

MASTERTHESIS
Friederike Christina Gollor

Entwicklung eines Systems für Oberflächenscans und die Erstellung von 3D-Modellen von Personen

FAKULTÄT TECHNIK UND INFORMATIK
Department Informations- und Elektrotechnik

Faculty of Computer Science and Engineering
Department of Information and Electrical Engineering

Friederike Christina Gollor

Entwicklung eines Systems für Oberflächenscans und die Erstellung von 3D-Modellen
von Personen

Masterarbeit eingereicht im Rahmen der Masterprüfung
im Studiengang *Master of Science Automatisierung*
am Department Informations- und Elektrotechnik
der Fakultät Technik und Informatik
der Hochschule für Angewandte Wissenschaften Hamburg

Betreuender Prüfer: Prof. Dr.-Ing. Marc Hensel
Zweitgutachter: Prof. Dr. rer. nat. Ulrike Herster

Eingereicht am: 12. September 2022

Friederike Christina Gollor

Thema der Arbeit

Entwicklung eines Systems für Oberflächenscans und die Erstellung von 3D-Modellen von Personen

Stichworte

Bildverarbeitung, Registrierung, Stereokamera, 3D-Modell, RANSAC, ICP

Kurzzusammenfassung

In dieser Arbeit wird ein System zur Erstellung von 3D-Modellen von Personen entwickelt, das die Bilddaten mithilfe einer Stereokamera aufnimmt. Im Fokus steht ein funktionsfähiges Gesamtsystem, welches die Bildaufnahmen mithilfe von Bildverarbeitungsalgorithmen zu einem Modell zusammenfügt.

Friederike Christina Gollor

Title of Thesis

Development of a system for surface scans and the creation of 3D-models of people

Keywords

Image processing, registration, stereo camera, 3D-model, RANSAC, ICP

Abstract

In this thesis, a system for the creation of a 3D-model of a person is developed that captures image data with the help of a stereo camera. The focus is on a functional overall system that creates models out of recordings using image processing algorithms.

Inhaltsverzeichnis

Abbildungsverzeichnis	vii
Tabellenverzeichnis	ix
1 Einführung	1
1.1 Motivation	1
1.2 Zielsetzung	2
2 Grundlagen	3
2.1 3D-Datenerfassung	3
2.2 Segmentierung von Objekten in Punktwolken	4
2.2.1 k-Means	5
2.2.2 DBSCAN	5
2.3 Registrierung von Punktwolken	6
2.3.1 Fast Point Feature Histograms (FPFH)	6
2.3.2 Random Sample Consensus (RANSAC)	8
2.3.3 Iterative Closest Point Algorithmus (ICP)	10
2.4 Oberflächenrekonstruktion	12
3 Stand der Technik	13
3.1 Anwendungsbereiche	13
3.2 Verwandte Veröffentlichungen	14
4 Anforderungsanalyse	16
4.1 Systembeschreibung	16
4.2 Stakeholder	17
4.3 Anwendungsfälle	20
4.4 Rechtliche Bedingungen	23
4.5 Normen	23

4.6	Anforderungen	24
4.6.1	Hardware	24
4.6.2	Software	27
5	Konzept	29
5.1	Hardware	29
5.1.1	Sensorhalterung	29
5.1.2	Sensorauswahl	30
5.2	Software	35
5.2.1	Entwicklungsumgebung	35
5.2.2	Gesamtablauf	36
5.2.3	Bedienoberfläche	36
5.2.4	Aufnahme der Punktwolken	37
5.2.5	Bildverarbeitung	38
6	Design	46
6.1	Hardware	46
6.1.1	Kamerahalterung	46
6.1.2	Montage der Sensoren	49
6.1.3	Schrittmotor	51
6.2	Software	52
6.2.1	Gesamtablauf	52
6.2.2	Schnittstellen	52
6.2.3	Bedienoberfläche	53
6.2.4	Aufnahme der Punktwolken	54
6.2.5	Bildverarbeitung	57
7	Umsetzung	63
7.1	Hardware	63
7.1.1	Kamerahalterung	63
7.1.2	Motorhalterung	67
7.1.3	Montage der Sensoren	68
7.2	Software	68
7.2.1	Bedienoberfläche	69
7.2.2	Aufnahme der Punktwolken	72
7.2.3	Bildverarbeitung	74

8 Test	79
8.1 Hardware	79
8.1.1 Allgemein	79
8.1.2 Sicherheit der Hardware	83
8.1.3 Sensor	85
8.2 Software	86
8.2.1 Bildverarbeitung	90
9 Fazit und Ausblick	101
Literaturverzeichnis	103
A Anhang	111
Selbstständigkeitserklärung	112

Abbildungsverzeichnis

2.1	Modell der Stereokamera nach [3, 18]	4
2.2	DBSCAN	6
2.3	Darboux-Koordinatensystem - Darstellung aus [57]	7
2.4	Diagramm der Einflussregion für das Fast Point Feature Histogram - Darstellung nach [57]	8
2.5	Beispielhafte Darstellung des RANSAC-Algorithmus anhand eines 2D-Datensatzes	9
2.6	ICP-Ablaufdiagramm	11
2.7	Reduzieren des Abstands mit dem Punkt-zu-Ebene-Verfahren - Abbildung nach [33]	11
2.8	Ball Pivoting im 2D-Raum - Abbildung nach [54]	12
3.1	NEXR 3D-Körperscanner [43]	14
4.1	Systemkontext	17
4.2	Anwendungsfall-Diagramm	20
4.3	Abstand des Sensors zur Person	27
5.1	Gesamtablauf	37
5.2	Aktivitätsdiagramm der Bedienoberfläche - 3D-Modell erstellen	37
5.3	Aktivitätsdiagramm der Bedienoberfläche - Bildvorschau	38
5.4	Aktivitätsdiagramm der Bildverarbeitung	38
5.5	Kamerasichtfeld und Hintergrund	39
6.1	Konstruktion	47
6.3	Motorhalterung und Befestigung des U-Rahmens	48
6.2	Detailansicht der Aufhängung	48
6.4	Montage der Kameras	50
6.5	Schrittmotorsteuerung	51

6.6	Übersicht des Gesamtablaufs	53
6.7	Übersicht der Klassen	54
6.8	Bedienoberfläche	55
6.9	Drahtlose Konfiguration	56
6.10	Kabelgebundene Konfiguration	56
6.11	Ablauf der Aufnahme der Punktwolken	58
6.12	Übersicht der Module für die Bildverarbeitung	59
6.13	Ablauf des DBSCAN-Algorithmus	60
6.14	Ablauf der Registrierung	61
6.15	Ablauf der Oberflächenrekonstruktion	62
7.1	Konstruktion	64
7.2	Konstruktion 2	65
7.4	Konstruktion 3	66
7.3	Konstruktion 3 mit U-Rahmen	66
7.5	Motorhalterung - Maße in mm	68
7.6	Darstellung des Gesichts durch Punktwolken aus drei Perspektiven	69
7.7	Startbildschirm	70
7.8	Aufnahme	70
7.9	Countdown	71
7.10	Exploreransicht	71
7.11	Visualisierung der Modelle	72
7.12	Readme	73
7.13	Umsetzung der Registrierung	77
7.14	Vergleich der Fitnesswerte	78
7.15	Vergleich der rekonstruierten Oberflächen mit der Punktwolke	78
8.1	Punktwolke mit Hintergrund	92
8.2	Hintergrund entfernen	92
8.3	Test 1	94
8.4	Registrierung (Fitnesswert RANSAC: 0,29 und ICP: 0,14)	97
8.5	Registrierung (Fitnesswert RANSAC: 0,74 und ICP: 0,24)	98
8.6	Test 3a: 20° (Fitnesswert RANSAC: 0,83 und ICP: 0,35)	99
8.7	Test 3b: 45° (Fitnesswert RANSAC: 0,73 und ICP: 0,27)	100
8.8	Oberflächenrekonstruktion	100

Tabellenverzeichnis

1.1	Zielbeschreibung	2
2.1	Anzahl der Iterationen N , um mit einer Wahrscheinlichkeit von $P = 0,99$ mindestens ein Modell keine Ausreißer hat. Tabelle aus [20]	10
4.1	Anwendungsfall: 3D-Objekt erstellen starten	21
4.2	Anwendungsfall: Aufnahme abbrechen	21
4.5	Anwendungsfall: 3D-Modell speichern	21
4.3	Anwendungsfall: Daten erheben	22
4.4	Anwendungsfall: 3D-Modell erstellen	22
4.6	Anwendungsfall: 3D-Modell anzeigen	22
5.1	Vergleich der Arbeitsbereiche in z-Richtung nach [23, 26, 63, 38]	31
5.2	Vergleich der Arbeitsbereich Schnittstellen [23, 26, 63, 38]	31
5.3	Vergleich der Anschaffungskosten nach [9, 64]	32
5.4	Vergleich der Kameramodelle	32
5.5	Vergleich der Intel RealSense D400-Modelle nach [26]	33
5.6	Vergleich Intel RealSense	34
5.7	Vergleich der Ansätze	40
5.8	Vergleich Clusteranalysealgorithmen	41
5.9	Vergleich globale Registrierung	43
5.10	Vergleich lokale Registrierung	44
6.1	Aluprofilkonstruktion - Angaben nach [14]	47
6.2	Kosten Konzept 1: Höhenverstellbare Kamera - Angaben nach [9, 14]	49
6.3	Kosten Konzept 2: Fest montierte Kameras - Angaben nach [9, 14]	49
6.4	Vergleich der Konzepte	50
6.5	Motorbefestigung - Angaben nach [14]	51
6.6	Schrittmotorsteuerung - Angaben nach [9]	52

7.1	Vergleich der Konstruktionen	67
7.2	Aluprofilkonstruktion - Angaben nach [14]	67
8.1	Anforderungen: Allgemein	80
8.2	Anforderungen: Sicherheit	83
8.3	Anforderungen: Sensor	85
8.4	Anforderungen: Software	86
8.5	Anforderungen: Software (optional)	89

Abkürzungsverzeichnis

RANSAC Random Sample Consensus

ICP Iterative Closest Point

SPFH Simplified Point Feature Histogram

FPFH Fast Point Feature Histogram

DBSCAN Density-Based Spatial Clustering of Applications with Noise

CGI Computer Generated Imagery

rpm revolutions per minute *deutsch*: Umdrehungen pro Minute

MinPts Minimum number of Points

GPU *englisch* graphic processing unit, *deutsch* Grafikprozessoren

1 Einführung

1.1 Motivation

3D-Scanner finden zunehmend einen Platz in unserem Alltag. Von individuell angepassten Schuhen von zuhause aus bis hin zu Ganzkörperscans zur Verfolgung von Trainingsfortschritten im Fitnessstudio. Auch für die Produktion von Filmen werden zunehmend 3D-Scanner eingesetzt. Mit ihnen lassen sich Modelle wie Gebäude oder Personen einscannen, um diese mithilfe von Computer Generated Imagery, kurz CGI, im Filmgeschehen einzusetzen.

Im Rahmen dieser Arbeit soll ein Gesamtsystem eines 3D-Scanners, bestehend aus einem Hardwareaufbau zur Befestigung eines Sensors für die 3D-Datenaufnahme und einem Softwareteil zur Verarbeitung der aufgenommenen Bilddaten zu einem 3D-Modell, entwickelt werden. Das System soll die Grundlage für die Vertiefung der einzelnen Themenbereiche in zukünftigen Arbeiten bilden. Diese untergliedern sich zum einen in den mechatronischen Aufbau, der eine automatisierte 3D-Datenerfassung ermöglichen soll und zum anderen in den Softwareteil bestehend aus 3D-Bilderfassung, die Verarbeitung und Zusammenführung von Punktwolken und die Rekonstruktion einer Oberfläche. Neben der funktionellen Umsetzung kann dabei in zukünftigen Projekten auch die Ausführungsgeschwindigkeit bei der Verarbeitung großer Datenmengen beispielsweise durch Nutzung von GPUs (*englisch* graphic processing unit, *deutsch* Grafikprozessoren) behandelt werden. Da sich die Kosten für Systeme dieser Art, aufgrund der verwendeten Sensorik, im oberen fünfstelligen Bereich belaufen, soll hier eine kostengünstigere Alternative geschaffen werden. Das System soll zudem für die Innen- und Außendarstellung der HAW genutzt werden können.

1.2 Zielsetzung

Für die Zielformulierung wird die SMART-Methode aus [71] verwendet. Die Zielbeschreibung erfolgt in Tabelle 1.1.

Tabelle 1.1: Zielbeschreibung

Specific	Es soll ein System für die Erstellung eines digitalen 3D-Modells des Kopfes von Personen entwickelt werden. Die Aufnahme der Daten können Personen durchführen die in keinem fachlichen Zusammenhang mit dem Projekt stehen.
Measurable	Es wird ein 3D-Modell erstellt, das mit einem 3D-Drucker druckbar ist.
Achievable	Entwickelt wird ein funktionierender Prototyp des Gesamtsystems von der Bildaufnahme bis zur Rekonstruktion der Oberfläche, wobei weniger die Qualität der einzelnen Funktionen und die Erstellung eines vollständigen 3D-Modells im Vordergrund stehen.
Realistic	Dafür wird eine Sensorhalterung und eine Software für die Bildverarbeitung entwickelt.
Time-Bound	Das Projekt wird im Rahmen einer Masterarbeit umgesetzt.

2 Grundlagen

Diese Kapitel beschreibt die Grundlagen für die Erstellung von 3D-Modellen, die auf Basis von 3D-Daten berechnet werden. Die Daten können mit einem Sensor zur Aufnahme von Tiefeninformation aufgenommen werden. Neben der Person, von der ein 3D-Modell erstellt werden soll, enthalten diese Daten auch Rauschen und einen Hintergrund, der vor der Weiterverarbeitung entfernt werden muss. Für ein vollständiges Modell werden Daten aus mehreren Perspektiven zu einem Modell kombiniert und eine Oberfläche erzeugt.

2.1 3D-Datenerfassung

Zur Erzeugung eines dreidimensionalen Bildes werden Tiefeninformationen benötigt, die Auskunft über die Entfernung eines Objekts zum Bildaufnehmer geben. Diese Informationen können mithilfe von Stereoskopie oder durch das Laufzeitverfahren aufgenommen werden. Im Folgenden wird die Stereoskopie, wie sie bei Stereokameras angewendet wird, nach [3, 18] erläutert.

Bei der Stereoskopie wird eine Szene von mindestens zwei Kameras aus unterschiedlichen Perspektiven aufgenommen. In Abbildung 2.1 wird ein geometrisches Modell einer Stereokamera dargestellt. Die beiden Kameras sind nebeneinander so angeordnet, dass die Bildebenen komplanar und die optischen Achsen parallel sind. Der Abstand zwischen den optischen Achsen wird als Basislinie b bezeichnet. Beide Kameras haben dieselbe Brennweite f . Der Punkt P wird auf die x -Achse (Epipolarlinie) abgebildet und hat zu beiden Kameras die gleiche Entfernung z , auch Gegenstandsweite g genannt, die parallel zur Bildebene verläuft. Über die Disparität (Parallaxe) zwischen den Bildaufnehmern, die die Verschiebung der Kameras entlang der x -Achse mit $p = x'_1 - x'_2$ beschreibt, wird die Tiefe berechnet. Unter Verwendung des Strahlensatzes erhält man Folgendes, wobei $\Delta x = b$ ist:

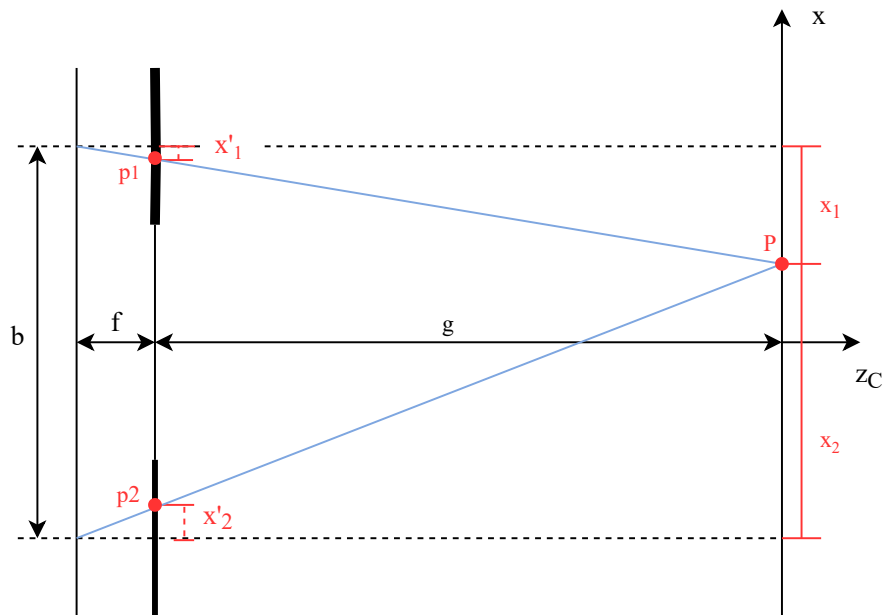


Abbildung 2.1: Modell der Stereokamera nach [3, 18]

$$\frac{x_1}{g} = \frac{x'_1}{f} \quad (2.1)$$

$$\frac{x_1 - \Delta x}{g} = \frac{x'_2}{f} \quad (2.2)$$

Nach dem Umstellen nach x_1 , können die Gleichungen gleichgesetzt und nach g aufgelöst werden:

$$g = z = \frac{\Delta x \cdot f}{x'_1 - x'_2} = \frac{b \cdot f}{p} \quad (2.3)$$

2.2 Segmentierung von Objekten in Punktwolken

Tiefeninformationen werden in einem Datensatz als Punktwolke gespeichert, die die Punkte einer Oberfläche als dreidimensionale Koordinaten abbilden. Um aus einer Punktwolke nur die benötigten Bereiche herauszufiltern werden Bereichs- oder auch Clusterana-

lysen durchgeführt. Im Folgenden werden zwei Algorithmen zur Analyse von Datensätzen vorgestellt, die die Daten oder hier Punkte in Bereiche aufteilen. Dafür wird zunächst der k-Means-Algorithmus vorgestellt, der nach dem Partitionierungsprinzip arbeitet, und im Anschluss der DBSCAN-Algorithmus, der die Aufteilung dichtebasiert vornimmt.

