preisgünstige Abstandssensoren ausgewählt, nämlich der Ultraschallsensor HC-SR04, der oft zusammen mit Arduino-Bausätzen angeboten wird, und zum anderen der Laserentfernungsmesser LIDAR-Lite der Firma PulsedLight3D, der z.B. für die Abstandsmessung in Modellbaudrohnen verwendet wird. Mit verschiedenen Messreihen wurde überprüft, welcher der Sensoren besser für den geplanten Scanner geeignet ist.

4.1.1 Funktionsweise

Beide Abstandssensoren arbeiten, wie ausgeführt, nach der Flugzeitmethode, d. h. ein Signal wird vom Sensor ausgesandt, die Laufzeit des von einem Gegenstand reflektierten Signals wird gemessen und so der Abstand ermittelt. Beim HC-SR04 wird hierzu ein Ultraschallsignal verwendet. Lt. Herstellerangabe beträgt die maximale Reichweite ca. 3 m und die erreichbare Messgenauigkeit beläuft sich auf ca. 1 cm [12].

Beim LIDAR-Lite Sensor wird stattdessen von einem Laser ein Lichtimpuls ausgesandt. Der LIDAR-Sensor hat lt. Angaben der Firma PulsedLight3D eine maximale Reichweite von 40 m und misst auf ca. 2 cm genau. Diese für einen LIDAR-Sensor relativ hohe Messgenauigkeit wird lt. Beschreibung des Herstellers durch eine so genannte Korrelationsmethode erreicht. Dabei wird die Pulsform des ausgesandeten Lichtimpulses gespeichert und dann mit dem reflektierten Lichtimpuls verglichen (korreliert). Dadurch soll vermieden werden, dass sonstiges Streulicht versehentlich als reflektiertes Licht detektiert wird. Die Details der vom Sensor verwendeten Korrelationsmethode werden auf der Web-Seite des Herstellers beschrieben [13].

4.1.2 Messungen zur Überprüfung der Eignung der Abstandssensoren

Zur Ermittlung des am besten geeigneten Sensors wurden nun folgende Messreihen jeweils unter gleichen Bedingungen für beide Sensoren durchgeführt:
Zunächst wurde die Genauigkeit der Sensoren bei kurzer Distanz getestet. Die beiden Sensoren wurden dabei in einem Meter Entfernung zur Wand aufgestellt und 100 Werte gemessen.

Der LIDAR-Sensor misst durchschnittlich 92,32 cm, der Ultraschallsensor 98,34 cm.

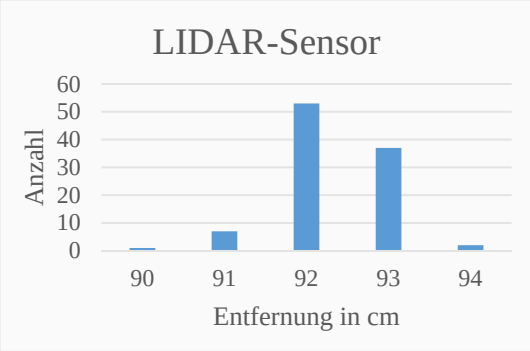


Abbildung 10: LIDAR-Sensor Messung: Durchschnittswert 92,32 cm Standardabweichung 0,68 cm

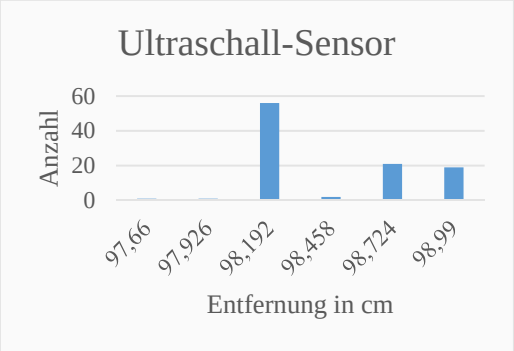


Abbildung 11: Ultraschall-Sensor Messung: Durchschnittswert 98,34 cm Standardabweichung 0,34 cm

Auffällig ist dabei, dass der LIDAR-Sensor meist 7 cm – 8 cm zu wenig misst. Dies könnte an der Tatsache liegen, dass der Sensor im Vergleich zum Ultraschall-Sensor relativ lang (4 cm vs. 1,3 cm) ist und man den Punkt, von dem die Entfernung gemessen wird, nicht genau bestimmen kann. Zudem ist der LIDAR-Sensor mit 40 m maximaler Reichweite eher auf größere Distanzen ausgelegt. Die Messgenauigkeit des LIDAR-Sensors ist, wie gemäß den Herstellerangaben zu erwarten, etwas geringer als die Messgenauigkeit des Ultraschallsensors, was sich an der etwa doppelt so großen Standardabweichung[1] der Messreihe für den LIDAR-Sensor im Vergleich zum Ultraschall-Sensor ablesen lässt.

Der Ultraschall-Sensor scheint also zumindest für kurze Distanzen zunächst besser für die Verwendung im 3D-Scanner geeignet, da er die Entfernung genauer bestimmen konnte.

[1] Die Standardabweichung $\sigma(X)$ einer Zufallsvariablen X ist definiert als die Quadratwurzel der Varianz $Var(X)$: $\sigma(X) = \sqrt{Var(X)}$. Dabei ist die Varianz $Var(X) = E((X - E(X))^2)$. Das Symbol $E()$bezeichnet den Erwartungswert.

Beim nächsten Test wurden die Sensoren vier Meter von der Wand entfernt aufgestellt. Ebenfalls wurden wieder jeweils 100 Messungen durchgeführt.

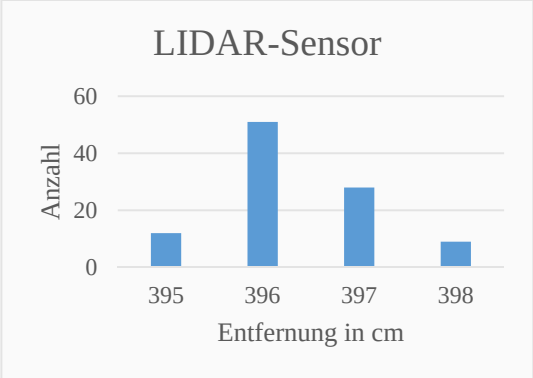


Abbildung 13: LIDAR-Sensor Messung: Durchschnittswert 396,34 cm Standardabweichung 0,80 cm

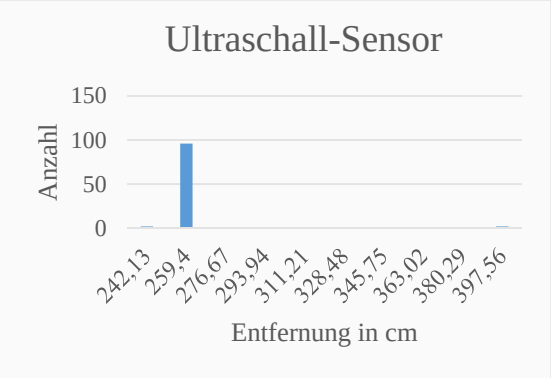


Abbildung 12: Ultraschall-Sensor Messung: Durchschnittswert 247,11 cm Standardabweichung 21,41 cm

Der LIDAR-Sensor liefert auch bei diesem Test mit durchschnittlich 396,34 cm und einer Standardabweichung von 0,8 cm brauchbare Werte. Deutlich wird aber, dass der Durchschnittswert um ca. 2 bis 4 cm entsprechend der Länge des Sensors (s.o.) korrigiert werden sollte.

Unerwarteter Weise liefert der Ultraschall jedoch einen Durchschnittswert von 256,4 cm, was ca. 40% neben dem richtigen Ergebnis liegt. Auffallend ist auch, dass die Standardabweichung mit 21,41cm sehr hoch ist, da der Scanner in jeweils zwei Fällen ca. 240 cm und ca. 400 cm gemessen hat. Eine naheliegende Erklärung ist, dass sich die Schallwellen im Raum deutlich weniger fokussiert ausbreiten als die Laserstrahlen des LIDAR-Sensors. Da beide Sensoren ca. 10 cm über dem Boden standen, ist es wahrscheinlich, dass der Ultraschallsensor im Wesentlichen ein in ca. 2,4 m Entfernung vom Boden reflektiertes Signal erfasst hat und einmal ein Signal, das von der Wand reflektiert wurde.