2.2.1 k-Means

k-Means partitioniert die Punktwolke in k -Bereiche, die Cluster. Das k wird vor Beginn der Ausführung festgelegt. Zu Beginn werden k -Clusterschwerpunkte in der Punktwolke zufällig positioniert. Anschließend werden die Punkte den Clusterschwerpunkten zugeordnet, zu dem ihre euklidischen Distanz am geringsten ist. Aus den Abständen eine Gruppe zu ihrem Clusterschwerpunkt wird ein Mittelwert berechnet, um die Position des Clusterschwerpunkts neu auszurichten. Es werden erneut die Abstände zu den Schwerpunkten berechnet und die Zuordnung der Punkte angepasst. Der Prozess wird wiederholt bis sich die Änderung der Position des Clusterschwerpunktes innerhalb eines definierten Toleranzbereichs befindet. [1, 10]

2.2.2 DBSCAN

DBSCAN steht für Density Based Spatial Clustering of Applications with Noise und ist ein dichtebasierter Algorithmus zur Clusteranalyse. Für die Analyse der Cluster wird die Nachbarschaft aller Punkte der Punktwolke betrachtet. Liegen im Radius r um einen Punkt \mathbf{P} eine Mindestanzahl an Punkten *MinPts* (*englisch* minimum number of points) wird der Punkt als Kernobjekt klassifiziert, siehe Abbildung 2.2 in rot. In der Nachbarschaft der Punkte die das Kernobjekt umgeben wird anschließend nach weiteren Punkten gesucht, die die Anforderungen erfüllen. Befindet sich ein Punkt in dieser Nachbarschaft der zwar im Radius eines dieser Punkte liegt, aber nicht die minimale Anzahl an Punkten in der Nachbarschaft besitzt, wird er als Randpunkt klassifiziert (grün). Diese Punkte werden zu einem Cluster zusammengefasst. Kann ein Punkt die Anforderungen nicht erfüllen ist er ein Rauschpunkt (blau). [31]

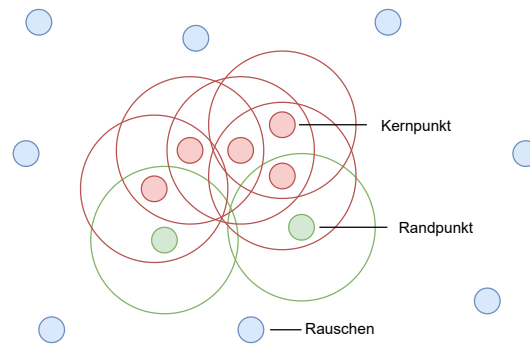


Abbildung 2.2: DBSCAN

2.3 Registrierung von Punktwolken

Das Zusammenführen von Punktwolken, die aus unterschiedlichen Perspektiven aufgenommen worden sind und sich überschneidende Bereiche besitzen, wird als Registrierung bezeichnet. Um die Überschneidungsbereiche der Punktwolken in Deckung zu bringen, wird die Ausgangspunktwolke durch Rotation und Translation an die Zielpunktwolke angenähert.

Für die Registrierung werden die Punkte in der Ziel- und der Ausgangspunktwolke anhand von Oberflächenmerkmalen charakterisiert. Die lokale Diskription erfolgt beispielsweise auf Grundlage von Normalen oder Eigenvektoren der lokalen Oberfläche. In Abschnitt 2.3.1 wird der in dieser Arbeit verwendete histogrammbasierte Merkmalsdeskriptor Fast Point Feature Histogram beschrieben. Im zweiten Schritt werden Korrespondenzen zwischen den Punktwolken anhand der Oberflächenmerkmale ermittelt und eine globale Registrierung durchgeführt. Diese Aufgabe wird im vorliegenden Fall durch den Random Sample Consensus (RANSAC) Algorithmus ausgeführt. Dieser wird in Abschnitt 2.3.2 beschrieben. Globale Registrierung bedeutet, dass die Punktwolken zu Beginn nicht zueinander ausgerichtet sind und zunächst eine grobe Ausrichtung durchgeführt werden muss.

2.3.1 Fast Point Feature Histograms (FPFH)

Mithilfe des Fast Point Feature Histogram, kurz FPFH, werden die geometrischen Eigenschaften in der Nachbarschaft um einen Punkt \mathbf{p} einer Punktwolke \mathbf{P} in einem mehr-

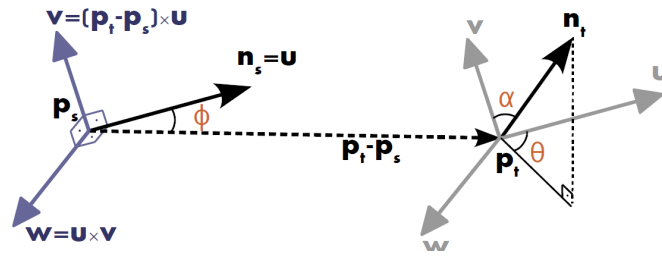


Abbildung 2.3: Darboux-Koordinatensystem - Darstellung aus [57]

dimensionalen Histogramm beschrieben. Die Erklärung des FPFH erfolgt auf Grundlage von [58] und [59].

Für einen durch einen Ortsvektor \mathbf{p}_s repräsentierten Punkt werden in einem Radius r die k Nachbarn des Punkts bestimmt, siehe Abbildung 2.4. Zur Berechnung des relativen Abstands zwischen zwei Punkten \mathbf{p}_s und \mathbf{p}_t sowie ihren Normalen \mathbf{n}_s und \mathbf{n}_t , die senkrecht zur Oberfläche der Punktwolke stehen, wird ein Koordinatensystem definiert, siehe Abbildung 2.3. Es gilt:

$$\mathbf{u} = \mathbf{n}_s; \quad \mathbf{v} = (\mathbf{p}_t - \mathbf{p}_s) \times \mathbf{u}; \quad \mathbf{w} = \mathbf{u} \times \mathbf{v} \quad (2.4)$$

Für jeden Abfragepunkt \mathbf{p} wird das 3-Tupel $[\alpha, \phi, \theta]$ zwischen dem Punkt und seinen Nachbarn berechnet. Das bezeichnet man als Simplified Point Feature Histogram (SPFH), siehe Abbildung 2.4a.

$$\alpha = \mathbf{v} \cdot \mathbf{n}_t; \quad \phi = (\mathbf{u} \cdot (\mathbf{p}_t - \mathbf{p}_s)) / \|\mathbf{p}_t - \mathbf{p}_s\|; \quad \theta = \arctan(\mathbf{w} \cdot \mathbf{n}_t, \mathbf{u} \cdot \mathbf{n}_t) \quad (2.5)$$

Innerhalb der Punktwolke \mathbf{P} wird die Berechnung für jeden Abfragepunkt \mathbf{p} und seine ihn umgebenden Nachbarn durchgeführt. Nach Abbildung 2.4 wird die Berechnung im ersten Schritt zwischen \mathbf{p}_s und seinen Nachbarn \mathbf{p}_{t1} bis \mathbf{p}_{t5} , dargestellt durch die rotgestrichelten Linien, durchgeführt. Im nächsten Schritt wird die Berechnung für jeden der k -Nachbarpunkte \mathbf{p}_{t1} bis \mathbf{p}_{t5} wiederholt. Die Ergebnisse der SPFH der Nachbarn dienen der Gewichtung der Werte im Histogramm von \mathbf{p}_s , siehe Abbildung 2.4b. Das FPFH berechnet sich wie folgt:

$$FPFH(p_s) = SPFH(p_s) + \frac{1}{k} \sum_{i=1}^k \frac{1}{\omega_t} \cdot SPFH(p_t) \quad (2.6)$$

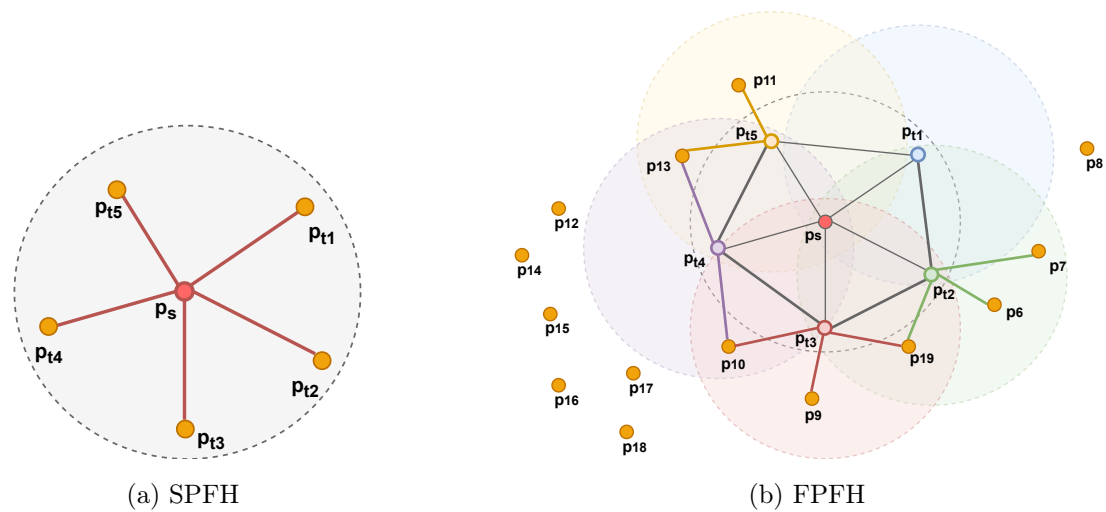


Abbildung 2.4: Diagramm der Einflussregion für das Fast Point Feature Histogramm - Darstellung nach [57]

Der Betrag des Abstands zwischen dem Abfragepunkt \mathbf{p}_s und seinen Nachbarn \mathbf{p}_t wird durch das Gewicht ω_t dargestellt.

2.3.2 Random Sample Consensus (RANSAC)

Der Random Sample Consensus, kurz RANSAC, wird unter anderem für die globale Registrierung von Punktwolken verwendet. Aus den Daten der Punktwolken schätzt der Algorithmus Modelle. Er ist robust gegenüber nicht übereinstimmenden Punktepaaren, die den Registrierungsprozess stören. Im Gegensatz zu anderen Algorithmen verwendet RANSAC nur eine geringe Datenmenge zur Modellbildung. Das Funktionsprinzip wird anhand von [16] und [52] erklärt.

Der Ablauf des Algorithmus' ist in Abbildung 2.5 anhand einer Geraden im zweidimensionalen Raum dargestellt. Aus einer Punktwolke werden zufällig n -Datenpunkte ausgewählt (1). Unter der Annahme, dass die Teilmenge keine Ausreißer enthält, wird ein Modell berechnet (2). Im Anschluss wird bestimmt, wie viele Punkte der Punktwolke in einem festgelegten Toleranzbereich um das berechnete Modell liegen (3). Die Schritte werden mehrfach mit neuen Modellen wiederholt. Ziel ist es, das Modell zu finden, in dessen Toleranzbereich die meisten Punkte der Punktwolke liegen.

Um sicherzustellen, dass mit einer definierten Wahrscheinlichkeit ein Modell berechnet wird, dass frei von Ausreißern ist, müssen N Versuche durchgeführt werden. Die Erklär-

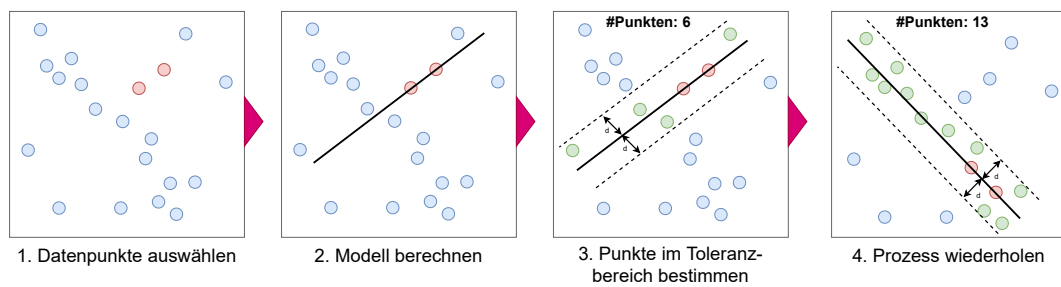


Abbildung 2.5: Beispielhafte Darstellung des RANSAC-Algorithmus anhand eines 2D-Datensatzes

Die Berechnung erfolgt nach [20] und [65]. p definiert die Wahrscheinlichkeit, dass mindestens ein gebildetes Modell frei von Ausreißerpunkten ist. Umgekehrt definiert $1 - p$ die Wahrscheinlichkeit, dass mindestens ein Datenpunkt im gebildeten Modell ein Ausreißer ist. Die Ausreißerwahrscheinlichkeit e gibt an, wie groß der Anteil an Ausreißern ist, der in einem Iterationsschritt toleriert wird. Ein Modell gilt dann als akzeptabel, wenn dieser Anteil eingehalten wird. Für k zufällig ausgewählte Datenpunkte ist die Wahrscheinlichkeit, dass alle Datenpunkte in einem Durchlauf keine Ausreißer sind p^k . Dabei ist $p^k = (1 - e)^k$. Die Wahrscheinlichkeit, dass alle Datenpunkte Ausreißer sind, ist $1 - p^k$. Angenommen in allen Iterationen N gibt es Ausreißer, dann ergibt sich Folgendes:

$$1 - p = (1 - p^k)^N \quad (2.7)$$

Daraus ergibt sich die minimale Anzahl an Versuchen N , um mit einer vorgegebenen Wahrscheinlichkeit p ein Modell zu berechnen, das keine Ausreißer enthält:

$$N = \frac{\log(1 - p)}{\log(1 - p^k)} \quad (2.8)$$

Die Größe der Stichprobe k , aus der das Modell berechnet wird, soll möglichst klein sein. Wie aus Tabelle 2.1 hervorgeht, erhöht sich die Anzahl an Iterationen zur Findung des passenden Modells mit der Größe der Stichprobe. Angenommen mit einer Wahrscheinlichkeit von $p = 0,99$ wird ein Modell mit $k = 2$ Punkten berechnet und der Anteil der tolerierten Ausreißer beträgt 10%, werden somit nach der Tabelle drei Iterationen benötigt.

Tabelle 2.1: Anzahl der Iterationen N , um mit einer Wahrscheinlichkeit von $P = 0,99$ mindestens ein Modell keine Ausreißer hat. Tabelle aus [20]

Stichprobengröße k	Anteil der Ausreißer e					
	5%	10%	20%	30%	40%	50%
2	2	3	5	7	11	17
3	3	4	7	11	19	35
4	3	5	9	17	34	72
5	4	6	12	26	57	146
6	4	7	16	37	97	293
7	4	8	20	54	163	588
8	5	9	26	78	272	1177

Bei der Registrierung von Punktwolken kann RANSAC ein Modell mithilfe der Merkmale der Punkte bestimmen. Die Merkmale werden zuvor mit einem feature-based Algorithmus, wie dem FPFH aus Abschnitt 2.3.1, berechnet.

2.3.3 Iterative Closest Point Algorithmus (ICP)

Der Iterative Closest Point Algorithmus, kurz ICP, ist ein lokaler Registrierungsalgorithmus, der den euklidischen Abstand zwischen sich überschneidenden Bereichen zweier Punktwolken minimiert. Vor der Verwendung muss eine globale Registrierung, wie sie in 2.3.2 beschrieben wird, stattfinden, um die Punktwolken initial auszurichten. Die Beschreibung erfolgt nach [7, 29, 32, 33, 56].

Der ICP-Algorithmus wird in zwei Phasen unterteilt: die Korrespondenzbestimmung und die Abstandsminimierung, siehe Abbildung 2.6. Bei der Korrespondenzbestimmung werden Punktpaare zwischen zwei Punktwolken $\mathbf{P} = \mathbf{p}_i$ und $\mathbf{Q} = \mathbf{q}_i$ bestimmt, die einen kleinen Abstand zueinander haben, also nächstgelegene Nachbarn sind. Hierfür werden die Punkte der Punktwolken im k -dimensionalen Raum mit einem Binärsuchbaum wie dem kd -Baum strukturiert, um die Zeitkomplexität bei der Paarbildung gegenüber der Brute-Force-Methode zu reduzieren.

In der zweiten Phase werden die Abstände zwischen den Punktpaaren in einem iterativen Prozess durch die Transformation \mathbf{T} aus Rotation \mathbf{R} und Translation \mathbf{t} minimiert. Dafür werden die Punkte \mathbf{p}_i der Ausgangspunktwolke \mathbf{P} über $(\mathbf{T} \cdot \mathbf{p}_i - \mathbf{q}_i)$ den Punkten \mathbf{q}_i der Zielpunktwolke \mathbf{Q} angenähert. Für die Transformation wird für jeden Punkt des Ausgangssystems der Abstand zur Tangentialebene des korrespondierenden Punkts minimiert, siehe hierzu Abbildung 2.7. Dabei ist $\mathbf{p}_i = (p_{ix}, p_{iy}, p_{iz})^T$ ein Punkt der Aus-

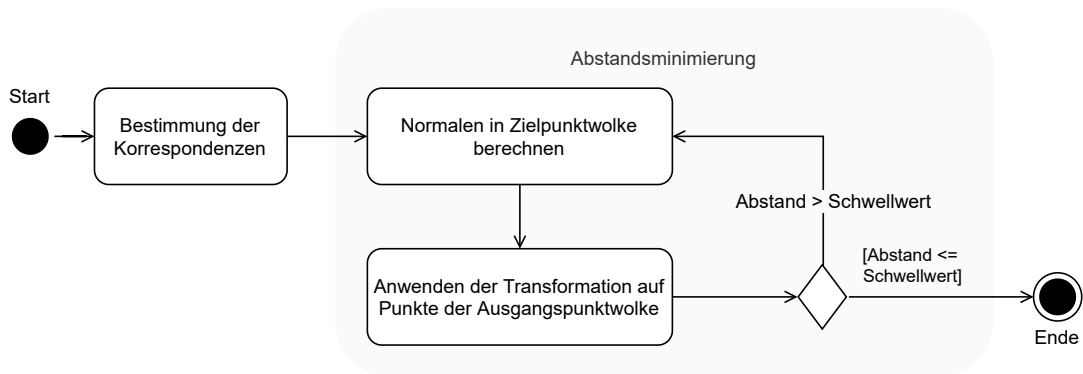


Abbildung 2.6: ICP-Ablaufdiagramm

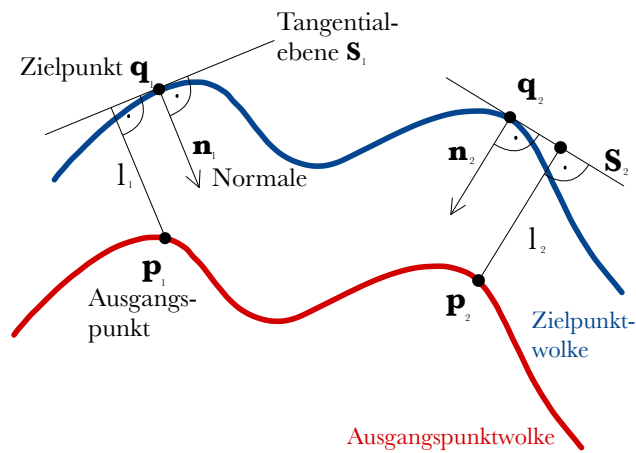


Abbildung 2.7: Reduzieren des Abstands mit dem Punkt-zu-Ebene-Verfahren - Abbildung nach [33]

gangspunktwolke und $\mathbf{q}_i = (q_{ix}, q_{iy}, q_{iz})^T$ der dazugehörige Punkt der Zielpunkt wolke. Für den Zielpunkt \mathbf{p}_i wird der Normalenvektor $\mathbf{n}_i = (n_{ix}, n_{iy}, n_{iz})$ bestimmt. Für die Transformation \mathbf{T} ergibt sich damit Folgendes:

$$\mathbf{T} = \arg \min_{\mathbf{T}} \sum_i ((\mathbf{T} \cdot \mathbf{p}_i - \mathbf{q}_i) \cdot \mathbf{n}_i)^2 \quad (2.9)$$

Der beschriebene Prozess wird bis zum Erreichen der Abbruchkriterien wiederholt. Als Abbruchkriterium kann eine maximale Anzahl an Iterationsschritten gelten oder die Unterschreitung des Schwellwerts, der den Abstand zwischen den Punktwolken festlegt. Der Abstand zwischen beiden Punktwolken minimal werden.

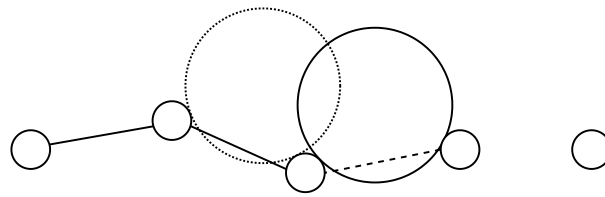


Abbildung 2.8: Ball Pivoting im 2D-Raum - Abbildung nach [54]

2.4 Oberflächenrekonstruktion

Für die Rekonstruktion der Oberfläche von Punktwolken kann der Ball Pivoting Algorithmus verwendet werden. Dieser rekonstruiert ein Dreiecksgitter (*englisch triangle mesh*) und wird nach [2] beschrieben. Angenommen ein Ball mit einem Radius r wird in einer Punktwolke \mathbf{P} , in dem die Abstände d der Punkte kleiner sind als $2 \cdot r$, so platziert, dass er drei Punkte berührt. Diese drei Punkte formen zusammen ein Dreieck. Der Ball wird weiter gerollt, wobei er immer mit zwei Punkten aus dem vorherigen Dreieck in Berührung bleibt. Stößt der Ball auf einen neuen Punkt wird wieder ein Dreieck geformt und schlussendlich entsteht ein Netz aus Dreiecken. In Abbildung 2.8 wird das Verfahren im 2D-Raum mit zwei anstelle von drei Punkten dargestellt.

3 Stand der Technik

In diesem Kapitel werden Anwendungsbeispiele für die Umsetzung von 3D-Scannern beschrieben. Dabei werden zunächst Anwendungsbereiche vorgestellt und im nächsten Schritt verwandte Veröffentlichungen und ihre Umsetzungsmethodik untersucht. Die Beispiele geben einen Einblick in den Stand der Technik zum Zeitpunkt der Erstellung der Masterarbeit.

3.1 Anwendungsbereiche

Die Anwendungsbereiche für 3D-Scanner sind vielfältig und reichen von der Unterhaltungselektronik bis in den medizinischen Bereich. So werden 3D-Scanner für die Einbindung von realen Personen in virtuelle Filmsets oder Computerspiele verwendet. Zudem werden sie für die Qualitätskontrolle von Bauteilen in der Industrie oder die Erfassung der Körpermaße von Personen im Fitnessstudio zur Messung des Trainingsfortschritts verwendet [41]. Im medizinischen Bereich unterstützen sie bei computergestützten Operationsverfahren wie zum Beispiel am Gehirn.

Das Unternehmen NEXR Technologies entwickelt 3D-Körperscanner, wie zum Beispiel in Abbildung 3.1 dargestellt, für die Erstellung von 3D-Modellen von Personen. In Zusammenarbeit mit dem Modeunternehmen H&M wurde der Scanner als virtuelle Umkleekabine getestet. Mit dem System erstellen Kund:innen Avatare und können diese in der App des Modeunternehmens für die Anprobe von Kleidung verwenden. Dafür werden die Körpermaße der Kund:innen erfasst, um die Kleidergröße zu bestimmen. Das Ziel des Systems soll unter anderem eine Reduktion der Retouren sein [40]. Mithilfe des 3D-Scanners können außerdem 3D-Figuren von realen Personen erstellt werden, die Teil einer virtuellen Realität werden. Die Figuren können mithilfe von Motion Capture Software Live animiert und auf virtuellen Events genutzt oder in Filmszenen eingebaut werden [42]. Der Scanner ist als geschlossene Kabine aufgebaut, in die sich die Benutzer:innen stellen. Für die Erfassung der Daten werden 132 Kameras und 36 Tiefensensoren verwendet, die

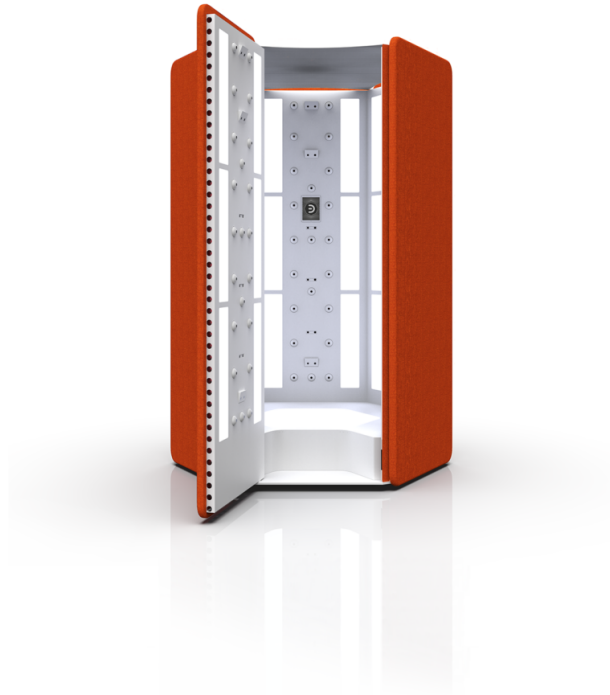


Abbildung 3.1: NEXR 3D-Körperscanner [43]

fest an der Kabinenwand verbaut sind und in verschiedenen Höhen eine 360° Perspektive der Person erfassen [44]. Das 3D-Modell wird mit einer künstlichen Intelligenz erstellt.

3.2 Verwandte Veröffentlichungen

In [53] wird ein 3D-Scanner für die Vermessung von Bauteilen in der Industrie verwendet. Um ein großes Bauteil als Ganzes zu erfassen, wird eine Kamera an einem Roboterarm befestigt und das Bauteil aus verschiedenen Perspektiven aufgenommen. Für die Aufnahme der Bilddaten wird das strukturierte Licht Verfahren verwendet, bei dem ein Projektor ein Muster auf die Oberfläche des aufzunehmenden Objekts projiziert, das von der Kamera aufgenommen wird. Für die Registrierung der Punktwolken werden zwei Ansätze verglichen: die Registrierung auf Grundlage von markanten Merkmalen der Oberfläche

des gescannten Objekts in Verbindung mit dem ICP-Algorithmus und die Registrierung von Punktwolken durch die Nachverfolgung der Aufnahmepositionen der Kamera. Mithilfe einer Microsoft Kinect wird in [69] ein 3D-Modell einer Person erstellt. Für die Aufnahme steht die Kamera an einem festen Ort und die Person dreht sich in unterschiedlichen Winkeln vor der Kamera. Die Rekonstruktion des 3D-Modells erfolgt anhand eines Körpermodells in Verbindung mit dem ICP-Algorithmus. Das Modell ermöglicht die Rekonstruktion eines 3D-Körpermodells auf Grundlage von unvollständigen Daten.

4 Anforderungsanalyse

Für die Entwicklung des 3D-Scanners wird eine Anforderungsanalyse durchgeführt, um das in Abschnitt 1.2 auf Seite 2 beschriebene Ziel umzusetzen. Dafür wird das zu entwickelnde System zunächst beschrieben und von der Umwelt abgegrenzt. Die Analyse der Stakeholder beschreibt welche Interessensgruppen einen direkten oder indirekten Einfluss auf das System haben. Hieraus ergeben sich die Anwendungsfälle, die die Verwendung beschreiben. Die gesetzlichen Vorschriften und Normen die bei der Entwicklung einzuhalten sind, werden im Anschluss erläutert. Auf dieser Grundlage werden die Anforderungen festgelegt, die die Eigenschaften und Fähigkeiten des fertigen Produkts definieren und die Grundlage für die Verifikation der Hard- und Software zum Ende der Arbeit legen.

4.1 Systembeschreibung

Das Gesamtsystem 3D-Scanner setzt sich aus einem Softwareteil und einem Hardwareteil zusammen, die Teil des Entwicklungsprozesses sind. Der Systemkontext wird in Abbildung 4.1 dargestellt. Die Hardware besteht aus einer Anordnung, die mithilfe eines Sensors Bilddaten von Personen aufnimmt. Die Software dient der Erstellung von 3D-Modellen und steuert die Hardware. Die Systemgrenze trennt das System von der Umwelt. Sie ist nicht Teil des Entwicklungsprozesses und nicht veränderbar. Über Schnittstellen interagiert das System mit Teilen des Systemkontexts:

- Kommunikation mit Benutzer:innen des Systems
- Slicer-Software zur Vorbereitung für den 3D-Drucker

Der 3D-Scanner wird in beheizten Innenräumen mit künstlicher oder natürlicher Beleuchtung verwendet. Die Lichtverhältnisse können einen Einfluss auf den Sensor des Systems haben. Außerhalb der Hardwaregrenze können sich Personen befinden die durch den Sensor erfasst werden. Die Verarbeitung der aufgenommenen Daten unterliegt nach der

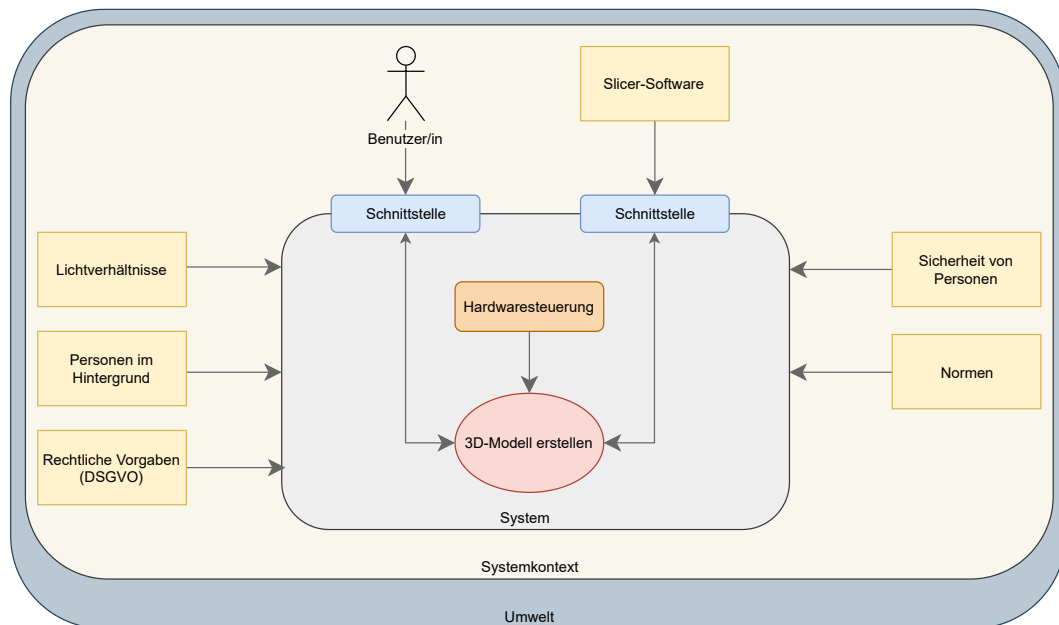


Abbildung 4.1: Systemkontext

Datenschutzgrundverordnung rechtlichen Vorgaben. Außerdem wird das System durch Normen, die die Sicherheit von Sensoren definieren, beeinflusst. Die Sicherheit von Benutzer:innen des Systems muss ebenfalls durch Anforderungen an die körperliche Unversehrtheit berücksichtigt werden.

4.2 Stakeholder

Nach [11] und [50] sind Stakeholder Interessensgruppen, die von dem System in der Entwicklung und durch den Einsatz direkt oder indirekt betroffen sind. Zu den Interessensgruppen gehören neben Personen auch Standards, Normen und Richtlinien. Aus ihren Interessen leiten sich Anforderungen ab, die bei der Entwicklung des Systems beachtet werden müssen.

Personen im Hintergrund

Personen, die sich nicht im Aufnahmebereich befinden und im Hintergrund der Bildaufnahme auftreten, haben ein Interesse daran, dass ihre Daten nicht personenbezogenen verarbeitet werden.

Rechtliche Vorgaben

Die Datenschutzgrundverordnung (DSGVO) stellt Anforderungen an die Verarbeitung von personenbezogenen Daten, die während des Entwicklungsprozesses von Personen im Umfeld des Scanners und im Aufnahmebereich erhoben werden.

Sicherheit von Personen

Anwender:innen des Systems haben ein Interesse daran das System ohne elektrische und mechanische Gefahren zu verwenden. Sofern ein Sensor mit Laser verwendet wird, sind Geährdungen für die Augen nach DIN EN 60825-1 bzw. IEC 60825-1 auszuschließen.

Anwender:in

Anwender:innen sind Personen, die den 3D-Scanner für die Erstellung eines 3D-Modells verwenden. Sie können gleichzeitig die Person sein, die die grafische Oberfläche verwendet und die Person, die gescannt wird. Gescannte Personen und die Oberfläche bedienende Person können ebenfalls zwei unabhängige Personen sein. Bei allen Anwender:innen des 3D-Scanners werden keine Vorkenntnisse im technischen Bereich vorausgesetzt. Die Bedienung der Software erfolgt über eine Bedienoberfläche und die Anwendung funktioniert fehlerfrei und zuverlässig. Das 3D-Modell weist gesichtstypische Merkmale wie Augen, Nasen, Mund und Ohren auf, die dem realen Modell entsprechen.