Um dieses Phänomen näher zu untersuchen, wurden die beiden Sensoren in einer Entfernung von drei Metern zur Wand aufgestellt. Zudem wurde in zwei Metern Entfernung ein Hindernis aufgestellt. Dieses Hindernis befand sich 5 cm neben der Ausrichtung der Sensoren.

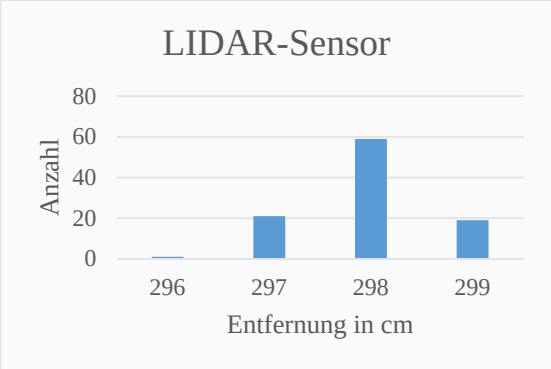


Abbildung 15: LIDAR-Sensor Messung: Durchschnittswert 297,96 cm Standardabweichung 0,66 cm

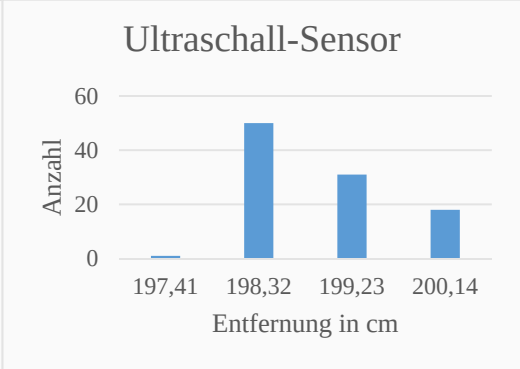


Abbildung 14: Ultraschall-Sensor Messung: Durchschnittswert 198,38 cm Standardabweichung 0,78 cm

Während der LIDAR-Sensor wieder das richtige Ergebnis lieferte, erfasst der Ultraschall-Sensor ausschließlich das Hindernis in 2 m Entfernung.

4.1.3 Fazit

Der Ultraschallsensor misst zwar bei großen Flächen wie einer Wand und kurzen Abständen die Entfernung sehr präzise, allerdings kommt es aufgrund der geringen räumlichen Fokussierung der Schallwellen und der Reflektion an neben dem Messpunkt liegenden Hindernissen zu einer starken Verfälschung der Messergebnisse. Die räumliche Auflösung des Scanners ist damit sehr schlecht und der Ultraschallsensor für eine Verwendung im Scanner ungeeignet. Im Gegensatz dazu liefert der LIDAR-Sensor zuverlässig relativ präzise Messeergebnisse, die auf Grund der guten Fokussierung des Laser-Strahls räumlich gut aufgelöst sind. Zudem kann der LIDAR-Sensor deutlich größere Entfernungen erfassen. Auf Grund der Untersuchungen wird im Scanner deshalb der LIDAR-Sensor verbaut.

4.2 Aufbau des Scanners

4.2.1 Mechanik des Scanners

Der mechanische Teil des Scanners wird aus Fischertechnik-Bauteilen [14] aufgebaut. Diese sind für ein derartiges Projekt besonders gut geeignet, da die Bauteile aus hartem Kunststoff bestehen und so sehr hohe Präzision ermöglichen. Ähnlich wie bei Lego-Steinen lassen sich diese fast beliebig kombinieren, um komplexe Gebilde zu konstruieren. Der Vorteil gegenüber Lego-Bauteilen besteht darin, dass sich die Fischertechnik-Bauteile auch sehr gut senkrecht etc. zusammensetzen lassen und damit besonders gut für mechanische Konstruktionen geeignet sind.

Die mechanische Konstruktion des Scanners soll eine präzise Drehung des Abstandssensors sowohl um die horizontale als auch die vertikale Achse ermöglichen. Zunächst wird für die Drehung um die vertikale Achse eine Drehscheibe auf eine Kunststoffplatte montiert. Die Scheibe besitzt einen äußeren Zahnkranz und ist auf einer Blockhöhe (1,5 cm) über der Platte angebracht, damit sie gut von einem Motor angetrieben werden kann.

Auf die Drehscheibe werden dann Ausleger zu beiden Seiten angebracht. Diese sind zusätzlich verstrebt, um das Gerüst stabiler zu machen und damit eine hohe Präzision des Scanners zu ermöglichen. Auf die Ausleger wird jeweils senkrecht erhöht wieder eine Drehscheibe für die Drehung um die horizontale Achse angebracht. Eine der Drehscheiben besitzt ebenfalls einen äußeren Zahnkranz, damit man diese mit einem Motor bewegen kann. Zwischen die beiden Drehscheiben wird nun der Scan-Kopf eingebaut (siehe Abbildung 16).

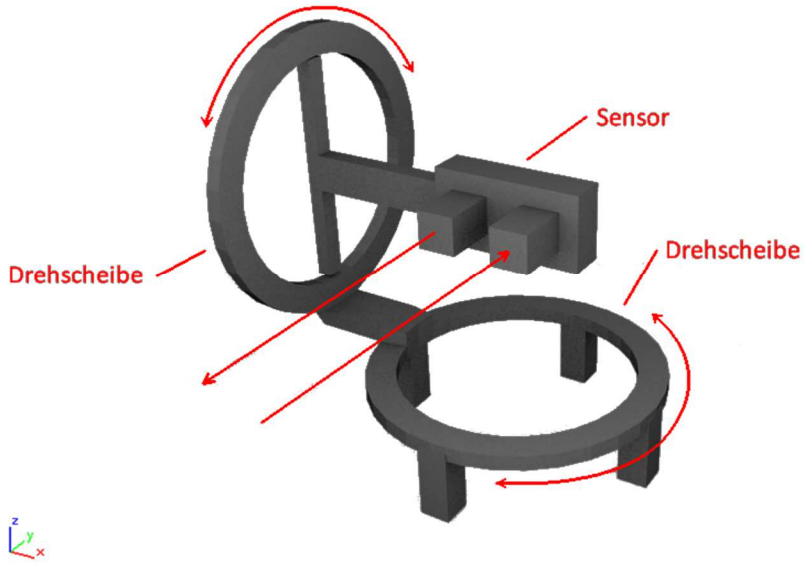


Abbildung 16: Konzeptdarstellung des Aufbaus (mit Blender [23] erstelltes Renderbild, mit Gimp [24] bearbeitet)

Als Antrieb für die Drehung um zwei Achsen werden zwei so genannte Encoder-Motoren von Fischertechnik verwendet (siehe Abbildung 17). Bei diesen Motoren handelt es sich um 9V-Gleichstrommotoren. Deren Besonderheit besteht darin, dass die Motoren einen eingebauten Impulszähler aufweisen, das heißt er sendet pro Umdrehung eine bestimmte Anzahl an elektrischen Impulsen, in diesem Fall 75/Umdrehung. Dies ermöglicht es, den Scanner sehr genau anzusteuern und auszurichten.