Auftraggeber

Auftraggeber ist Herr Prof. Dr.-Ing. Hensel als betreuender Professor und Erstprüfer. Die Definition der Anforderungen ist online in Abspracheterminen und schriftlich per E-Mail kommuniziert worden. Als Auftraggeber fordert Herr Hensel ein funktionsfähiges System, das ein dreidimensionales Modell eines Kopfes erstellt. Bei dem System handelt es sich um einen Prototypen, der durch weitere Projekte erweitert werden soll. Hierfür sind die Erkenntnisse und das Projekt zu dokumentieren. Weiterhin darf das zur Verfügung gestellte Budget nicht überschritten werden.

Einsatz bei Dienstleister:innen

Dienstleister:innen, die den 3D-Scanner für Veranstaltungen anbieten haben ein Interesse an einem System mit einer geringen Ausfallwahrscheinlichkeit. Für den Transport zu Veranstaltungen ist die Hardware des Scanners modular zu konstruieren. Die Gesamtheit der Bauteile müssen ein transportfähiges Maß und Gewicht aufweisen. Die Bauteile

sind so zu wählen, dass sie sich für einen wiederkehrenden Auf- und Abbau eignen. Für Instandhaltungs- und Wartungszwecke ist eine kommerzielle Wiederbeschaffung der verschleißanfälligsten Bauteile, wie zum Beispiel Schrauben, möglich.

Soft- und Hardware-Entwicklerin

Die Entwicklerin der Software und Hardware ist die Erstellerin dieser Arbeit. Ziel ist die Entwicklung eines funktionsfähigen und für die Anwender:innen sicheren Prototypen. Das System soll ein nutzerfreundliches und funktionelles Produktdesign haben. Sowohl Hard- als auch Software sollen als Ganzes oder in Teilen für Projekte anderer Studenten nutzbar sein.

Weiterentwicklung durch Studierende

Für die Weiterentwicklung und Verwendung von Teilen oder der ganzen Ausarbeitung ist eine strukturierte Dokumentation von Hardware und Software zur Einarbeitung in das System notwendig. Eine Weiterentwicklung für die Aufnahme von ganzen Personen sowie eine Erweiterung mit mehr als einem Sensor zu Datenaufnahme sind denkbar.

4.3 Anwendungsfälle

Im Folgenden werden die Anwendungsfälle für den 3D-Scanner beschrieben. Das UML-Anwendungsfalldiagramm in Abbildung 4.2 gibt eine Übersicht mit welchen Teilsystemen die unter 4.1 beschriebenen Schnittstellen kommunizieren.

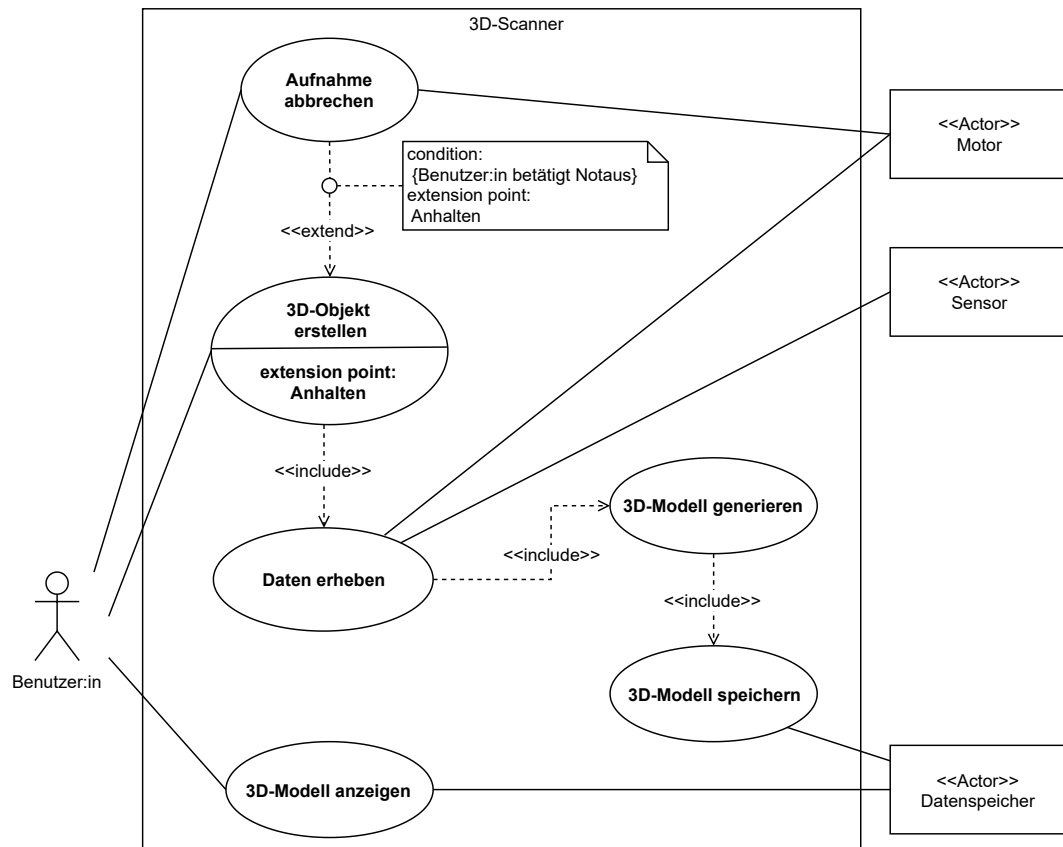


Abbildung 4.2: Anwendungsfall-Diagramm

Tabelle 4.1: Anwendungsfall: 3D-Objekt erstellen starten

Name	3D-Objekt erstellen
Kurzbeschreibung	Die Aufnahme der Sensordaten wird über die Schnittstelle gestartet.
Aktoren	Person die gescannt wird und/oder eine zweite Person
Vorbedingung	Die Software ist betriebsbereit und die Hardware befindet sich in der Ausgangsposition.
Ergebnis	Sensordaten für die Weiterverarbeitung
Beschreibung des Ablaufs	1) Start des Scanvorgangs über die Schnittstelle 2) Positionierung der Person vor dem Sensor. 3) Umkreisen der Person durch die Hardwaresteuerung 4) Die Hardware erreicht die Ausgangsposition.

Tabelle 4.2: Anwendungsfall: Aufnahme abbrechen

Name	Aufnahme abbrechen
Kurzbeschreibung	Die Aufnahme der Sensordaten wird vor der Fertigstellung über einen Notaus-Schalter abgebrochen.
Aktoren	Person die gescannt wird und/oder eine zweite Person
Vorbedingung	Die Sensordatenaufnahme wird ausgeführt.
Ergebnis	Die beweglichen Teile des Systems führen keine Bewegung mehr aus. Die Bildaufnahme wurde abgebrochen.
Beschreibung des Ablaufs	1) Eine Benutzer:in bricht die Aufnahme über einen Notaus-Schalter ab. 2) Weiterleitung des Signals an die Hardwaresteuerung. 3) Die Hardwaresteuerung stoppt bewegliche Teile. 4) Die Datenerhebung wird beendet. 5) Der Sensor fährt in die Ausgangsposition.

Tabelle 4.5: Anwendungsfall: 3D-Modell speichern

Name	3D-Modell speichern
Kurzbeschreibung	Das 3D-Modell wird von der Software gesichert.
Vorbedingung	Ein 3D-Modell wurde generiert.
Ergebnis	Eine Datei wurde erstellt.
Beschreibung des Ablaufs	Die Datei wird im Datenspeicher abgelegt.

Tabelle 4.3: Anwendungsfall: Daten erheben

Name	Daten erheben
Kurzbeschreibung	Der Sensor wird um die Person im Aufnahmebereich bewegt.
Vorbedingung	Die Aufnahme ist über die Schnittstelle von den Anwender:innen gestartet worden.
Ergebnis	Unverarbeitete Sensordaten
Beschreibung des Ablaufs	1) Die Daten werden von einem Sensor aufgenommen. 2) Die Daten werden gespeichert.

Tabelle 4.4: Anwendungsfall: 3D-Modell erstellen

Name	3D-Modell generieren
Kurzbeschreibung	Die Sensordaten werden aufbereitet und nicht relevante Bildbereiche sowie Rauschen entfernt. Im Anschluss werden die Daten zu einem 3D-Modell verarbeitet.
Vorbedingung	Es sind Daten vorhanden.
Ergebnis	Das 3D-Modell kann von einer Slicer-Software geöffnet werden.
Beschreibung des Ablaufs	1) Die Sensordaten werden verarbeitet, um Rauschen und nicht relevante Bildbereiche zu entfernen. 2) Aus den Sensordaten wird ein 3D-Modell generiert. 3) Das 3D-Modell wird in ein Datenformat konvertiert, das mit einer Slicer-Software kompatibel ist.

Tabelle 4.6: Anwendungsfall: 3D-Modell anzeigen

Name	3D-Modell anzeigen
Kurzbeschreibung	Die Benutzer:innen können sich die erstellten 3D-Modelle bei Bedarf anzeigen lassen.
Vorbedingung	Im Datenspeicher sind 3D-Modelle abgelegt.
Ergebnis	Das 3D-Modell wird angezeigt.
Beschreibung des Ablaufs	1) Die Benutzer:innen wählen über die Schnittstelle ein 3D-Modell aus, das angezeigt werden soll. 2) Das Modell wird auf dem Bildschirm angezeigt

4.4 Rechtliche Bedingungen

Durch den Sensor werden Körperstrukturen von Personen im Bereich des Scanners und im Hintergrund erfasst. Über diese Strukturen lassen sich abhängig von der Sensorart Rückschlüsse auf Personen ziehen. Es sind zwei Personengruppen zu unterscheiden die von dem System direkt oder indirekt erfasst werden: Personen, die sich aktiv in den Sensorbereich zum Erfassen von Daten stellen und Personen, die nicht aktiv an der Datenerhebung teilnehmen und sich außerhalb der Anordnung im Hintergrund bewegen. Bei der ersten Gruppe ist davon auszugehen, dass sie der Bildaufnahme und Verarbeitung ihrer Daten durch die Verwendung des Systems aktiv zugestimmt haben. Die TU Berlin hat 2020 eine Studie [21] zur Datenerfassung durch autonome Roboter veröffentlicht, die in Teilen auf das vorliegende System anwendbar ist. Im Kontext dieser Prototypenentwicklung und der anzunehmenden Einsatzgebiete der Hochschule kann vorausgesetzt werden, dass sich im Hintergrund des Sensorbereichs Personen mit technischen Vorkenntnissen aufhalten. Bei ihnen wird angenommen, dass sie eine Erfassung und Verarbeitung von Daten im Entwicklungszusammenhang erwarten. Wird die Anwendung auf Veranstaltungen eingesetzt, kann der Aufnahmebereich der Kamera ohne Einfluss auf die Funktionalität durch bauliche Maßnahmen eingeschränkt werden.

4.5 Normen

Der Einsatz von Sensoren mit Lasern im Bereich der Augen ist in der Norm DIN EN 60825-1 bzw. IEC 60825-1 definiert. Die Bundesanstalt für Arbeitsschutz und Arbeitsmedizin (BAuA) formuliert in ihrer Broschüre zum Umgang mit Laserstrahlung [6], dass Laserprodukte der Klassen 1 bei längerer Einwirkung auf das Auge keine Schäden verursachen. Das gilt auch dann, wenn sich Lupen oder Linsen im Strahlengang zwischen Auge und Lasereinrichtung befinden. Im vorliegenden Fall ist das auf Brillen anzuwenden. Laser der Klasse 1M stellen dann keine Gefahr für das Auge dar, wenn der Laserstrahl nicht von optischen Instrumenten verkleinert wird. Bei Lasereinrichtungen die in den Klassen 2 oder höher eingeordnet sind, steigt das Risiko von Augenschäden mit der Expositionsdauer. Als sicher wird aus den vorgelegten Gründen die Verwendung von Lasern der Klasse 1 eingeschätzt.

4.6 Anforderungen

Nachdem das Systemumfeld und die Stakeholder analysiert worden sind, werden daraus die Anforderungen abgeleitet. Funktionale und nicht funktionale Anforderungen werden in Klammern mit (F) beziehungsweise (NF) gekennzeichnet. Funktional sind Anforderungen, wenn sie die Funktionalität eines Systems beschreiben. Nicht funktional sind Anforderungen, wenn sie die Qualität und Eigenschaften definieren. Soweit nicht weiter gekennzeichnet sind die Anforderungen als *muss* umzusetzen. *Optionale*-Anforderungen müssen nicht umgesetzt werden und stellen eine Erweiterung des Systems dar.

4.6.1 Hardware

Dieser Abschnitt beschreibt die Anforderungen an die Hardware des 3D-Scanners. Sie ermöglicht die automatisierte Aufnahme von Sensordaten des Kopfes einer Person aus einer 360°-Perspektive im Aufnahmebereich. Die Anforderungen sind in die Teilbereiche Allgemein, Sicherheit der Hardware und Sensor unterteilt.

4.6.1.1 Allgemein

Im Folgenden werden die allgemeinen Anforderungen an das System definiert. Dazu gehören die Projektkosten, der Datenschutz und der Arbeitsbereich des Sensors.

HW-A 1. (NF) Das System ist geeignet für die Aufnahme von Köpfen von Personen mit einer Größe zwischen 1,56 m und 1,9 m.

Hinweis: Es wird angenommen, dass das System von männlichen, weiblichen und diversen Personen im Kontext der Hochschule verwendet wird. Die Altersgruppe beschränkt sich auf 18- bis 65-jährige Personen. Nach Angaben des statistischen Bundesamtes vom Mikrozensus 2017 wird die untere Grenze dieses Personenkreises mit einer durchschnittlichen Körpergröße von 1,65 m und die obere Grenze mit 1,81 m angegeben. Bei den Angaben handelt es sich um Durchschnittswerte, weshalb eine Abweichung von 5 % eingerechnet wird. Die untere Grenze wird damit auf 1,56 m festgelegt, die obere Grenze ist 1,90 m. [60]

HW-A 2. (F) Der Sensor bewegt sich mit einem elektrischen Antrieb bei der Aufnahme um die Person herum.

HW-A 3. (NF) Das System muss die Vorgaben der Datenschutzgrundverordnung für Personen im Hintergrund des Systems sowie für Personen, die das System aktiv nutzen, einhalten (siehe Abschnitt 4.4).

HW-A 4. (NF) Die maximalen Projektkosten von 500 € müssen eingehalten werden.

HW-A 5. (NF) Die Hardware muss so ausgewählt werden, dass sie in zukünftigen Projekten der Hochschule wiederverwendet werden kann.

HW-A 6. (NF) Für den Transport und die Lagerung ist die Hardware zerstörungsfrei auseinanderbaubar und besitzt auseinandergebaut maximal die Maße 0,6 m x 0,6 m x 1,2 m (L x B x H) und ein Gewicht von 31,5 kg.

*Hinweis: Die Angaben zu Maße und Gewicht entsprechen dem größten Paket, das über DHL versendet werden kann, ohne als Sperrgut deklariert zu werden.*¹

4.6.1.2 Sicherheit der Hardware

Das System wird zur Anfertigung von 3D-Scans durch Studierende der HAW und Laien zum Beispiel auf Veranstaltungen konzipiert. Laien sind in diesem Fall Personen ohne technischen Hintergrund, weshalb Kenntnisse im Umgang mit dem System für die Bedienung nicht vorausgesetzt werden können, da die Anwendung in der Regel einmalig ist. Im Umgang mit dem System wird Fehlerfreiheit und Zuverlässigkeit erwartet.

HW-SI 1. (NF) Eine Gefährdung der Augen durch Infrarot oder Laser muss ausgeschlossen werden. Es ist die DIN EN 60825-1 oder IEC 60825-1 (VDE 0837-1) anzuwenden.

HW-SI 2. (NF) Mechanische Gefährdungen durch Quetschungen, Stoßen, Stolpern oder bewegte Teile sind auszuschließen oder Schutzeinrichtungen zu verwenden.

¹<https://www.dhl.de/de/privatkunden/pakete-versenden/deutschlandweit-versenden/preise-national.html> - Zugriffsdatum 23.03.2022

HW-SI 3. (NF) Gefahren durch Kanten, Spitzen oder Oberflächenrauigkeit sind für die Handhabung durch Personen auszuschließen oder geeignete Schutzeinrichtungen zu verwenden.

HW-SI 4. (NF) Eine elektrische Gefährdung von Personen bei der Bedienung des Systems ist auszuschließen.

HW-SI 5. (F) Elektrisch bewegte Teile sind abschaltbar.

4.6.1.3 Sensor

Für die Erstellung von dreidimensionalen Modellen von realen Objekten ist ein Sensor notwendig, der Tiefeninformationen erfasst und für die Verarbeitung bereitstellt. Der Arbeitsbereich des Sensors muss bei dieser Anwendung für Nahaufnahmen geeignet sein. Nach Anforderung 3 ist dieser für Personen bis 1,81 m auszulegen. Für die vollständige Erfassung von Strukturen wie Nase oder Ohren werden Aufnahmen in einem Abstand d oberhalb und unterhalb des Kopfes gemacht, siehe Abbildung 4.3 a). Der Abstand der Kamera zum Gesicht wird seitlich durch die Schulterbreite begrenzt. Nach [12] beträgt die Schulterbreite von Männern in der Altersgruppe zwischen 18 und 65 Jahren durchschnittlich maximal 0,525 m. Abzüglich der Breite des Kopfes von 0,165 m, bleibt eine Differenz von 0,180 m pro Seite, siehe Abbildung 4.3 b). Frauen haben durchschnittlich eine geringere Schulterbreite. Abzüglich der Kopfbreite ergibt sich eine Differenz von 0,167 m. Die Angaben nach [12] gelten für unbedeckte Menschen. Unter Einbeziehung von Kleidung und unter Einbeziehung einer Toleranz, da es sich nur um Durchschnittswerte handelt, wird der minimale Abstand zwischen Kamera und Gesicht auf 0,3 m festgelegt. Die minimale obere Grenze berechnet sich aus der maximalen Körpergröße 1,9 m addiert mit dem Mindestabstand zwischen Gesicht und Kamera. Somit ergibt sich ein Arbeitsbereich zwischen 0,3 m bis 2,2 m.

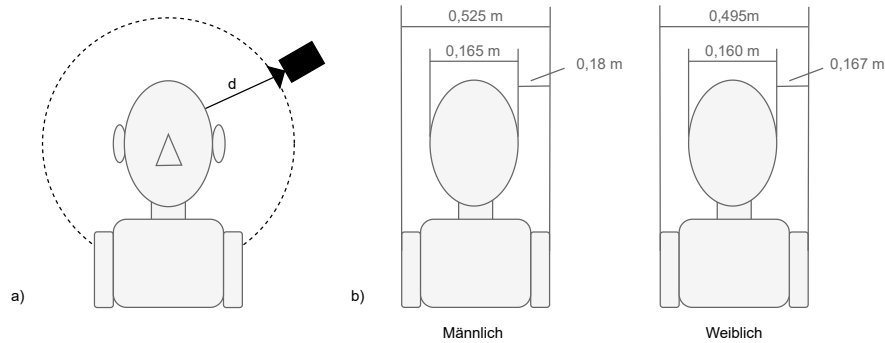


Abbildung 4.3: Abstand des Sensors zur Person

HW-SN 1. (NF) Der Arbeitsbereich des Sensors liegt zwischen 0,3 m und 2,2 m. Der Arbeitsbereich definiert den Abstand zwischen Objekt und Sensor und gibt die untere sowie die obere Grenze dieses Abstands an.

HW-SN 2. (F) Der Sensor muss für Aufnahmen im Innenbereich geeignet sein.

Die folgende Anforderung an den Sensor kann optional umgesetzt werden:

HW-SN 3. (F) Integration weiterer Sensoren in die Anordnung für die Beschleunigung der Datenaufnahme.

4.6.2 Software

Die Bildaufnahme und -verarbeitung zum fertigen 3D-Modell soll nach dem Start durch die Anwender:in über die Bedienoberfläche automatisiert erfolgen.

SW 1. (NF) Die erstellte Software ist auf dem Windows-Betriebssystem ab Version 10 lauffähig.

SW 2. (NF) Die Software wird modular programmiert, sodass Teile des Programms weiterverwendet und weiterentwickelt werden können.

SW 3. (F) Für die Interaktion zwischen Anwender:innen und dem System wird eine Bedienoberfläche erstellt.

SW 4. (F) Über die Bedienoberfläche werden die Aufnahmen gestartet.

SW 5. (F) Die Aufnahme kann bei Bedarf über die Bedienoberfläche vor Fertigstellung der Datenerfassung beendet werden.

SW 6. (F) Die Bedienoberfläche wird über Maus und/oder Tastatur gesteuert.

SW 7. (F) Vor der Bildaufnahme startet ein Countdown von 10 Sekunden damit eine Aufnahme von einer Person alleine durchgeführt werden kann. Diese Option kann durch die Anwender:innen übersprungen oder abgewählt werden.

SW 8. (F) Die Software muss aus Daten mit Tiefeninformationen, die aus einer 360° Perspektive aufgenommen worden sind, ein dreidimensionales Modell eines Kopfes erstellen.

SW 9. (F) Das 3D-Modell muss in einem neutralen CAD-Format gespeichert werden damit es in einer kostenlosen CAD-Software geöffnet werden kann. Zu den neutralen Formaten gehören beispielsweise *.stp*, *.stl* oder *.obj* [5].

SW 10. (F) Es können beliebig viele Aufnahmen von Personen durchgeführt werden. Die Verwaltung des Speichers erfolgt durch das Betriebssystem.

Zusätzlich können folgende Punkte umgesetzt werden. Die Anforderungen sind optional:

SW 11. (F) Im 3D-Modell werden Texturen wie Haare dargestellt.

SW 12. (F) Sensordaten können durch die Benutzer:innen über die grafische Oberfläche gelöscht werden.

SW 13. (F) Nach Abschluss der Datenerhebung können die Benutzer:innen auswählen, ob die Datenverarbeitung gestartet werden soll oder ob die aufgenommenen Daten verworfen werden.

SW 14. (F) Das 3D-Modell wird nach Fertigstellung angezeigt.

5 Konzept

Für die Entwicklung des 3D-Scanners wird ein Konzept zur Umsetzung entwickelt, das die Anforderungen aus Kapitel 4 umsetzt. Im Abschnitt Hardware wird die Auswahl des Sensors und der Hardwarekomponenten für die Sensorhalterung beschrieben. Der Abschnitt Software beschäftigt sich mit der Auswahl der Algorithmen für den Softwareentwurf.

5.1 Hardware

Nach der Materialauswahl für die Sensorhalterung erfolgt die Auswahl des Sensors.

5.1.1 Sensorhalterung

Wie aus der Zielbeschreibung in Abschnitt 1.2 auf Seite 2 hervorgeht, bewegt sich der Sensor um die Person herum. Für die Befestigung der Sensoren und zur Aufnahme von Bilddaten aus verschiedenen Perspektiven wird folglich eine bewegliche mechanische Konstruktion benötigt. Nach Anforderung HW-A 5 muss die Hardware für zukünftige Projekte an der HAW wiederverwendbar sein. Daher eignen sich Material, das einen geringen Verschleiß beim Auf- und Umbau hat und vielseitig eingesetzt werden kann. Ein Aufbau aus Holz eignet sich unter diesen Voraussetzungen nicht, da Bohrungen irreversibel sind und die Anpassungsfähigkeit für neue Projekte mindert. Vielmehr eignet sich für den Aufbau Metallstangen und ein System, das nach dem Baukastenprinzip funktioniert. Um die Kompatibilität mit vorhandenem Material aus vergangenen Projekten herzustellen, ist die Masterarbeit von Malte Dittmar herangezogen worden. In dieser werden Aluprofile zum Aufbau eines Teststands verwendet [13]. Ein Vorteil bei der Verwendung von Aluprofilen ist, neben dem geringen Gewicht, die reversible Befestigungsmöglichkeit in der Nut, die eine zerstörungsfreie und verschleißarme Verwendung des Materials ermöglicht. In der Prototypenentwicklung, bei der verschiedene Konzepte getestet werden, ist das ein großer Vorteil. Auch für einen Einsatz bei Dienstleistern ist von einem häufigen Auf- und

Abbau auf Veranstaltungen auszugehen. Dabei ist es vorteilhaft, dass dafür handelsübliches Werkzeug verwendet werden kann. Aluprofile ermöglichen außerdem eine Vielfalt an Gestaltungsmöglichkeiten, da sie in vielen beliebigen Größen beschaffbar und durch Verbinder erweiterbar sind. Zudem erfüllen sie die Anforderungen zur Reduzierung des Verletzungsrisikos für Benutzer:innen nach HW-SI 3 aufgrund ihrer glatten Oberfläche. Durch die vielseitige Einsetzbarkeit und der Wiederverwendbarkeit in anderen Arbeiten werden für diese Masterarbeit Aluprofile ausgewählt.

5.1.2 Sensorauswahl

5.1.2.1 Vergleich von Stereokameras und Lidar

Für die Aufnahme von Tiefeninformationen werden die Vor- und Nachteile von Stereokameras und Lidar verglichen. Der Vergleich erfolgt anhand von Produkten verschiedener Hersteller. Die Kriterien werden durch die Anforderungen aus dem Abschnitt 4.6.1 ab Seite 24 festgelegt. Nach einer Recherche zu den verfügbaren Produkte, werden für den Vergleich die Stereokameras von Intel aus der RealSense D4-Reihe und der Lidar L515, sowie die Stereokamera ZED 2 von Stereo Labs und die Microsoft Azure Kinect miteinander verglichen. Alle Produkte werden in Tabelle 5.5 dargestellt. Zur Bewertung werden die Kriterien mit „+“ für erfüllt, „o“ für teilweise erfüllt und mit „-“ für nicht erfüllt verwendet.

Anwendungsbereich

Der 3D-Scanner soll nach Anforderung HW-SN 2 in geschlossenen Räumen verwendet werden. Alle Kameras erfüllen diese Anforderung.

Arbeitsbereich in z-Richtung

Der Arbeitsbereich der Sensoren in z-Richtung wird in Anforderung HW-SN 1 festgelegt. Alle Sensoren können den Mindestabstand zur Objektoberfläche von 0,3 m einhalten oder bieten in drei von vier Fällen (D4-Reihe, L515 und Azure Kinect) die Möglichkeit die Kamera näher an die Oberfläche zu bringen, siehe Tabelle 5.1. Die in der Anforderung festgelegte obere Grenze von 2,2 m kann ebenfalls von allen Kameras eingehalten werden.

Tabelle 5.1: Vergleich der Arbeitsbereiche in z-Richtung nach [23, 26, 63, 38]

Stereokameras		Lidar	
D4-Reihe	ZED 2	L515	Azure Kinect
0,2 m - 10 m	0,3 m - 20 m	0,25 m - 9 m	0,25 m - 2,21 m

Schnittstelle

Alle Kameras verwenden gängige USB-Schnittstellen und verfügen über herstellereigene Software Development Kits (SDK). Über die SDKs können Kameraeinstellungen wie die Aufnahmeauflösung eingestellt werden oder die Kameras kalibriert werden. Außerdem können Bilddaten angezeigt und mit Bibliotheken wie OpenCV bearbeitet werden. [27, 38, 61]. Die Intel Kameras können untereinander über eine eigene Schnittstelle verbunden werden, um gleichzeitig Punktwolken aus verschiedenen Perspektiven aufzunehmen [24, 25]. Diese Möglichkeit wird über den Sync Port bei der Azure Kinect ebenfalls gegeben [37]. Eine Synchronisierung mehrerer Kameras ist bei der ZED 2 ebenfalls möglich. Diese erfolgt über einen Desktop PC. Eine separate Schnittstelle besitzt die Kamera dafür nicht [62].

Tabelle 5.2: Vergleich der Arbeitsbereich Schnittstellen [23, 26, 63, 38]

Stereokameras		Lidar	
D4-Reihe	ZED 2	L515	Azure Kinect
USB 3.1 Gen. 1	USB 3.0	USB 3.1 Gen. 1	USB 3.1 Gen. 2

Wiederverwendbarkeit

In der Masterarbeit von Malte Dittmar [13] ist eine Kamera der D4-Reihe verwendet worden. Über die benannten Schnittstellen ist eine gleichzeitige Verwendung beider Kameras für eine Anwendung in einem Folgeprojekt möglich. Eine Kombination mit den anderen Kameras ist aufgrund der fehlenden Schnittstelle aufwändiger.

Sicherheit

Die Intel Kameras, die Azure Kinect sowie die ZED 2 sind laut Herstellerangaben für die Verwendung am Menschen geeignet. Die ZED 2 verwendet keinen IR-Sensor. Die IR-Sensoren der Intel Kameras und der Azure Kinect verwenden Laser der Klasse 1 nach IEC 60825-1:2014 [23, 26, 36]. Alle Kameras erfüllen damit die Anforderung HW-SI 1.

Anschaffungskosten

Der Preis für die ZED 2 beträgt zum Zeitpunkt der Anschaffung im Mai 2021 etwa 540 €. Die in Anforderung HW-A 4 festgelegte Grenze von 500 € wird durch die Anschaffung überschritten. Neben den Anschaffungskosten für die Kamera entfallen Kosten auf die Beschaffung der Aluprofile. Die Kamera kommt daher für das Projekt nicht in Frage.

Die Azure Kinect wird im Mai 2021 von Microsoft, Amazon oder Reichelt nicht verkauft. Im Juli 2022 kann die Kamera wieder für 410 € über den Microsoft Store beschafft werden [64]. Die Kamera kommt aufgrund der fehlenden Kompatibilität mit der Kamera aus der vorangegangenen Masterarbeit, der hohen Kosten und der fehlenden Verfügbarkeit zum Zeitpunkt der Bestellung nicht in Frage.

Im Mai 2021 kosten die Intel RealSense Kameras der D4-Reihe zwischen 150 und 250 € [9]. Die Intel RealSense L515 kostet etwa 350,00 € [9]. Bei Anschaffung einer Kamera aus der D4-Reihe verbleibt ausreichend Budget für die Anschaffung von Hardwarebauteilen.

Tabelle 5.3: Vergleich der Anschaffungskosten nach [9, 64]

Stereokameras		Lidar	
D4-Reihe	ZED 2	L515	Azure Kinect
150,00 € - 250,00 €	ca. 540,00 €	ca. 350,00 €	Nicht verfügbar

Der Vergleich der Kameras in Tabelle 5.3 zeigt, dass die Kameras der D4-Reihe alle Anforderungen nach Tabelle 5.4 erfüllen. Sie liegen am Besten im Budget, sodass weitere Hardwarebauteile für die Kamerahalterung angeschafft werden können. Im nächsten Abschnitt wird ein Kameramodell der D4-Reihe ausgewählt.

Tabelle 5.4: Vergleich der Kameramodelle

	Stereokameras		Lidar	
	D4-Reihe	ZED 2	L515	Azure Kinect
Anwendungsbereich	+	+	+	+
Arbeitsbereich	+	+	+	+
Schnittstelle	+	+	+	+
Wiederverwendbarkeit	+	o	o	o
Verwendung am Menschen	+	+	+	+
Kosten	+	-	o	-

5.1.2.2 Vergleich von Tiefenkameras

In diesem Abschnitt werden drei Stereokameras der Intel RealSense Produktfamilie verglichen. Für den Vergleich werden Daten der Kameras in Tabelle 5.5 dargestellt, die aus dem Datenblatt [26] entnommen worden sind. In Tabelle 5.6 erfolgt die Bewertung der Kameras anhand der im folgenden beschriebenen Kriterien.

Tabelle 5.5: Vergleich der Intel RealSense D400-Modelle nach [26]

	D415	D435	D455
Arbeitsbereich	0,3 m - 10 m	0,2 m - 10 m	0,4 m - 10 m
Idealer Arbeitsbereich	0,5 m - 3 m	0,3 m - 3 m	0,6 m - 6 m
Kamerasichtfeld (H×V)	69.4° × 42.5° (±3°)	86° × 57° (±3°)	87° × 58° (±3°)
Baseline	55 mm	50 mm	95 mm
Minimaler Abstand (640×480)	310 mm	175 mm	320 mm
Minimaler Abstand (1280×720)	450 mm	280 mm	520 mm
Tiefenfehler	<2% bis 2 m	<2% bis 2 m	<2% bis 4 m

Alle Kameras besitzen eine Farbkamera und einen Infrarotprojektor. Die Modelle werden über das SDK von den Betriebssystemen Windows, Linux und macOS unterstützt. Über die Implementierung des SDKs in den Frameworks Open3D und OpenCV können die Bilddaten verarbeitet werden. Das SDK unterstützt die Programmiersprachen Python und C++. Nach Anforderung HW-SN 1 soll die Kamera einen minimalen Abstand von 30 cm zur Oberfläche des Objekts haben können. Die D455 erfüllt diese Anforderung mit einem Arbeitsbereich ab 0,4 m nicht. Der Arbeitsbereich der Kameras ist für bis zu 10 m in z-Richtung angegeben. Mit steigender Entfernung erhöht sich der Tiefenfehler der Kameras und die Genauigkeit der Aufnahme nimmt ab. Der Tiefenfehler der D415 und D435 ist bis 2 m Entfernung kleiner als 2 %. Der ideale Arbeitsbereich wird bis 3 m angegeben. Die D455 ist für eine größere Entfernung nutzbar. Der ideale Arbeitsbereich wird mit bis zu 6 m angegeben. Ein Tiefenfehler von weniger als 2 % wird bei 4 m erreicht. Das Kamerasichtfeld der D415 in x- und y-Richtung ist kleiner, als das der anderen Kameras. Für ein 360°-Modell muss die Kamera mehr Aufnahmen machen. Bei gleicher Anzahl an Pixeln hat die D415 aufgrund des kleineren Sichtfelds eine höhere

Pixeldichte pro Grad. Die Auflösung ist damit höher. Das ist ein Vorteil bei feineren Strukturen, wie sie im Gesicht oder an den Ohren vorkommen. Das Tiefenrauschen der D435 ist etwa zweimal so hoch, wie das der anderen Modelle. Der Tiefenfehler D_{RMS} berechnet sich nach [28] indem der Abstand zum Objekt d quadriert, mit der Anzahl der Subpixel S multipliziert und durch die Brennweite der Kamera f multipliziert mit der Baseline B , die den Abstand zwischen den beiden Kameras angibt, geteilt wird:

$$D_{RMS} = \frac{d^2 \cdot S}{f \cdot B} \quad (5.1)$$

Die Brennweite berechnet sich mit der Auflösung in x-Richtung X_{Res} und dem horizontalen Sichtfeld der Kamera H_{FOV} wie folgt:

$$f = \frac{1}{2} \cdot \frac{X_{Res}}{\tan\left(\frac{H_{FOV}}{2}\right)} \quad (5.2)$$

Das höhere Tiefenrauschen bei der D435 ist bedingt durch die kleinere Baseline und die oben beschriebene geringere Auflösung. Die D455 kann das durch die größere Baseline ausgleichen. Das Rauschen erzeugt eine bergige Struktur bei der Aufnahme von glatten Oberflächen. Die Vorverarbeitung der Punktwolke ist aufwändiger. Bei feinen Strukturen ist mit Datenverlust zu rechnen. Die D415 und die D455 sind gegenüber der D435 in diesem Punkt zu bevorzugen.