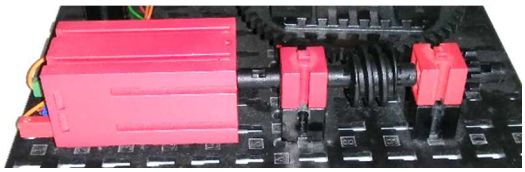


Abbildung 17: Encodermotor von Fischertechnik

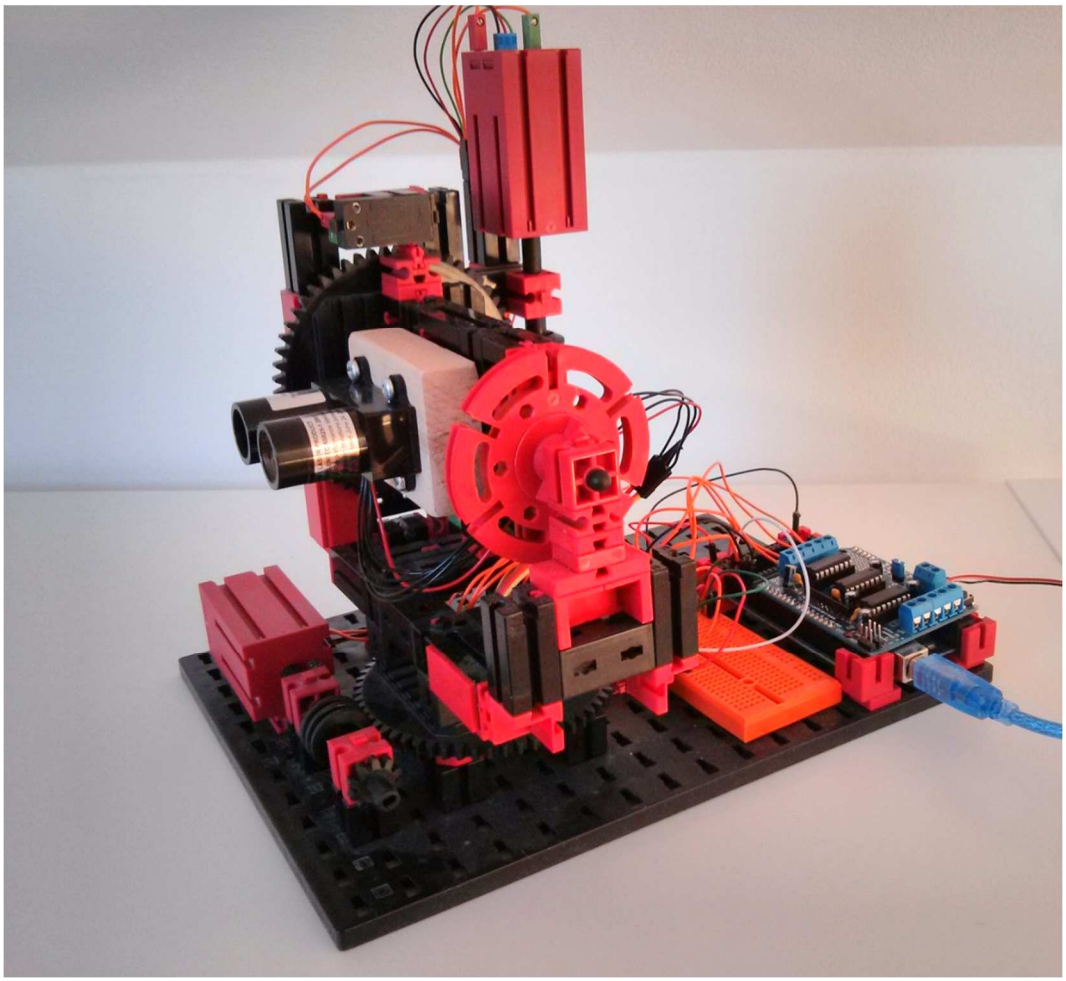


Abbildung 18: Aufgebauter Scanner

Der LIDAR Sensor wurde auf einen Holzblock geschraubt, der dann auf die Fischertechnik-Bauteile zwischen den beiden senkrechten Drehscheiben geklebt wurde. Damit kann der Sensor beliebig vertikal gedreht und horizontal geschwenkt werden.

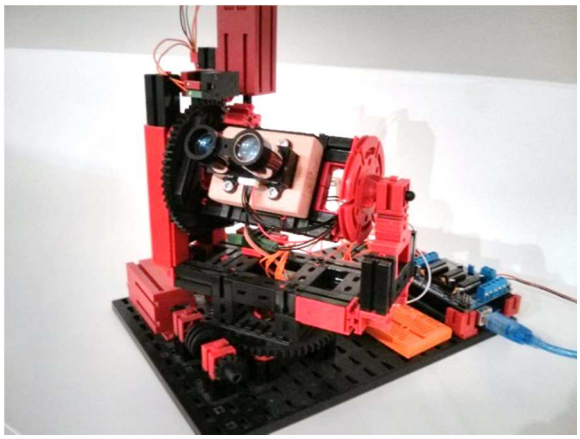


Abbildung 19: Aufgebauter Scanner rotiert

4.2.2 Elektronische Bauteile

Zur Ansteuerung der Motoren und zum Auslesen des Sensors wird ein Arduino Microcontroller Board genutzt. Bei der Arduino Plattform handelt es sich um eine Open Source „Physical-Computing-Plattform“, die für diverse Mess- und Regelungsaufgaben genutzt werden kann. Die Hardware besteht aus einem Board mit dem Prozessor und verschiedenen analogen und digitalen Ein- und Ausgängen. In dieser Arbeit wird das Board Mega 2560 genutzt, das preisgünstig im Handel erhältlich ist (siehe Abbildung 20) [15].

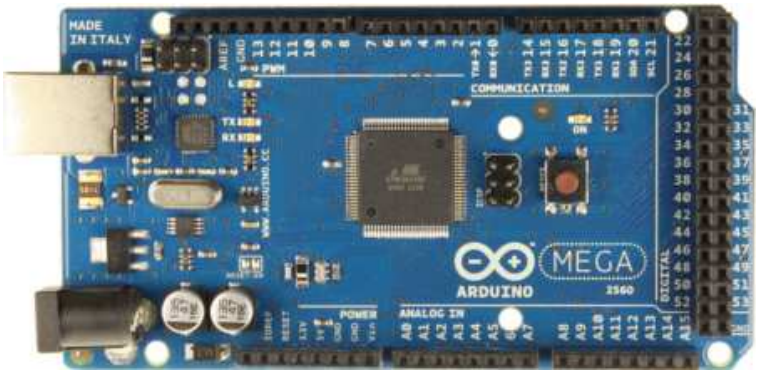


Abbildung 20: Arduino Mega 2560 [16]

Daneben umfasst die Arduino Plattform eine Entwicklungs-Umgebung für die Programmierung der Steuerung. Standardprogrammiersprache ist C bzw. C++.

Weiter lässt sich mit dem Arduino Microcontroller auch eine serielle Verbindung zum Computer über ein USB-Kabel herstellen. Dies bedeutet, dass das Arduino Board und der PC Daten austauschen können und so das Programm auf dem Arduino mit einem Programm auf dem Computer interagieren kann. Über eine Programm-Schnittstelle (s.u.) wird so die wesentliche Steuerung und Signalverarbeitung über ein Java-Programm auf dem PC realisiert.

Da das Arduino Board nur eine sehr geringe Stromstärke über seine Ausgänge ausgibt, benötigt man für die Ansteuerung von Motoren zusätzlich eine externe Stromquelle. Dies wird durch ein sogenanntes Motorshield ermöglicht, das auf das Microcontroller-Board aufgesteckt wird und über Bibliotheken angesteuert werden kann. In diesem Projekt wurde das „SainSmart L293D Motor Drive Shield“ verwendet, das eine Ansteuerung von bis zu vier Gleichstrommotoren ermöglicht [17].

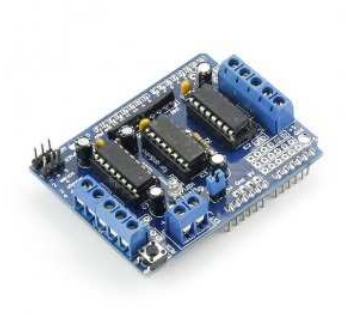


Abbildung 21: SainSmart motor shield [17]

Es wurde auch geprüft, ob statt des Arduino-Boards der Fischertechnik Controller („Robo TX Controller“) verwendet werden soll. Diese Controller-Plattform ist jedoch deutlich weniger leistungsfähig als die Arduino-Plattform. Insbesondere ist es sehr schwierig, Sensoren wie den LIDAR-Lite, die nicht von Fischertechnik stammen, anzusteuern. Auch eine komplexere Steuerung der Motoren, wie sie für den Scanner erforderlich ist, ist mit der Robo TX-Plattform von Fischertechnik kaum möglich. Die Arduino-Plattform erscheint damit für das Scanner-Projekt besser geeignet.

4.3 Scan-Software

Die Scan-Software, die in diesem Kapitel beschrieben wird, dient zur Steuerung der Motoren und zum Auslesen der Messdaten. Die Software zur Auf- und Nachbearbeitung sowie zur Visualisierung der Messergebnisse wird dann im darauf folgenden Kapitel dargestellt.