Trotz des kleineren Kamerasichtfelds ist die D415 aufgrund der höheren Auflösung den anderen Modellen vorzuziehen und wird für den 3D-Scanner verwendet.

Tabelle 5.6: Vergleich Intel RealSense

	D415	D435	D455
Idealer Aufnahmebereich	+	+	+
Kamerasichtfeld	o	+	+
Auflösung	+	o	o
Abstand zum Objekt	+	+	+
Tiefenrauschen	+	-	+

5.2 Software

In diesem Abschnitt wird das Softwarekonzept anhand der Anforderungen aus Abschnitt 4.6.2 entwickelt. Nach der Auswahl der Entwicklungsumgebung wird der Gesamttablauf des Programms vorgestellt. Die vorgestellten Programmteile werden im Folgenden erläutert.

5.2.1 Entwicklungsumgebung

Die für die Entwicklung der Software verwendete Programmiersprache sowie die Bibliotheken und Frameworks werden in diesem Abschnitt ausgewählt. Die Auswahl erfolgt auf Grundlage der in Abschnitt 5.1.2.2 ausgewählten Intel RealSense D415 Kamera. Intel bietet für die Entwicklung ein Software Development Kit an, das mit Python oder C++ verwendet werden kann. Das SDK enthält eine API, über die auf die Einstellungen der Kamera und auf die Kamerastreams zugegriffen werden kann. Es ist außerdem in die Open-Source Bibliothek Open3D integriert. Open3D bietet Algorithmen zur 3D-Bildverarbeitung an. Das SDK und Open3D kann mit C++ und Python verwendet werden.

Der PYPL-Index (Popularity of Programming Language) gibt an, wie oft Tutorials in einer Programmiersprache bei Google gesucht worden sind. Die Daten werden über die Google Trends ermittelt. Im Jahr 2022 war Python mit 31,84 % auf dem ersten Platz. C/C++ liegt mit 4,21 % auf dem sechsten Platz [19]. Im Vergleich zu Python ist die Programmausführung in C++ schneller, da es kompiliert und in Maschinensprache ausgeführt wird. Für Python kann das durch die Verwendung von Softwarepaketen wie `numba`, die den Python Code in Maschinensprache übersetzen, beschleunigt werden. Python hingegen überzeugt mit seiner einfachen Syntax und der großen Anzahl an kostenlosen Bibliotheken. Aus diesem Grund wird Python ausgewählt.

Neben der Erstellung eines 3D-Modells soll nach Anforderung SW 3 eine Bedienoberfläche programmiert werden. In Python können Frameworks wie `kivy`, `tkinter` oder `Qt` verwendet werden. `Qt` bietet mit dem `Qt-Designer` eine Entwicklungsumgebung, in der Benutzeroberflächen gestaltet werden können. Für Bildungszwecke ist ein vollständiger Zugang zu allen `Qt`-Produkten möglich [67]. Der `Qt Designer` ist jedoch auch in der Community Version unter der General Public License Version 3, kurz `LGPLv3`, verfügbar und ermöglicht damit unter Verwendung von Open Source Bibliotheken die Entwicklung einer Bedienoberfläche. Eine Vollversion mit allen `Qt`-Modulen wird nicht benötigt

[66]. Aufgrund der vielen Möglichkeiten und der einfachen Bedienung wird Qt verwendet.

5.2.2 Gesamtablauf

In diesem Teilabschnitt wird das Konzept vom Start des Programms bis zum 3D-Modell beschrieben. Der Gesamtablauf ist in Abbildung 5.1 dargestellt. Das System ist betriebsbereit und wird von der Benutzer:in über die Bedienoberfläche gestartet. Die Kamera beginnt mit der Aufnahme der Punktwolken, die durch die Benutzer:in jederzeit abgebrochen werden kann. Ist das der Fall, endet der Prozess vorzeitig. Wird die Aufnahme fortgeführt und abgeschlossen, kann durch die Benutzer:in ausgewählt werden, ob die Aufnahme gespeichert oder verworfen werden soll. Wird die Aufnahme verworfen, werden die Punktwolken gelöscht und der Prozess endet. Wird die Aufnahme gespeichert, werden die Punktwolken verarbeitet und ein 3D-Modell generiert und am Speicherort der aufgenommenen Punktwolken abgelegt.

Die Bildverarbeitung erstellt aus den rohen Punktwolken ein 3D-Modell. Sie wird nach der Aufnahme der Punktwolken automatisch gestartet, wenn die Punktwolken nicht verworfen worden sind. Auf die Bildverarbeitung kann über die Bedienoberfläche kein Einfluss genommen werden.

5.2.3 Bedienoberfläche

Die Bedienoberfläche ist die Schnittstelle zwischen den Benutzer:innen und dem 3D-Scanner. Grundlage sind die Anforderungen SW 3 bis SW 7. Über die Bedienoberfläche können die Benutzer:innen auswählen, ob sie ein 3D-Modell erstellen oder anzeigen lassen wollen. Wird der Punkt *3D-Modell erstellen* ausgewählt, folgt der in Abbildung 5.2 dargestellte Ablauf. Zunächst wird eine Bildvorschau des Aufnahmebereichs angezeigt, damit sich die Nutzer:innen positionieren können. Anschließend wird die Aufnahme über die Bedienoberfläche gestartet. Nach einem Countdown werden die Punktwolken aufgenommen. Ist die Aufnahme abgeschlossen, wird auf dem Bildschirm eine Fertigmeldung angezeigt. Die aufgenommenen Punktwolken und die fertigen 3D-Modelle können über den zweiten Menüpunkt angezeigt werden.

Über die Bedienoberfläche können außerdem die erstellten 3D-Modelle angezeigt werden. Nach der Auswahl der Bildvorschau durch die Benutzer:in folgt der in Abbildung 5.3

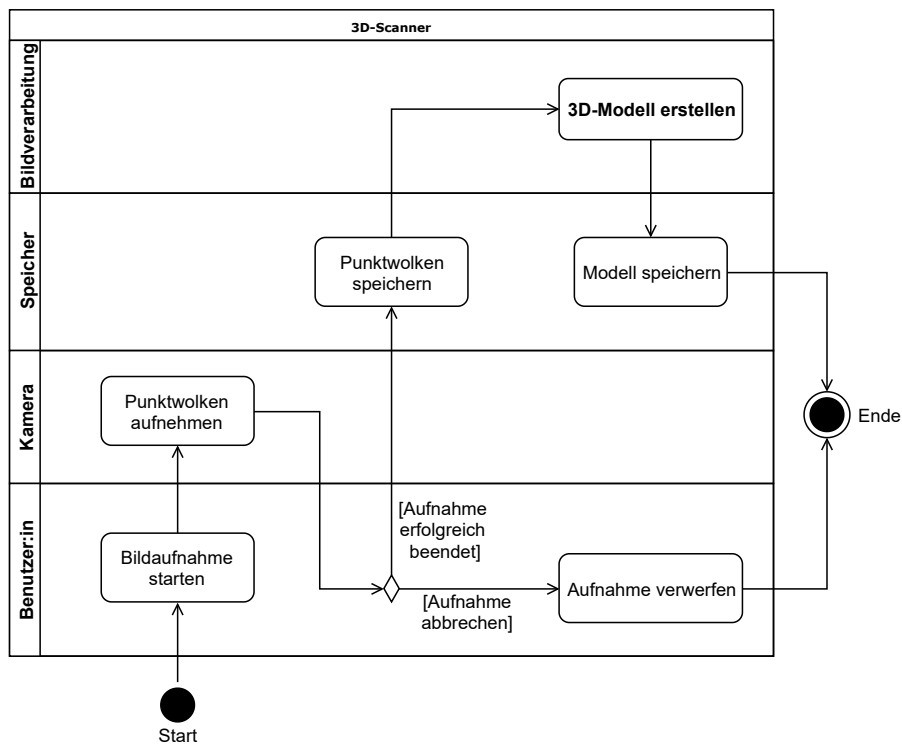


Abbildung 5.1: Gesamtablauf

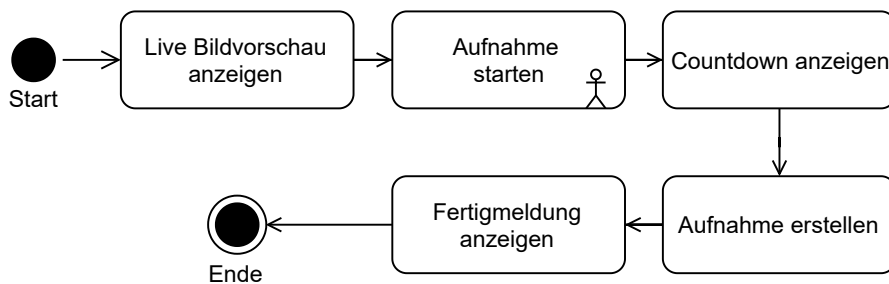


Abbildung 5.2: Aktivitätsdiagramm der Bedienoberfläche - 3D-Modell erstellen

dargestellte Ablauf. Das 3D-Modell wird angezeigt, wenn eine Datei ausgewählt wird. Durch das schließen des Fensters wird die Ansicht beendet.

5.2.4 Aufnahme der Punktwolken

Für die Bildaufnahme wird die Kamera schrittweise um die Person im Aufnahmebereich bewegt. Die Kamerahalterung stoppt für jede Bildaufnahme und bewegt die Kamera im

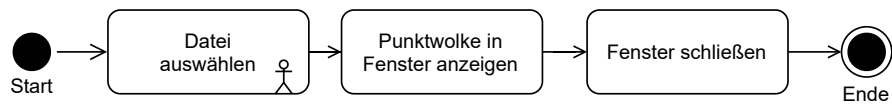


Abbildung 5.3: Aktivitätsdiagramm der Bedienoberfläche - Bildvorschau

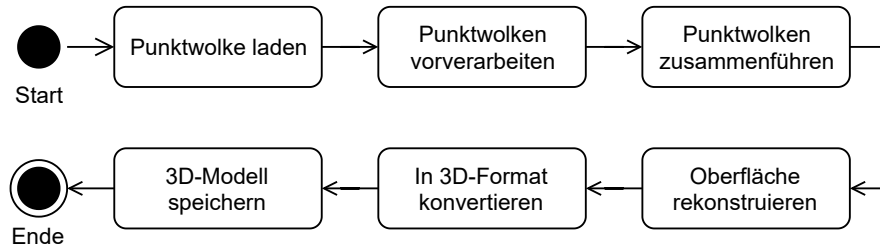


Abbildung 5.4: Aktivitätsdiagramm der Bildverarbeitung

Anschluss einen definierten Winkel weiter. Der Winkel ist abhängig von dem in x- und y-Richtung aufgespannten Aufnahmebereich der Kamera und dem für die Registrierung notwendigen Überschneidungsbereich zwischen zwei Punktwolken. Die Punktwolken werden nach jeder Aufnahme auf der Festplatte gespeichert.

5.2.5 Bildverarbeitung

In diesem Abschnitt wird der Ablauf der Bildverarbeitung beschrieben, der in Abbildung 5.4 dargestellt ist. Zunächst werden die Punktwolken geöffnet und der Hintergrund und vorhandenes Rauschen in der Punktwolkenvorverarbeitung entfernt, da für das 3D-Modell nur der Vordergrund der Aufnahme benötigt wird. Anschließend werden die verarbeiteten Punktwolken in der Registrierung, wie in Anforderung SW 8 gefordert, zu einem 3D-Modell zusammengeführt und es wird eine Oberfläche rekonstruiert, um das 3D-Modell in einem der Anforderung SW 9 entsprechendem 3D-Format zu speichern.

Im Folgenden werden die Konzepte für die Punktwolkenvorverarbeitung, die Registrierung und die Oberflächenrekonstruktion vorgestellt.

5.2.5.1 Punktwolkenvorverarbeitung

Die aufgenommenen Punktwolken enthalten Punkte, die für die Registrierung nicht relevant sind. Im Folgenden wird ein Ansatz ausgewählt mit dem die Punkte entfernt werden sollen.

Auswahl des Ansatzes

Die Kamera bewegt sich in der x - z -Ebene in einer Kreisbahn mit dem Radius r um die Person im Aufnahmebereich herum, siehe Abbildung 5.5 a. Der Abstand der Kamera zur Person richtet sich dabei nach Anforderung HW-SN 1, in der der Arbeitsbereich des Sensors definiert wird. Bei der Aufnahme erfasst die Kamera nur Daten von Oberflächen, die aus Sicht der Kamera nicht verdeckt werden. Bereiche, die sich im Schatten der im Vordergrund stehenden Person befinden, werden nicht erfasst, siehe Abbildung 5.5 a und b. Für die Entfernung dieser Daten aus der Punktwolke werden im Folgenden zwei Ansätze verglichen: Entfernen des Hintergrunds ab einem definierten Abstand sowie die Entfernung des Hintergrunds auf Grundlage eines Algorithmus zur Clusteranalyse.

Ansatz 1: Abstandsdefinition

Voraussetzung für dieses Verfahren ist eine kalibrierte Kamera und eine geringe Messabweichung der Kamera zur genauen Erfassung der Abstände zur Person. Die Kamera wird in der beschriebenen mechanischen Konstruktion montiert. Die zu scannende Person positioniert sich innerhalb des Aufnahmebereichs. Punkte, die sich außerhalb eines festgelegten Bereichs in der x -, y - und z -Ebene befinden, werden entfernt. Bei einer frontalen Aufnahme des Gesichts wird ab dem Abstand d von der Kamera zum Mittelpunkt des Kopfes der Hintergrund, wie in den Abbildungen 5.5a und 5.5b, entfernt. Der erhaltene Bereich ist grün dargestellt, der entfernte Bereich ist rot. Ebenso werden die Bereiche oberhalb und unterhalb, sowie seitlich des Kopfes entfernt, wie in Abbildung 5.5c dargestellt.

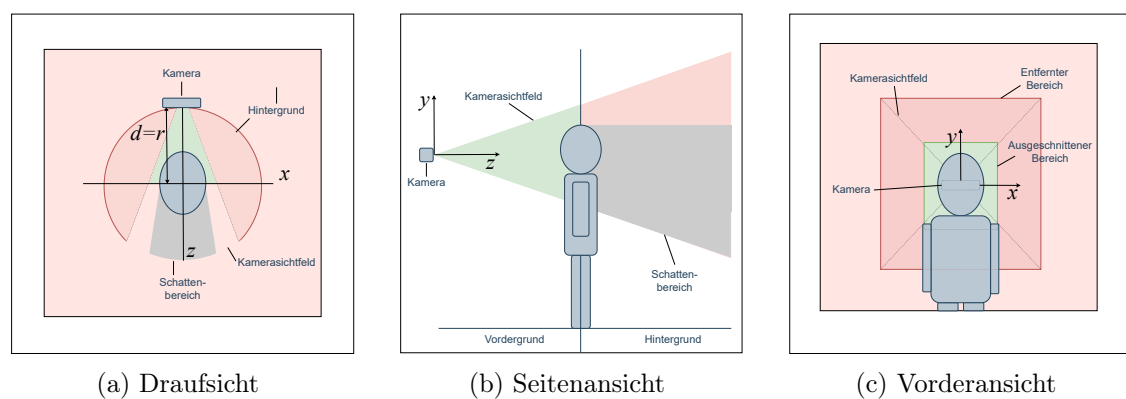


Abbildung 5.5: Kamerasichtfeld und Hintergrund

Ansatz 2: Algorithmus zur Clusteranalyse

Der Algorithmus zur Clusteranalyse unterteilt die aufgenommene Punktwolke in n Bereiche mit ähnlichen Eigenschaften. Abhängig vom verwendeten Algorithmus erfolgt die Zuordnung der Punkte dichtebasiert oder durch Partitionierung in eine festgelegte Anzahl an Gruppen, auch Cluster genannt. Es wird festgelegt, dass sich das Cluster, welches die Person abbildet, auf der z-Achse am nächsten an der Kamera befindet. Dieses Cluster wird extrahiert.