Folgende Schritte müssen durch die Scan-Software durchlaufen werden:

1. Ausrichtung des Abstandssensors in horizontaler und vertikaler Richtung.
2. Ermittlung der Ausrichtungswinkel α und β in vertikaler und horizontaler Richtung aus der Impulszahl des Encoder-Motors.
3. Auslesen der Entfernung l aus dem Abstandssensor.
4. Ermittlung der der x-, y- und z-Koordinaten gemäß Formeln aus Kapitel 3.2.1.
5. Speicherung als Datenpunkt in der Punktwolke.

Die Schritte 1 bis 4 werden wiederholt durchlaufen, bis der vorgesehene Raum-Winkel-Bereich durch den Scanner abgetastet ist.

Technisch gesehen soll die Scan-Software zwischen dem Arduino-Board und dem PC aufgeteilt werden. Die wesentliche Steuerungslogik soll durch ein Java-Programm auf dem PC erfolgen. Der Arduino soll also nur Anweisungen vom Computer annehmen, diese dann ausführen und dann gegebenenfalls ein Resultat zurücksenden.

So muss die Software auf dem Arduino nicht verändert werden, um verschiedene Aufgaben durchzuführen.

Im Detail wird folgendermaßen vorgegangen:

Das Arduino-Board und der PC stehen über ein USB-Kabel, das heißt einer seriellen Verbindung, in Kontakt. Beide Seiten können in der seriellen Verbindung lesen und schreiben.

Das Arduino Programm kann die Verbindung über den Befehl „**Serial**.begin(9600);“ öffnen.

In einem Java-Programm am PC ist dies mit Hilfe der RXTX-Bibliothek möglich, allerdings deutlich komplizierter [18].

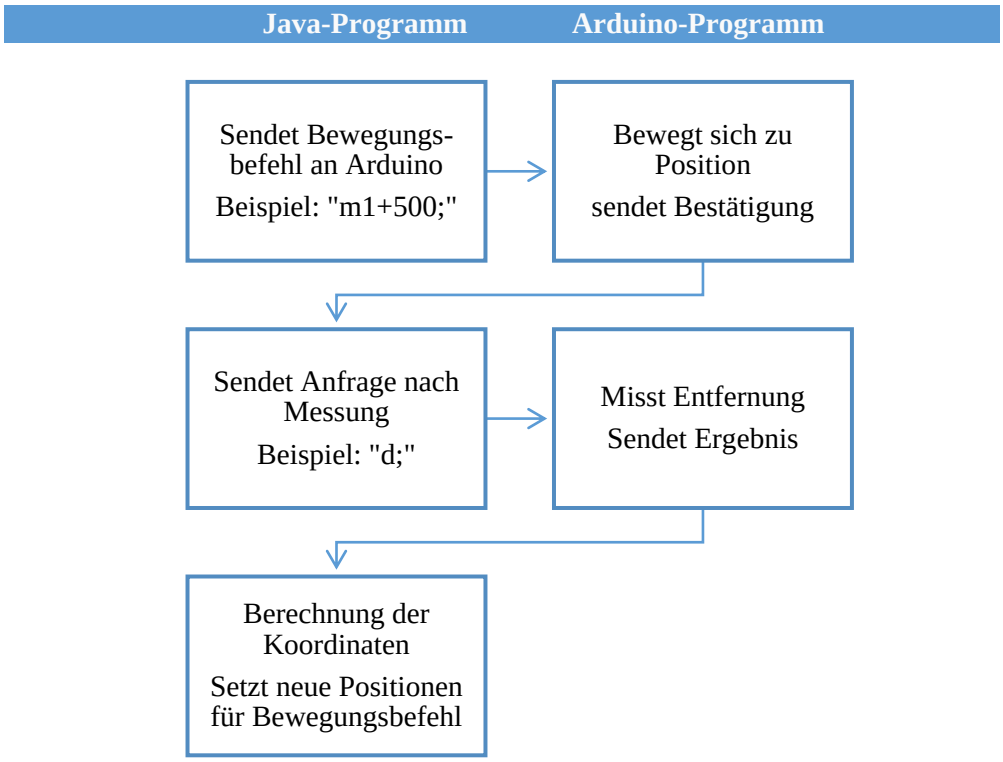
Während des Scanvorgangs schreibt das Java-Programm Befehle in die serielle Verbindung. Das Arduino Programm liest diese aus und schickt dann eine Bestätigung oder das Ergebnis.

```
void recieve()
{
  switch(input.charAt(0))
  {
    case 'd':
    {
      sendDistance();
      break;
    }
    case 'c':
    {
      if(input.charAt(1) == '1')sendMotorCounter1();
      else if(input.charAt(1) == '2')sendMotorCounter2();
      else Serial.println("invalid");
      break;
    }
    case 'm':
    {
      int i = input.substring(3).toInt();
      if(input.charAt(2) == '-')i = -i;
      if(input.charAt(1) == '1')moveTo(1, i);
      else if(input.charAt(1) == '2')moveTo(2, i);
      else Serial.println("invalid");
      break;
    }
    default: Serial.println("invalid");
  }
}
```

Abbildung 22: Lese Methode des Arduino-Boards zum Empfangen von Befehlen durch den PC (im Arduino-Editor [26] umgesetzt)

So ist der Befehl „c1;“ beispielsweise eine Anweisung für das Arduino Board, die aktuelle Position des ersten Motors zurückzugeben. Zunächst wird dabei im Programm die Art des Befehls herausgefiltert. Dafür wird der erste char (char = character(engl.) = Buchstabe, Zahl oder Zeichen beim Programmieren), in diesem Fall „c“, des Befehls, der in der „input“ Variable gespeichert wird, mit der Methode .charAt(0) gesucht und mit den möglichen Befehlsarten in der „switch“ Anweisung verglichen. So weiß das Programm, dass es sich um eine Anfrage durch das Java-Programm des PCs handelt, den Wert eines Motorzählers vom Arduino-Board an den PC zu übermitteln. Anschließend liest das Programm mit der Methode .charAt(1) die zweite Stelle des Befehls aus, um zu erfahren, welcher Motorzähler gefordert ist. In diesem Fall ist der Motorzähler des ersten Motors gefordert.

Das folgende Flussdiagramm zeigt, wie die Schritte zum Abtasten des Raums und zur Berechnung der 3D-Koordinaten für jede Messung zwischen dem Java-Programm auf dem PC und dem Programm auf dem Arduino-Board aufgeteilt sind:



Die Berechnung der Koordinaten erfolgt entsprechend der Formel aus Kapitel 3.2.1 folgendermaßen:
(l = Gemessene Entfernung, angleY = Winkel der Rotation um Y-Achse, angleZ = Winkel der Rotation um Z-Achse)

```
public void calculatePoint(float l, float angleY, float angleZ)
{
	x = (float) (l * Math.cos(Math.toRadians(angleY)) *
	Math.cos(Math.toRadians(angleZ)));
	y = (float) (l * Math.cos(Math.toRadians(angleY)) *
	Math.sin(Math.toRadians(angleZ)));
	z = (float) (l * Math.sin(Math.toRadians(angleY)));
}
```

Abbildung 23: Methode zur Berechnung der Koordinaten anhand der gemessenen Entfernung und zweier Winkel (in Eclipse [25] umgesetzt)

Dieser Ablauf wird für jeden Punkt des Rasters durchgeführt.

Nachdem das Abtasten abgeschlossen ist, werden die Punkte in einer Datei abgespeichert. Diese enthält dann die x, y- und z-Koordinate jedes Punktes der Punktwolke, also das Scanergebnis für den gemessenen Raum.

Es hat sich als sinnvoll erwiesen, einen Abstand von ca. 2,4° zwischen den Messpunkten zu verwenden. Dies bedeutet, dass bei einem Scan, welcher 360° x 50° des Raumes (d.h. vollständiger Scan eines Raums) abscannt, ca. 100.000 Messpunkte erfasst werden. Da

jeder Punkt einzeln angesteuert werden muss, dauert ein solcher Scan ca. 7 – 8 Stunden.

4.4 Visualisierung sowie Auf- und Nacharbeitung der Scanergebnisse

Um die durch einen Scan-Vorgang erzeugten Messergebnisse (Punktwolke) nutzen zu können, müssen die Daten visualisiert und ggf. auf- bzw. nachbearbeitet werden.

4.4.1 3D-Darstellung der Punktwolke

Um den Scan im dreidimensionalen Raum richtig darstellen zu können, müssen die Punkte perspektivisch dargestellt werden. Hierzu wird eine Java-Programm-Bibliothek, die sogenannte Java FX-Bibliothek genutzt, die bereits wichtige Elemente für eine 3D-Darstellung, z.B. eine perspektivische Kamera, als Programm-Code bereitstellt [19].

Im Einzelnen geht das selbst erstellte Programm zur 3D-Darstellung folgendermaßen vor (Der vollständige Programmcode findet sich im Anhang der digitalen Version.):

Das Programm lädt die Datei (Punktwolke), die das Scanergebnis beinhaltet, extrahiert die einzelnen Punkte und stellt jedes Element der Punktwolke an der entsprechenden Position durch einen kleinen 3D-Würfel dar. Der dreidimensionale Eindruck entsteht für den Betrachter zum einen durch die perspektivische Darstellung bezogen auf eine virtuelle Kameraposition, zum anderen dadurch, dass durch den Abtastvorgang die Punktdichte für geringe Abstände höher ist und darüber hinaus näher liegende Würfel größer dargestellt werden als weiter entfernt liegende Punkte und dahinterliegende Würfel verdecken.

Um nun das Scanresultat auch von verschiedenen Blickrichtung ansehen zu können, kann man diese Sicht- bzw. Kameraposition bewegen. Dies wird durch Tastatur und Mausabfragen gelöst. Dabei kann man sich mit den Tasten W und S vor und zurückbewegen, mit den Tasten A und D links und rechts und mit den Tasten Q und E nach oben und unten. Bei gedrückter Maustaste kann man sich durch Mausbewegung umschauen.

Die folgende Abbildung zeigen das Ergebnis für den Scan eines Dachraums.

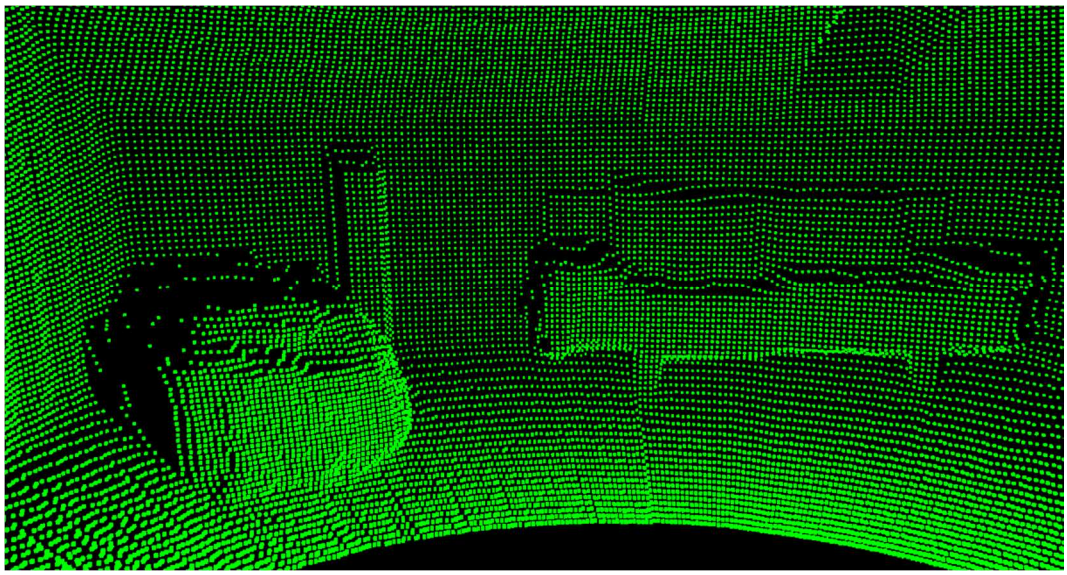


Abbildung 24: 3D-Darstellung des Scans (Screenshot aus selbstgeschriebenem 3D-Viewer)

Abbildung 25: Foto des Scanausschnitts (Während des Scans lag eine Tasche auf der rechten Seite des Sofas)

Wie man sieht, sind die Umrisse der verschiedenen Objekte wie Sofa oder Lautsprecher deutlich zu erkennen.

Allerdings ist die 3D-Bibliothek von Java-FX nicht besonders effizient, was bedeutet, dass bei größeren Scans mit bis zu 100.000 Messpunkten die Bewegung und das Umschauen leicht stocken.

Zudem bietet die Bibliothek keine Möglichkeit der Bearbeitung.

4.4.2 Auf- und Nachbearbeitung des Scanergebnisses mithilfe eines 3D-Modellierungsprogramms

Für die Auf- und Nachbearbeitung des Scanergebnisses bietet sich das 3D-Modellierungsprogramm „Blender“ an, das als Open-Source-Software verfügbar ist und z.B. in der Spieleprogrammierung zur Generierung von 3D-Darstellungen genutzt werden kann. Das Programm „Blender“ bietet eine sehr effiziente Darstellung von 3D- Objekten und bietet diverse Bearbeitungsmöglichkeiten [23].

Mithilfe eines selbst geschriebenen Konverters (Programmcode siehe Anlage) wurde das Scanergebnis zunächst in ein Dateiformat konvertiert, welches sich in Blender importieren lässt.

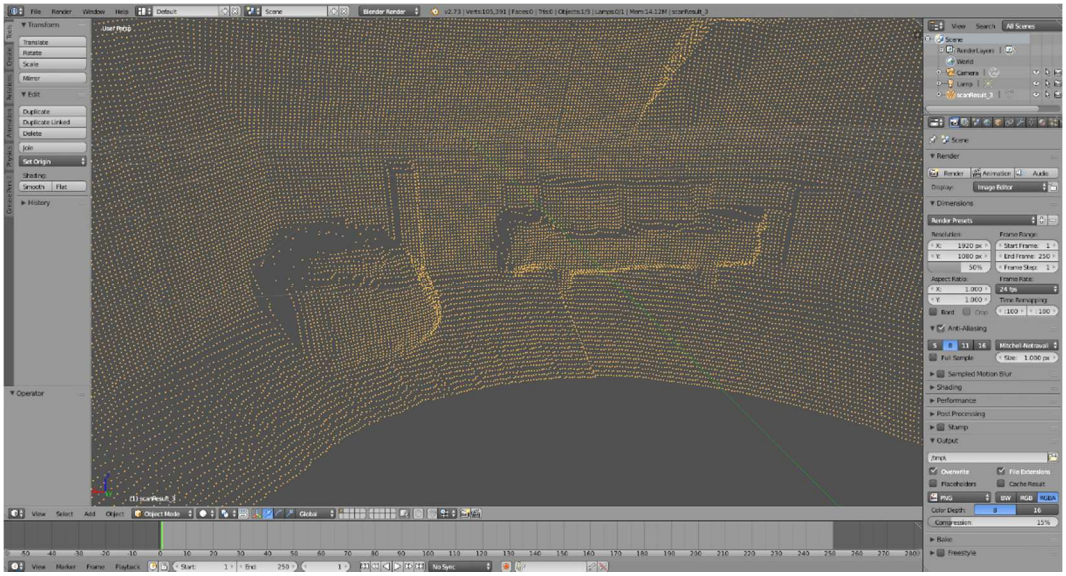


Abbildung 26: 3D-Modellierungsprogramm Blender"[23]

Zusätzlich zu Funktionen wie Skalieren, Rotieren und Verschieben, bietet das Programm mit Hilfe des PlugIns „Point Cloud Skinner“ die Möglichkeit, die Punkte zu einem Netz, genannt „Mesh“ zusammenzusetzen. Dabei werden benachbarte Bildpunkte zu Flächen verbunden, so dass die Punktwolke in eine Flächendarstellung umgewandelt wird [20]. Der räumliche Eindruck der Darstellung kann weiter dadurch optimiert werden, dass in Blender eine virtuelle Beleuchtung an einem definierten Punkt eingefügt wird, so dass zusätzlich Schatteneffekten entstehen. Abbildung 27 zeigt das Ergebnis einer solchen Bearbeitung. Die virtuelle Beleuchtung befindet sich in dieser Darstellung rechts oben. Auch in dieser Darstellung kann die Sicht- bzw. Kameraposition wieder beliebig verändert werden.