Die beschriebenen Ansätze werden anhand der in Tabelle 5.8 genannten Kriterien Umsetzungsaufwand, Rechenzeit, Robustheit und Wiederverwendbarkeit verglichen und bewertet. Aufgrund der komplexeren Algorithmik zur Analyse der Cluster und Ausgabe des gewünschten Clusters ist der Umsetzungsaufwand für den zweiten Ansatz höher. Außerdem ist bei diesem Ansatz die Rechenzeit höher, da die Komplexität der Algorithmen größer ist. Im Gegensatz zum ersten Ansatz ist der zweite Ansatz robuster gegenüber Änderungen im Aufnahmebereich. Der Abstand zwischen Kamera und Person kann beliebig variiert werden, sofern die Voraussetzung, dass die Person im Vordergrund steht, eingehalten wird. Ändert sich bei Ansatz 1 die Position der Kamera oder der Personen können zu viele oder zu wenig Punkte entfernt werden. Für die Anwendung bei einem Ganzkörperscanner müssten bei Ansatz 1 die Abstände im Programmcode geändert werden. Der Algorithmus zur Clusteranalyse kann ohne Änderungen wiederverwendet werden. Für die Anwendung wird ein Algorithmus zur Clusteranalyse verwendet, auch wenn bei diesem Ansatz mit einer höheren Rechenzeit zu rechnen ist, überwiegen die Vorteile in den Punkten Robustheit und Wiederverwendbarkeit.

Tabelle 5.7: Vergleich der Ansätze

	Abstandsdefinition	Clusteranalyse
Umsetzungsaufwand	+	o
Rechenzeit	o	+
Robustheit	-	+
Wiederverwendbarkeit	-	+

Auswahl des Algorithmus' zur Clusteranalyse

Zur Auswahl des Algorithmus' zur Clusteranalyse werden die Algorithmen k-Means und DBSCAN verglichen. k-Means partitioniert die Punktwolken in k -Cluster. Die Anzahl k der Cluster wird vor der Ausführung festgelegt und kann bei der Ausführung des Programms nicht geändert werden, vergleiche 2.2.1 ab Seite 5. DBSCAN ist ein dichtebasier-

ter Algorithmus. Für die Ausführung wird ein Radius vorgegeben, in dem sich mindestens n Punkte befinden müssen, damit der Punkt im Zentrum zu dem Cluster gehört. Die Anzahl der Cluster wird für jede Punktwolke individuell bestimmt, vergleiche 2.2.2 auf Seite 5.

Durch die dichtebasierte Vorgehensweise von DBSCAN können Punkte die sich außerhalb der festgelegten Umgebung von anderen Punkten befinden, zuverlässig als Ausreißer erkannt werden. k-Means hingegen ordnet jeden Punkt der gegebenen Punktwolke einem Cluster zu. Für eine sichere Durchführung des Algorithmus werden Punktwolken ohne Rauschen und Ausreißer benötigt. Die aufgenommenen Punktwolken unterscheiden sich besonders im Hintergrund durch den Wechsel der Perspektive. Die Anzahl der Cluster variiert. Da Änderungen in der Anzahl der zu erwartenden Cluster bei k-Means nur im Programmcode vorgenommen werden können, kann eine zuverlässige Funktion des Algorithmus' nicht vorausgesetzt werden. DBSCAN hingegen ist robust gegenüber einer variierenden Anzahl an Clustern und extrahiert zuverlässig die Punktwolke der Person im Vordergrund. DBSCAN wird für diese Anwendung ausgewählt.

Tabelle 5.8: Vergleich Clusteranalysealgorithmen

	k-Means	DBSCAN
Robustheit gegenüber Rauschen	+	o
Automatische Clustererkennung	-	+
Zuverlässigkeit	-	+

5.2.5.2 Registrierung

Die Registrierung bezeichnet das Zusammenführen von Punktwolken. Voraussetzung ist, dass die Punktwolken einen benachbarten Bereich der Oberfläche eines Objekts abbilden und einen Überschneidungsbereich besitzen, der jeweils den gleichen Oberflächenbereich abbildet. Die Größe des benötigten Bereichs ist vom verwendeten Registrierungsalgorithmus abhängig. Die Registrierung erfolgt anhand charakteristischer Merkmale in diesem Bereich. Nach [72] wird dieser Prozess üblicherweise in globale und lokale Registrierung unterteilt. Als globale Registrierung wird die Erstausrichtung der Punktwolken bezeichnet. Sie bringt die Punktwolken in eine direkte Nachbarschaft zueinander. Die Punktwolken werden als Quell- und Zielpunktwolken bezeichnet. Die Position der Quellpunktwolken wird an die der Zielpunktwolke durch Rotation und Translation angenähert,

sodass die überschneidenden Bereiche in Deckung liegen. Die lokale Registrierung optimiert diese Ausrichtung. Im Folgenden werden Algorithmen für die globale und die lokale Registrierung verglichen.

Globale Registrierung

Die initiale Ausrichtung der Punktwolken kann anhand der Aufnahmeperspektive oder auf Grundlage von übereinstimmenden Punkten erfolgen. Die Aufnahme der Punktwolke erfolgt ausschließlich in der mechanischen Konstruktion und ermöglicht damit eine Rekonstruktion der Aufnahmeposition auch bei fehlender Odometriesensorik. Für diesen Ansatz ist eine genaue Erfassung der Kameraposition erforderlich. Im zweiten Ansatz wird für jeden Punkt im Überschneidungsbereich der Quell- und Zielpunktwolke ein korrespondierender Punkt gefunden. Dafür werden Deskriptoren verwendet, die die Punkte anhand ihres Umfelds charakterisieren. Die Diskription kann auf Grundlage von Normalen und Eigenwerten der Punkte erfolgen. Damit die Registrierung robust gegenüber ungenauen Kamerapositionen ist, wird der zweite, algorithmische Ansatz, ausgewählt. Für die folgende globale Registrierung werden in [72] zwei Ansätze vorgestellt: Die Registrierung mit dem Branch and Bound Ansatz und der stichprobenbasierte Algorithmus RANSAC. Die Ansätze werden anhand der in Tabelle 5.9 dargestellten Kriterien verglichen.

Mithilfe des Branch and Bound Verfahrens werden kombinatorische Optimierungsprobleme gelöst. Hierfür wird der gesamte Lösungsraum nach der optimalen Lösung durchsucht und ein Entscheidungsbaum erstellt [8]. Für den Vergleich wird der Branch and Bound Ansatz nach [17] verwendet. Im ersten Schritt werden in den Quell- und Zielpunktwolken charakteristische Punkte anhand der Deskriptoren ausgewählt. Anhand dieser Punkte werden potenziell übereinstimmende Punktepaare in beiden Punktwolken gebildet. Mithilfe eines Baumdiagramms wird ermittelt, wie gut das Punktepaar zueinander passt. Dieser Vorgang wird mit unterschiedlichen Konstellationen der Punktepaare wiederholt bis das Abbruchkriterium erreicht ist.

Das Branch and Bound Verfahren kann anhand von wenigen Punkten eine optimale Lösung ermitteln. Da das Verfahren nach [72] aufgrund der vielen Berechnungen zeitaufwändig ist, ist RANSAC im Vorteil. [72] RANSAC schätzt in einem iterativen Prozess Modelle für potenzielle Punktepaare, um die Punktwolken in Deckung zu bringen. Das Verfahren wird in Kapitel 2.3.2 ab Seite 2.3.2 beschrieben. Wie das Branch and Bound Verfahren ist RANSAC robust gegenüber verrauschten Punktwolken. Grund dafür ist die Modellbildung anhand von Deskriptoren, die in beiden Ansätzen verwendet wird. Ein weiterer Vorteil ist die Implementierung des RANSAC Algorithmus in der

Tabelle 5.9: Vergleich globale Registrierung

	RANSAC	Branch and Bound
Berechnungsdauer	+	o
Robustheit gegenüber Rauschen	+	+
Umsetzungsaufwand	+	-

Open3D-Bibliothek. Wie in Abschnitt 5.2.1 beschrieben, ist das Intel RealSense SDK 2.0 in Open3D implementiert und bietet somit eine Schnittstelle für die Weiterverarbeitung der aufgenommenen Punktwolken. Der Umsetzungsaufwand wird damit geringer. Eine vergleichbare Implementierung für den Branch and Bound Algorithmus steht zum Zeitpunkt der Erstellung der Arbeit nicht zur Verfügung. Auf Grundlage der Bewertungen in Tabelle 5.9 wird für die globale Registrierung der RANSAC-Algorithmus verwendet.

Lokale Registrierung

Nach der initialen Ausrichtung werden die Punktwolken mit der lokalen Registrierung näher aneinander angepasst. In diesem Abschnitt wird der Iterative Closest Point-Algorithmus mit dem Normal Distribution Transform-Algorithmus, kurz NDT, verglichen. ICP richtet die Punktwolken anhand von geometrischen Eigenschaften der Punkte zueinander aus, vergleiche Abschnitt 2.3.3 ab Seite 10. NDT verwendet für die Ausrichtung statistische Modelle. Der Vergleich erfolgt auf Grundlage der in Tabelle 5.10 aufgeführten Kriterien. Der NDT-Algorithmus wird anhand von [4, 68] beschrieben. Im ersten Schritt wird die Zielpunktwolke in Bereiche gleicher Größe unterteilt. Für jeden Bereich wird eine Normalverteilung berechnet, die die lokale Wahrscheinlichkeit angibt einen Punkt an einem bestimmten Punkt zu finden. Anschließend werden eine Kovarianzmatrix und der Mittelwert berechnet. Die Eigenvektoren und die Eigenwerte der Kovarianzmatrix beschreiben die Ausrichtung und die Oberflächenstruktur der Punktwolke. Durch die initiale Ausrichtung befinden sich die Quell- und Zielpunktwolke in direkter Nähe. Die Punkte der Quellpunktwolke werden dann den Bereichen der Zielpunktwolke zugeordnet. Anschließend wird für jeden Punkt der Quellpunktwolke geprüft mit welcher Wahrscheinlichkeit er Teil des Bereichs der Zielpunktwolke ist. Aus den Wahrscheinlichkeiten aller Punkte wird ein Wert gebildet, der innerhalb eines Toleranzbereichs liegen muss. Ist das nicht der Fall wird die Quellpunktwolke neu ausgerichtet und der Vorgang wiederholt, bis das Abbruchkriterium erreicht ist.

Durch die statistische Vorgehensweise ist der NDT-Algorithmus weniger anfällig gegenüber Fehlern bei der Erstausrichtung [35]. Aufgrund des Nächste-Nachbarn-Prinzips müssen die Überschneidungsbereiche der Quell- und Zielpunktwolken für den ICP-Algorithmus

Tabelle 5.10: Vergleich lokale Registrierung

	ICP	NDT
Robustheit gegenüber Erstausrichtung	o	+
Berechnungsdauer	+	+
Umsetzungsaufwand	+	-

in direkter Nähe sein. Ist das nicht der Fall, benötigt der Algorithmus eine große Anzahl an Iterationsschritten, beziehungsweise kann die Registrierung nicht durchführen. Das Nächste-Nachbarn-Prinzip ist außerdem besonders bei großen Datensätzen, wie sie bei der Kartierung von Umgebungen erzeugt werden, rechenintensiv [68]. Der Berechnungsaufwand des NDT-Algorithmus für große Datensätze ist im Gegensatz dazu aufgrund der beschriebenen Vorgehensweise geringer. Große Datensätze werden beispielsweise von Lidar-Sensoren im Straßenverkehr aufgenommen, wobei Punktwolken mit einer Größe von 655.000 bis 2,6 Millionen Punkten, wie im Fall des Ouster Lidars, aufgenommen und verarbeitet werden müssen [49]. Punktwolken die von der Intel RealSense aufgenommen werden liegen unter 400.000 Punkten pro Punktwolke. Die Anzahl der Punkte wird vor der Registrierung reduziert und die nicht benötigten Daten im Hintergrund entfernt, so dass Punktwolken mit einer Größe von unter 50.000 Punkten entstehen. Im vorliegenden Fall wird mit vergleichsweise kleinen Punktwolken gearbeitet, womit die Auswirkungen auf die Verarbeitungszeit als vernachlässigbar angesehen.

Da der ICP-Algorithmus ebenfalls Teil der Open3D-Bibliothek ist, ist der Umsetzungsaufwand geringer. Aufgrund der weiten Verbreitung sind zahlreiche Anpassungsmöglichkeiten für den ICP-Algorithmus verfügbar. Eine Anpassungsmöglichkeit, die auch die Open3D-Bibliothek zur Verfügung stellt, ist das Annäherungsverfahren. Erfolgt die Annäherung der Punktwolken mit dem Punkt-zu-Ebene Verfahren, ist der Algorithmus robuster gegenüber Rauschen und Ausreißern, als bei der Verwendung des Punkt-zu-Punkt Verfahrens [56]. Nach Tabelle 5.10 überwiegen die Vorteile des ICP-Algorithmus für diese Anwendung.

5.2.5.3 Oberflächenrekonstruktion

In Anforderung SW 9 wird festgelegt, dass das fertige Modell als *.stl* oder *.obj* in einem neutralen CAD-Format gespeichert wird, damit ein herstellerunabhängiges Öffnen von 3D-Modellen in CAD-Programmen und Slicer-Software möglich ist. Open3D bietet nach [45] die Funktion zum Speichern der Punktwolken in den genannten Formaten.

Voraussetzung für die Konvertierung einer *.ply*-Datei in eine *.stl* oder *.obj*-Datei ist die Rekonstruktion einer Oberfläche. Bei Verwendung der Open3D-Bibliothek ist der Umsetzungsaufwand bei beiden Formaten gleich, da dieselben Funktionen verwendet werden. Für die Rekonstruktion der Oberfläche sind in der Open3D-Bibliothek drei Algorithmen implementiert: der Alpha Shapes Algorithmus, der Ball Pivoting Algorithmus und der Poisson Oberflächenrekonstruktionsalgorithmus. In der vorliegenden Arbeit soll gezeigt werden, dass die registrierten Punktwolken als 3D-Modell gespeichert werden können. Eine ausführliche Betrachtung der Oberflächenrekonstruktion wird hier nicht vorgenommen. Für diesen Anwendungsfall sind alle drei Algorithmen gleichermaßen geeignet. Der Umsetzungsaufwand aller drei Algorithmen wird als ähnlich groß eingeschätzt. Für die Umsetzung wird der Ball Pivoting Algorithmus verwendet der in den Grundlagen in Kapitel 2.4 ab Seite 12 beschrieben wird.

6 Design

In diesem Kapitel wird das Design vorgestellt. Es wird unterteilt in Hardware und Software und basiert auf dem Konzept in Kapitel 5.

6.1 Hardware

In diesem Abschnitt wird das Design der Hardwarekomponenten vorgestellt. Nach der Auswahl der Kamerahalterung wird der Schrittmotor vorgestellt. Zu jedem Design werden in einer Tabelle die dafür benötigten Komponenten mit ihren Preisen aufgelistet. Durch politische und wirtschaftliche Ereignisse, die sich im Verlauf der Bearbeitungsphase dieser Masterarbeit ereignet haben, ist die Preisentwicklung der Komponenten nicht stabil. In dieser Arbeit werden die Preise zum Zeitpunkt der Beschaffung angegeben. Bei Abgabe der Masterarbeit sind einige Komponenten, wie die Intel RealSense Kamera für den doppelten Preis erhältlich.

6.1.1 Kamerahalterung

Für die Aufnahme von Bilddaten wird eine Konstruktion zur Halterung und Bewegung des Sensors aus Aluprofilen aufgebaut. Die Konstruktion ist quaderförmig und besitzt eine quadratische Grundfläche mit einer Kantenlänge von 1 m, die Höhe des Quaders beträgt 2 m. Im Inneren befindet sich der Aufnahmebereich für die zu scannenden Personen. Am oberen Ende der Konstruktion befinden sich zwei Aluprofile an denen mithilfe einer Trägerplatte, Abbildung 6.2a, eine Motorhalterung mit Schrittmotor, Abbildung 6.2b und ein U-förmiger Rahmen befestigt sind, siehe Abbildung 6.3. Mithilfe des Rahmens wird die Kamera in einer kreisförmigen Bewegung um die zu scannende Person bewegt. Die verwendeten Bauteile werden in Tabelle 6.1 aufgelistet. Der Motor und der U-Rahmen werden durch eine Welle verbunden, die mit einem Wellenhalter Flansch am U-Rahmen befestigt wird und in einem Linearlager sowie einem Flanschlager gelagert wird. Zum

Tabelle 6.1: Aluprofilkonstruktion - Angaben nach [14]

Menge	Bezeichnung	Maße	Kosten
15	Aluprofil 20x20L B-Typ Nut 6	1000 x 20 x 20 mm	73,50 €
2	Aluprofil 20x20L B-Typ Nut 6	600 x 20 x 20 mm	5,88 €
8	Innenwinkel 20 B-Typ Nut 6		10,96 €
50	Nutenstein	Gewinde: M5	47,00 €
50	Flachrundschraube	M5 x 8 mm	13,00 €
2	Würfelverbinder B-Typ Nut	20 x 20 x 20 mm	11,02 €
		Gesamtkosten	161,36 €

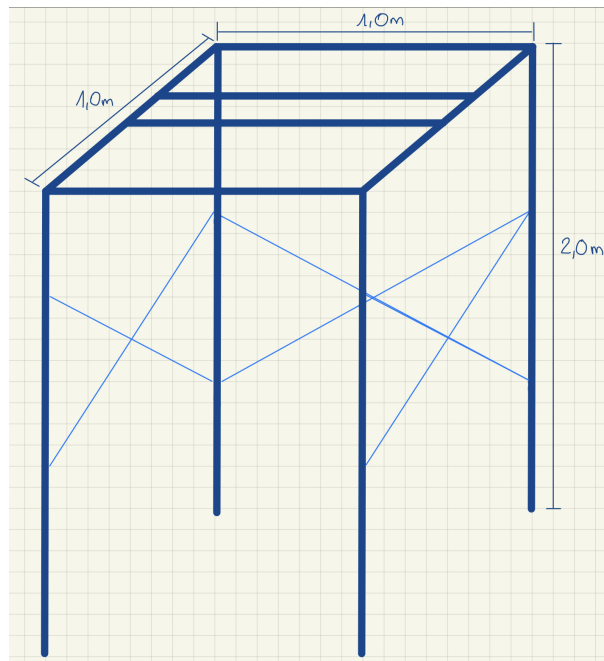


Abbildung 6.1: Konstruktion

Ausgleich der verschiedenen Wellendurchmesser von Motorwelle und Verbindungswelle wird eine Wellenkupplung verwendet. Als Absturzsicherung des U-Rahmens wird ein Sicherungsstift in der Welle über dem Flanschlager verbaut. Die Befestigung des U-Rahmens und die Motorwelle beanspruchen etwa 0,1 m womit der Aufnahmebereich für Personen bis zu einer Größe von etwa 1,9 m geeignet ist. Zur Stabilisierung besitzt die Konstruktion an drei von vier Außenseiten Kreuze.