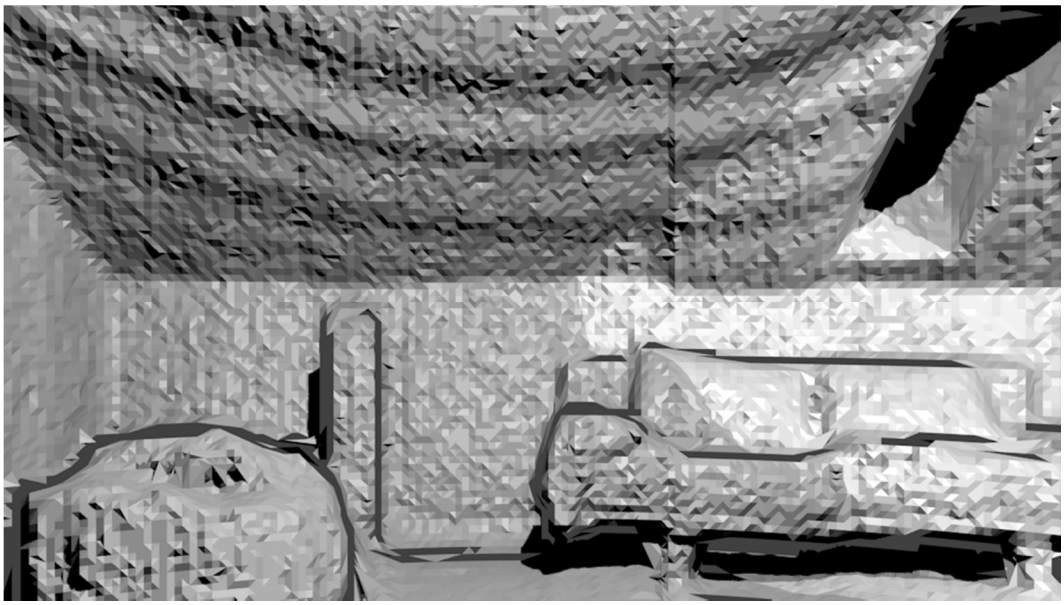


Abbildung 27: Gerenderter Ausschnitt des Scans (Programm Blender) [23]

4.4.3 Tiefenbild

Neben der bisher gezeigten Darstellung als Gitternetz (Mesh) kann man das Scanergebnis als ein zweidimensionales Tiefenbild darstellen (siehe Abbildung 28).

Das selbstgeschriebene Programm berechnet aus den Entfernungen jedes Messpunktes zum Scanner einen Graustufenwert und färbt die dementsprechende Position auf dem Bild ein. Je dunkler, desto weiter entfernt ist dabei der Punkt.

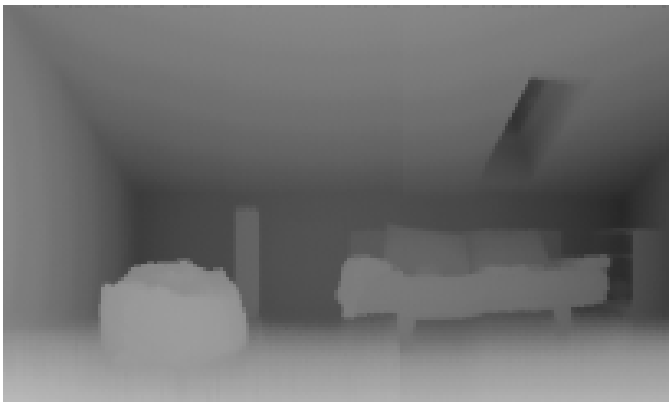


Abbildung 28: Tiefenbild des Scanausschnitts (mit selbstgeschriebenem Programm erstellt)

Diese Darstellung ist sehr einfach, Objekte sind gut erkennbar. Allerdings gehen in einem Tiefenbild viele Informationen, wie zum Beispiel die genaue Entfernung, verloren, da nur 256 verschiedene Graustufen und damit auch nur diese Zahl an Entfernungswerten dargestellt werden können. Darüber hinaus ist die Darstellung von Kanten teilweise weniger prägnant, da sich an Kanten der Tiefenwert nur wenig ändert.

4.4.4 Fehlerkorrektur

Von Zeit zu Zeit treten beim Scanvorgang Messfehler auf, das heißt, dass eine Entfernung falsch gemessen wurde.

Diese Fehler sieht man beispielsweise im Tiefenbild sehr deutlich:

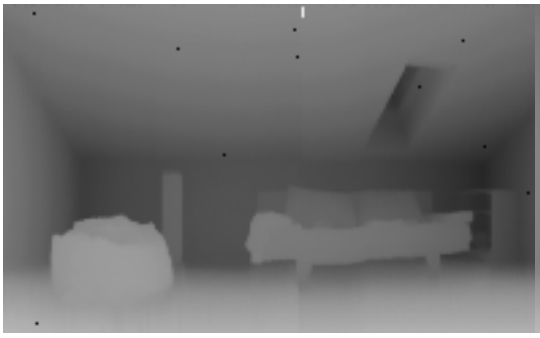


Abbildung 29: Tiefenbild ohne Nachbearbeitung (mit selbstgeschriebenem Programm erstellt)

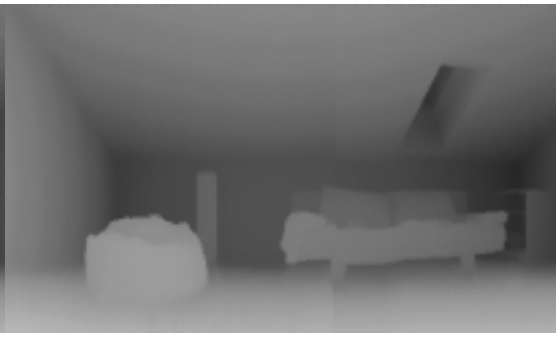


Abbildung 30: Tiefenbild mit Nachbearbeitung (mit selbstgeschriebenem Programm erstellt)

Die schwarzen und weißen Punkte zeigen, wo Messfehler aufgetreten sind. Bei schwarzen Punkten wurde eine viel zu große Entfernung, bei weißen Punkten eine Entfernung von 0 gemessen.

Um diese Fehler zu korrigieren, wurde ein Algorithmus zur Fehlerbehebung programmiert:

Der Algorithmus überprüft zunächst, wie weit der Punkt vom Durchschnitt der acht Punkte um ihn herum entfernt ist. Wenn diese Entfernung kleiner ist als ein festgelegter Wert, der dem Algorithmus übergeben wird, wird der Punkt als korrekt gemessen angesehen. Der Wert muss dabei so gewählt werden, dass z.B. Übergänge zwischen einem Objekt und seinem Hintergrund akzeptiert werden, ungewöhnlich große Sprünge aber ausgeschlossen werden.

Falls diese Überprüfung negativ ausfällt, werden die Entfernungen zu den einzelnen umliegenden Punkten geprüft. Im Fall, dass der Punkt einem umliegenden Punkt näher als der festgelegte Wert ist, wird er ebenfalls als korrekt angesehen. Diese Überprüfung ist vor allem für kleinere Objekte wichtig, die nur von wenigen Messungen erfasst wurde.

Falls alle Überprüfungen negativ ausfallen, erfolgt die Fehlerkorrektur und der Messwert für diesen Punkt wird durch den Durchschnittswert aller umliegenden Punkte ersetzt.

```
public Measurment checkPoint(Measurment[] points, float maxDistance)
{
	//points[0] ist der zu überprüfende Punkt. Die restlichen Punkte sind die
	umliegenden Punkte

	float a = 0;		//Durchschnittswert für x
	float b = 0;		//Durchschnittswert für y
	float c = 0;		//Durchschnittswert für z

	for(int i = 1; i < points.length; i++)
	{
		a += points[i].getX();
		b += points[i].getY();
		c += points[i].getZ();
	}

	a = a / (points.length - 1);
	b = b / (points.length - 1);
	c = c / (points.length - 1);

	//Überprüfung der Entfernung zum Durchschnittspunkt ( Math.abs(x) entspricht
	|x| )

		if(Math.abs(points[0].getX() - a) < maxDistance &&
		Math.abs(points[0].getY() - b) < maxDistance &&
		Math.abs(points[0].getZ() - c) < maxDistance)return points[0];

	//Überprüfung der Entfernungen zu den einzelnen umliegenden Punkten

	for(int i = 1; i < points.length; i++)
	{
		if(Math.abs(points[0].getX() - points[i].getX()) < maxDistance &&
		Math.abs(points[0].getY() - points[i].getY()) < maxDistance &&
		Math.abs(points[0].getZ() - points[i].getZ()) < maxDistance)
		return points[0];
	}

	return new Measurment(a, b, c);
}
```

Abbildung 31: Methode zur Überprüfung eines Punktes. Wird im Algorithmus für jeden einzelnen Messpunkt aufgerufen. (in Eclipse [25] umgesetzt).

4.5 360° Aufnahme

Um den Scanner abschließend zu testen, wurde eine 360° Aufnahme angefertigt. Diese zeigt einen gesamten Raum im Dachgeschoss. Bei diesem Scan sind die Form des Raumes und die Details im Raum sehr deutlich zu erkennen:

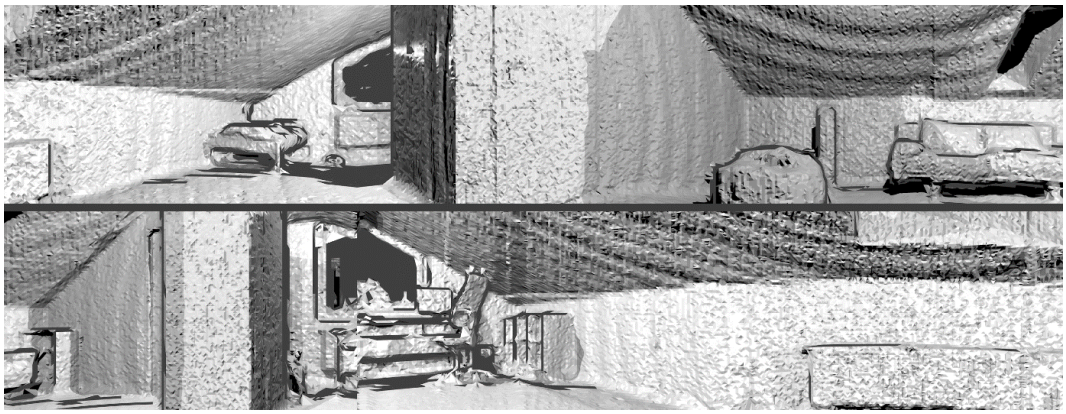


Abbildung 32: 360° Blender-Rendering des Scans von der Position des Scanners aus (oben 0° -180° unten 180°-360°) [23]

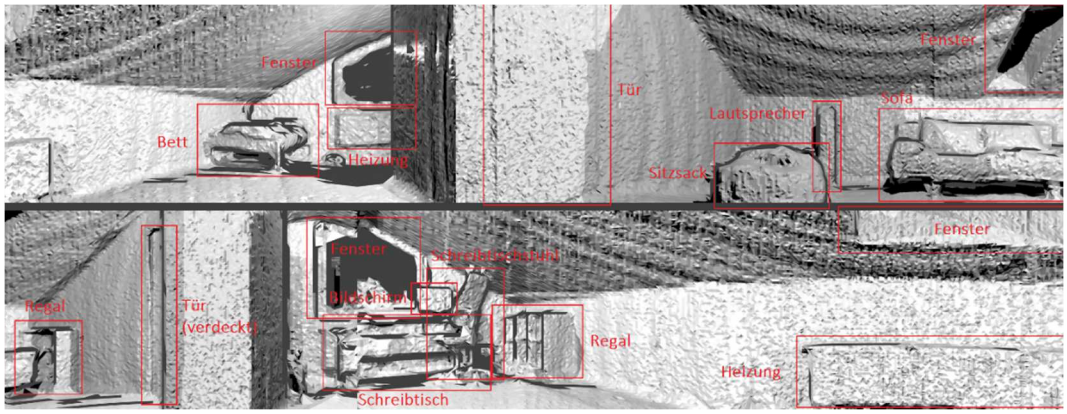


Abbildung 33: Abbildung 32 mit Beschriftung

Abbildung 34: Tiefenbild des Scans (mit selbstgeschriebenem Programm erstellt)

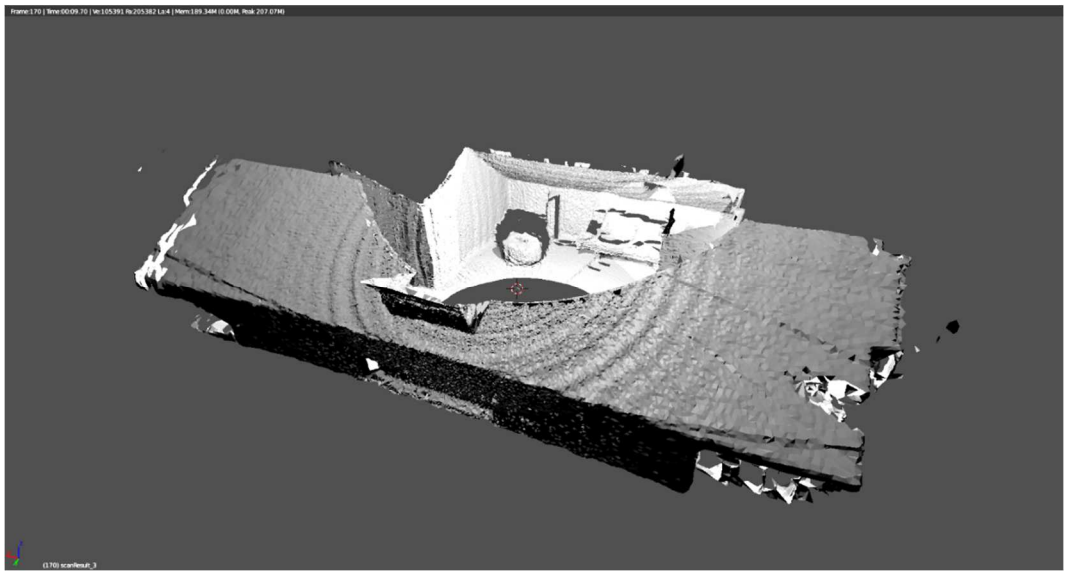


Abbildung 35:Blender- Rendering der Außenansicht. Grundriss ist deutlich zu erkennen [23].

Das Scanergebnis beinhaltet 105.391 Messpunkte (in Blender „Vertex“ dt. Eckpunkt). Durch das Verbinden der Messpunkte in Blender wurden zusätzlich 205.368 Dreiecksflächen (in Blender „Face“ dt. Fläche) erzeugt.

Abbildung 36: Informationsleiste in Blender [23]

Für diesen Scan benötigte der Scanner um die acht Stunden. Die resultierende Datei ist 3.429KB (nur Punkte) bzw. 13.398KB (mit Ecken, Kanten und Flächen) groß.

5. Fazit

In der vorliegenden Arbeit wurden ausgehend vom „Radar-Sinn“ der Comic-Figur Daredevil die heute üblichen Techniken des 3D-Scannens vorgestellt und mit dem Radar-Sinn von Daredevil verglichen. Im nächsten Schritt wurde ein 3D-Scanner basierend auf einem Time-of-Flight-LIDAR-Sensor realisiert. Dabei wurde die Mechanik aus Fischertechnik aufgebaut, zur Steuerung wurde ein Arduino Microcontroller Board genutzt.

Mit Hilfe des Scanners konnten mit einfachen technischen Mitteln und zu geringen Kosten erstaunlich gute Ergebnisse erreicht werden. So ermöglicht es der Aufbau, verschiedene Positionen präzise anzusteuern und die Entfernung auf wenige Zentimeter genau abzunehmen.

Die Programme, die den Scanner steuern und die Signale auswerten, wurden selbst in C und Java geschrieben und ermöglichen einen selbständigen und reibungslosen Betrieb des Scanners. Der Arduino empfängt Befehle, die von einem Java-Programm auf dem PC über die USB-Verbindung an ihn geschickt werden und führt diese aus. Das Ergebnis des Scans wird in Form einer Punktwolke in einer Datei abgespeichert, die eine Weiterverarbeitung des Ergebnisses ermöglicht.

Das Ergebnis lässt sich dann 3-dimensional mit Hilfe eines selbst geschriebenen 3D-Viewers aus verschiedenen Perspektiven anzeigen. Weiter wurde das Bildverarbeitungsprogramm Blender für eine Weiterverarbeitung der Punktwolke als 3D-Mesh (Punkte werden zu Flächen verbunden) genutzt. Mit Hilfe eines selbst verfassten Programms kann die Punktwolke auch als Tiefenbild (Entfernung vom Scanner wird in Graustufen gezeigt) dargestellt werden. Schließlich wurde ein Algorithmus zur Fehlerkorrektur implementiert.

Zusammenfassend lässt sich sagen, dass der Scanner eine überraschend hohe Genauigkeit erreichen kann und sehr wenige Messfehler auftreten. Details, wie z.B. ein Sofa-Kissen, lassen sich gut in der Visualisierungssoftware erkennen.

Für den Scanner und die Nachbearbeitung sind verschiedene Weiterentwicklungen denkbar, die aus Zeitgründen nicht in dieser Arbeit umgesetzt werden konnten. Beispielhaft werden die folgenden beiden Punkte genannt:

- So ist Arbeitsgeschwindigkeit des Scanners relativ niedrig und damit die Scandauer recht hoch (8 Stunden für einen 360° x 50° Ausschnitt). Dies lässt sich darauf zurückführen, dass der Scanner jeden Messpunkt einzeln anfahren muss, wobei er die Motoren an- und abschalten muss.
  Die Scandauer könnte man in Zukunft dadurch herabsetzen, dass man die Motoren nicht anhält, sondern die Messwerte abnimmt, während der Motor sich in Bewegung befindet. Mögliche Messfehler, die dadurch entstehen, dass sich der Sensor beim Messen bewegt, müssten korrigiert werden.
- Weiterhin könnte man eine aufwändigere Bildverarbeitung zur Weiterverarbeitung der Punktwolke einsetzen. Hierzu gibt es diverse Forschungsergebnisse, z.B. zum Detektieren von Kanten oder von bestimmte Formen. Dies würde ermöglichen, das Scanergebnis deutlich effizienter abzuspeichern, darzustellen und zu bearbeiten, da man beispielsweise für eine Wand nicht jeden einzelnen Messpunkt, sondern ein

Rechteck abspeichern müsste. Ein Beispiel für ein großes Open-Source-Projekt zur Bildbearbeitung von 3D-Punkt-Wolken ist die so genannte Point-Cloud-Library, die laufend von Forschern weltweit weiterentwickelt wird [21].

6. Literaturverzeichnis

[1] **Marvel**: Wiki-Eintrag zum Charakter „Daredevil“ http://marvel.com/universe/Daredevil_%28Matthew_Murdock%29 (letzter Zugriff 05.04.2015).

[2] **Wolff, Christian**: Frequenzmoduliertes Dauerstrichradar: Höchstmögliche Radarpräzision http://www.radartutorial.eu/02.basics/Frequenzmodulierte%20 Dauerstrichradarger%C3%A4te.de.html (letzter Zugriff 27.09.2015)

[3] **Mark Waid, Daredevil, Vol. 1**: Comicband, Marvel, August 2012

[4] **Wikipedia**: Wiki-Eintrag zu Chaff https://en.wikipedia.org/wiki/Chaff_%28countermeasure%29 (letzter Zugriff 27.09.2015)

[5] **The other Murdock Papers**: Artikelsammlung zu Daredevils Radarsinn http://www.theothermurdockpapers.com/category/ddscience/radar/ (letzter Zugriff 05.04.2015)

[6] **Botsch, Mario**: Vorlesung an der Universität Bielefeld http://wwwhomes.uni-bielefeld.de/ggoetze/Master/02_Scanning.pdf (letzter Zugriff 05.04.2015)

[7] **Wikipedia**: Eintrag zum Thema „3d Scanner“ http://en.wikipedia.org/wiki/3D_scanner (letzter Zugriff 07.06.2015)

[8] **Suite101**: http://suite101.de/article/3d-scanner-was-sind-beruehrungsbasierte-taktile-contact-scanner-a124407#.VXAuwUZ0nDc (letzter Zugriff 07.06.2015)

[9] http://www.leica-geosystems.com/de/Aktuelles_360.htm?id=2622 (letzter Zugriff 07.06.2015)

[10] Screenshot aus dem Video „Laser Sensors - Triangulation Principle“ https://www.youtube.com/watch?v=HsHyYB7ObCU (letzter Zugriff 07.06.2015)

[11] **Malhotra, Akash - Gupta, Kunal - Kant, Kamal**: Laser Triangulation for 3D Profiling of Target http://research.ijcaonline.org/volume35/number8/pxc3976159.pdf (letzter Zugriff 07.06.2015)

[12] **KT-elektronic**: Datenblatt zu Ultraschall Messmodul HC-SR04 http://www.mikrocontroller.net/attachment/218122/HC-SR04_ultraschallmodul_beschreibung_3.pdf (letzter Zugriff 27.09.2015)

[13] **PulsedLight3D, LIDAR-Lite v1**: Ausführliche Produktbeschreibung / Datenblatt http://lidarlite.com/docs/v2/pdf/LIDAR-Lite-v1-docs.pdf (letzter Zugriff 27.09.2015)

[14] **Knobloch Electronic- Produktions- und Vertriebsgesellschaft mbH**: Verwendete Fischertechnik Bauteile https://knobloch-gmbh.de/de/fischertechnik (letzter Zugriff 06.04.2015).

[15] https://www.arduino.cc/en/Main/ArduinoBoardMega2560 (letzter Zugriff 07.06.2015)

[16] https://www.arduino.cc/de/uploads/Main/ArduinoMega2560_R3_Front_450px.jpg (letzter Zugriff 07.06.2015)

[17] http://www.sainsmart.com/sainsmart-l293d-motor-drive-shield-for-arduino-duemilanove-mega-uno-r3-avr-atmel.html (letzter Zugriff 07.06.2015)

[18] **RXTX Bibliothek**: RXTX Bibliothek zur Öffnung einer USB-Verbindung mit Java http://rxtx.qbang.org/wiki/index.php/Main_Page (letzter Zugriff 06.04.2015)

[19] **Oracle, JavaFX**: Java-Bibliothek von Oracle für grafische Darstellung http://docs.oracle.com/javase/8/javase-clienttechnologies.htm (letzter Zugriff 27.09.2015)

[20] **Point Cloud Skinner**: Script zur Generierung von Oberflächen aus Punktwolken http://sourceforge.net/projects/pointcloudskin/ (letzter Zugriff 27.09.2015)
[21] **Point Cloud Library**: C-Bibliothek für die Bearbeitung von PointClouds http://pointclouds.org/ (letzter Zugriff 27.09.2015)
[22] **Geogebra**: Mathematiksoftware https://www.geogebra.org/ (letzter Zugriff 07.06.2015)
[23] **Blender**: 3D-Modellierungssoftware https://www.blender.org/ (letzter Zugriff 27.09.2015)
[24] **Gimp**: Bildbearbeitungssoftware http://www.gimp.org/ (letzter Zugriff 27.09.2015)
[25] **Eclipse**: Java Entwicklungsumgebung https://eclipse.org/ (letzter Zugriff 27.09.2015)
[26] **Arduino**: Homepage von Arduino www.arduino.cc (letzter Zugriff 05.04.2015)

Hintergrundinformationen:

[27] **Kakalios, James**: „Physik der Superhelden“ 3. Auflage April 2012
[28] **Bernardini, Fausto und Rushmeier, Holly**: (2002) The 3D Model Acquisition Pipeline *Comput. Graph. Forum* **21** (2): 149–172
[29] **Engelmann, Francis**: Affordable 3D Laser Scanning of Physical Objects – Bachelorarbeit RWTH Aachen University

7. Abbildungsverzeichnis

8. Verzeichnis der digitalen Anlagen

1. Digitale Version der Arbeit (.docx und .pdf)
2. Arduino Programm
3. Java Programm
    1. Programm zur Steuerung des Scanners
    2. 3D-Viewer
    3. Programm zur Fehlerbehebung
    4. Programm zum Umwandeln in ein Tiefenbild
    5. Programm zum Umwandeln in eine .obj Datei
4. Scanergebnis
    1. .txt Datei (Output der Software)
    2. Tiefenbild (original, bearbeitet)
    3. .obj Datei (Punktwolke, für Blender)
    4. .blend Datei (Mesh, für Blender)
5. Geogebra Dateien
6. Bilder des Aufbaus
7. Anleitungen und Anmerkungen zum Anhang

Ich erkläre hiermit, dass ich die Seminararbeit ohne fremde Hilfe angefertigt und nur die im Literaturverzeichnis angeführten Quellen und Hilfsmittel benützt habe.

....., den ....

Ort Datum **Unterschrift der Schülerin / des Schülers**