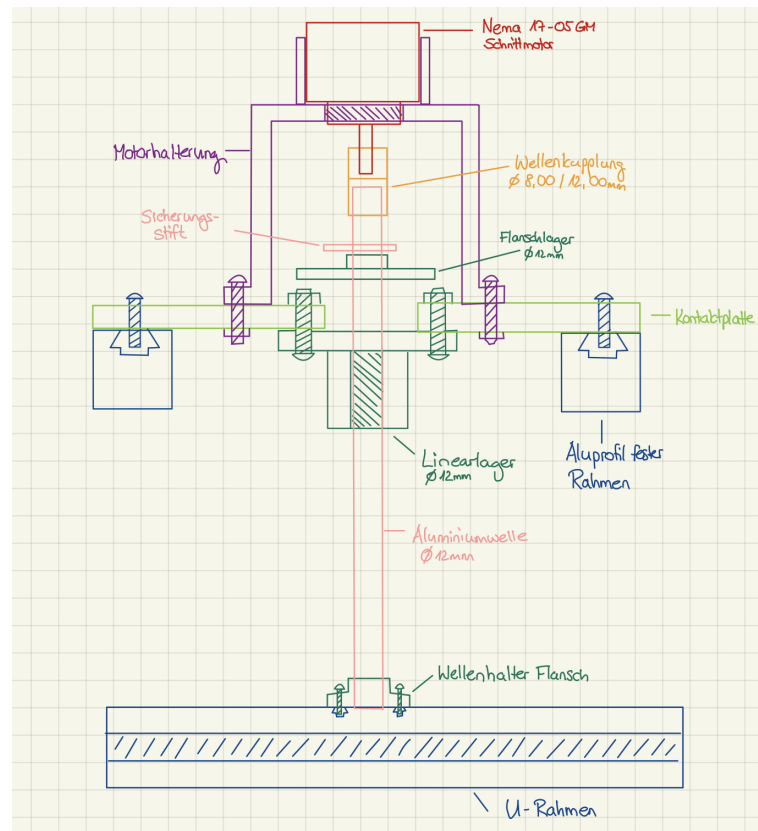
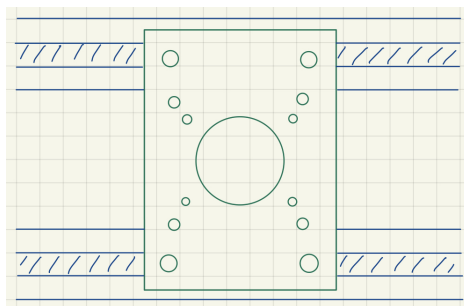
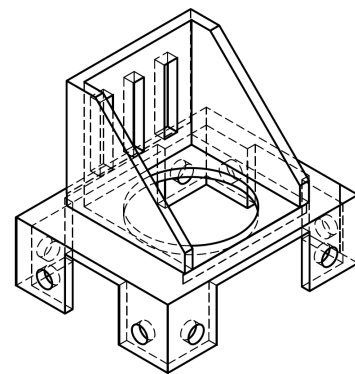


Abbildung 6.3: Motorhalterung und Befestigung des U-Rahmens



(a) Trägerplatte



(b) Motorhalterung

Abbildung 6.2: Detailansicht der Aufhängung

Tabelle 6.2: Kosten Konzept 1: Höhenverstellbare Kamera - Angaben nach [9, 14]

	Menge	Preis
Intel RealSense D415	1	150,00 €
Schrittmotor NEMA 1705	1	49,99 €
Gleiter	1	2,45 €
Zahnräder	2	3,20 €
Zahnriemen (1 m)	1	2,50 €
Kamerahalter	1	2 €
Gesamtkosten		210,14 €

Tabelle 6.3: Kosten Konzept 2: Fest montierte Kameras - Angaben nach [9, 14]

	Menge	Stückpreis
Intel RealSense D415	2	150,00 €
Gleiter	2	2,45 €
Kamerahalter	1	2 €
Gesamtkosten		307,59 €

6.1.2 Montage der Sensoren

Die Montage der Sensoren erfolgt am U-Rahmen der Konstruktion. Für die vollständige Erfassung des Kopfes, auch unter dem Kinn und der Nase, wie es in Abbildung 6.4 mit roten Kreisen gekennzeichnet ist, sind zwei Kameraperspektiven notwendig. Dafür werden zwei Konzepte verglichen: einerseits die höhenverstellbare Montage der Kamera und andererseits die Verwendung von zwei Kameras.

Konzept 1: Höhenverstellbare Kamera

Das erste Konzept ist in Abbildung 6.4a dargestellt und wird mit einer Kamera umgesetzt, die mit einem Schrittmotor in der Höhe verstellt wird. Dafür wird die Kamera auf einem Gleiter mit einem 3D gedruckten Kamerahalter montiert, der über einen Zahnriemen und Zahnräder bewegt wird. In Tabelle 6.2 werden die benötigten Bauteile inklusive der zu erwartenden Kosten aufgelistet.

Konzept 2: Fest montierte Kameras

Es werden zwei Kameras mit überschneidendem Sichtfeld am Aluprofil montiert, wie in Abbildung 6.4b dargestellt. Die Position der Kameras wird während der Aufnahme nicht verändert. Die Kameras werden an einem 3D gedruckten Kamerahalter montiert, der auf einem Gleiter am Aluprofil befestigt wird. Die Kosten sind in Tabelle 6.3 aufgelistet.

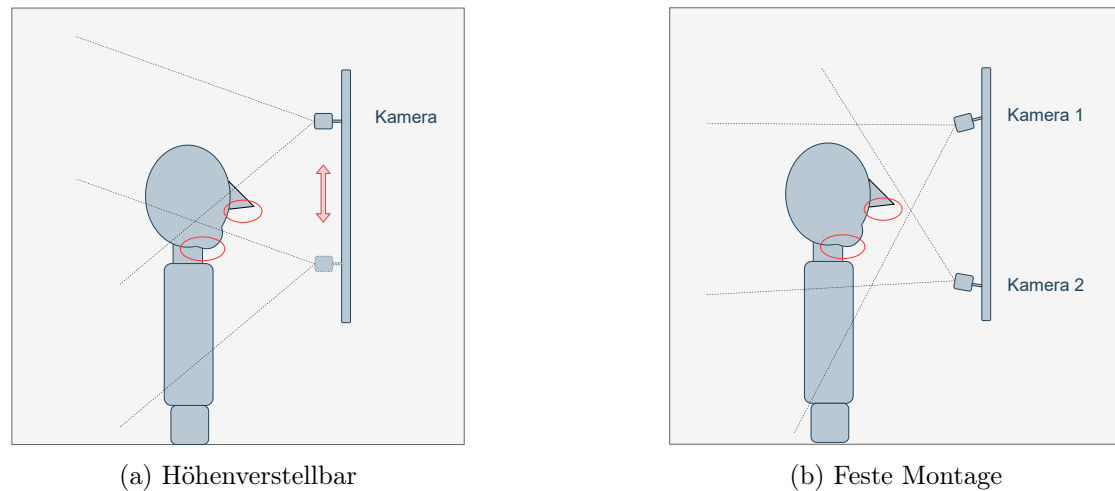


Abbildung 6.4: Montage der Kameras

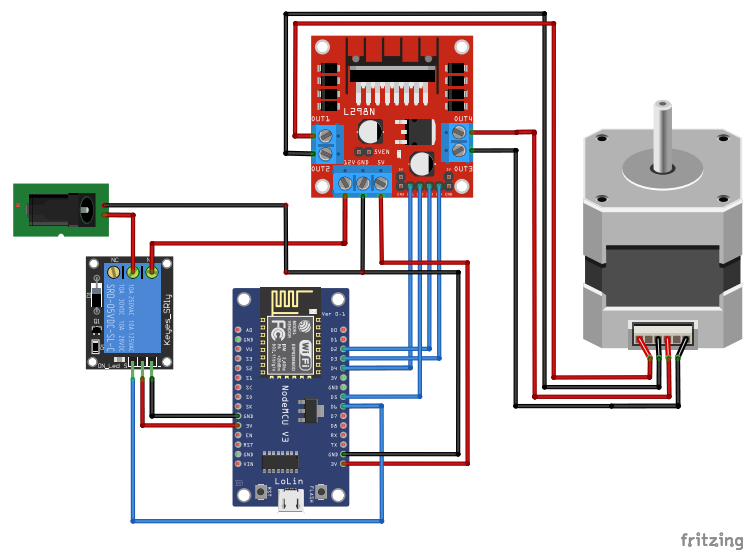
Tabelle 6.4: Vergleich der Konzepte

	Konzept 1	Konzept 2
Mechanische Umsetzungsaufwand	o	+
Softwaretechnische Umsetzungsaufwand	o	o
Kosten	+	-

Beide Konzepte werden nach den Kriterien aus Tabelle 6.4 bewertet. Die mechanische Umsetzung und der damit verbundene Aufwand ist bei Konzept 1 mit der höhenverstellbaren Kamera höher. Mögliche Probleme sind verrauschte Aufnahmen, weil die Kamera in den Endlagen nicht arretiert werden kann und durch die Drehung des U-Rahmens nachschwingt. Diese Probleme bestehen bei Konzept 2 nicht, da die Kameras am Aluprofil nicht beweglich befestigt werden können. Vor der Umsetzung von Konzept 2 ist zu klären, ob beide Kameras zur gleichen Zeit aufnehmen können, da sich die Aufnahmebereiche und damit die durch den Projektor auf das Objekt projizierten Muster überschneiden. Anhand der projizierten Muster werden die Tiefendaten erfasst. Alternativ ist eine serielle Aufnahme mit zwischenzeitlicher Abschaltung der Projektoren möglich. Das erhöht den softwaretechnischen Aufwand. In den Tabellen 6.2 und 6.3 werden die Kosten für die Umsetzung der Konzepte dargestellt. Wie bereits erwähnt, beträgt das Budget nach Anforderung HW-A 4 für das gesamte Projekt 500 €. Die Umsetzungskosten von Konzept 2 sind mit 309,09 € höher, als die Kosten für das erste Konzept die 211,64 € betragen. Da das vorhandene Budget nicht überschritten werden darf, wird das erste Konzept angewendet.

Tabelle 6.5: Motorbefestigung - Angaben nach [14]

Menge	Bezeichnung	Maße	Kosten
1	Aluminiumwelle	Durchmesser: 12 mm Länge: 200 mm	5,46 €
1	Wellenhalter Flansch	Durchmesser: 12 mm Tiefe: 6 mm	4,00 €
1	Linearlager	Durchmesser: 12 mm Tiefe: 6 mm	7,18 €
1	Wellenkupplung flexibel	Durchmesser 1: 8 mm Durchmesser 2: 12 mm	8,00 €
		Gesamtkosten	24,64 €



fritzing

Abbildung 6.5: Schrittmotorsteuerung

6.1.3 Schrittmotor

Zur Bewegung des U-Rahmens wird der Schrittmotor NEMA 17-05GM verwendet, der mit einer Spannung von 12 V betrieben wird. Für die Schrittmotorsteuerung wird der Doppel-Vollbrückenmotortreiber L298N verwendet, der über einen ESP8266 angesteuert wird. Der ESP ist seriell über USB mit dem Laptop verbunden. Über ein Relais wird die Spannung ein- und ausgeschaltet. In Abbildung 6.5 ist die Schaltung abgebildet. Die Abschaltbarkeit des Motors wird über einen Schalter an der Spannungsquelle realisiert. Die verwendeten Komponenten sind inklusive der Preise in Tabelle 6.6 dargestellt.

Tabelle 6.6: Schrittmotorsteuerung - Angaben nach [9]

Menge	Datenblatt	Bezeichnung	Kosten
1	[30]	Schrittmotor NEMA 17-05GM	49,99 €
1	[70]	Schrittmotorsteuerung L298	7,49 €
		Gesamtkosten	57,48 €

6.2 Software

In diesem Abschnitt wird das Softwaredesign vorgestellt. Nach der Festlegung des Gesamtablaufs werden die Schnittstellen zwischen den Softwaremodulen beschrieben. Im Anschluss folgt die Vorstellung der Programmteile.

6.2.1 Gesamtablauf

In diesem Abschnitt wird die Detaillösung für den in Abschnitt 5.2.2 auf Seite 36 vorgestellten Ablauf des gesamten Programms beschrieben. Das Programm unterteilt sich in drei Module: die Bedienoberfläche als Schnittstelle zu den Benutzer:innen, die Bildaufnahme als Schnittstelle zur Kamera und zur Schrittmotorsteuerung sowie die Bildverarbeitung, die die Punktwolken zu einem 3D-Modell verarbeitet. Die Module und der Ablauf sind in Abbildung 6.6 dargestellt. Die Benutzer:innen starten über die Bedienoberfläche das Programm, das dann die Bildaufnahme startet. Nach der erfolgreichen Aufnahme der Punktwolken werden diese gespeichert und im Modul Bildverarbeitung zu einem 3D-Modell verarbeitet, das im Anschluss über die Bedienoberfläche angezeigt wird. In den nachfolgenden Abschnitten werden die drei Module ausführlich beschrieben.

6.2.2 Schnittstellen

In diesem Abschnitt werden die Schnittstellen zwischen den einzelnen Modulen des gesamten Programms erläutert. Das Programm wird objektorientiert programmiert. Die in Abschnitt 6.2.1 vorgestellten Module Bedienoberfläche, Bildaufnahme und Bildverarbeitung werden in Klassen umgesetzt, was eine Erweiterbarkeit des Programms und Wiederverwendbarkeit einzelner Klassen für ähnliche Anwendungen ermöglicht und die Anforderung SW 2 dadurch erfüllt. In Abbildung 6.7 ist das Klassendiagramm des gesamten Programms dargestellt, in dessen Zentrum die Klasse *Logger*. Diese dokumentiert die Logdaten des Programms und speichert sie im Ordner mit den aufgenommenen

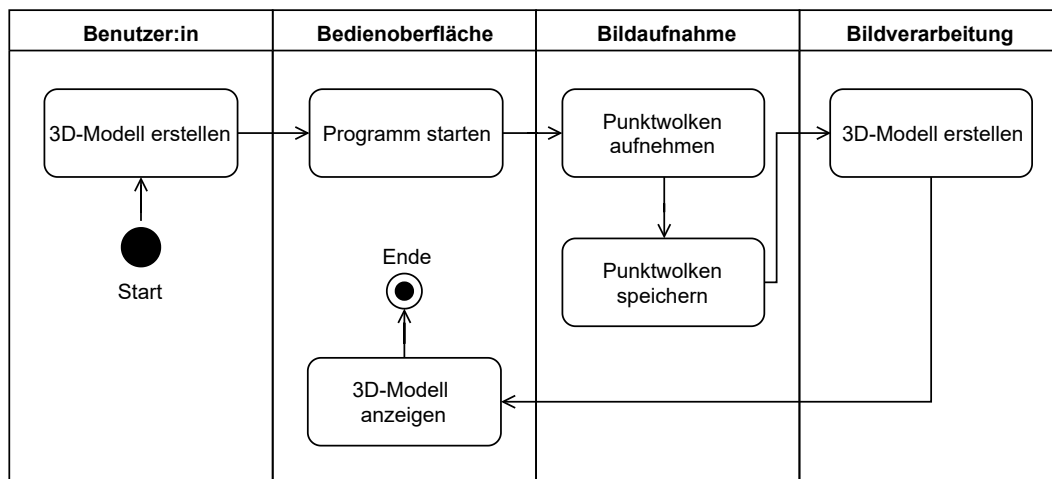


Abbildung 6.6: Übersicht des Gesamtablaufs

Punktvolken. Die Bedienoberfläche wird in der Klasse *MainWindow* und *Ui_MainWindow* umgesetzt. Die Klassen für die Bildaufnahme sind auf der linken Seite dargestellt, die für die Bildverarbeitung auf der rechten Seite. Zur Bildaufnahme gehören die Klassen *ImageCapture* und *CommunicationToESP*. *ImageCapture* dient als Schnittstelle zur Kamera und speichert die aufgenommenen Punktvolken. Über die Klasse *CommunicationToESP* wird eine serielle Kommunikation mit der Schrittmotorsteuerung zum Bewegen der Kamera hergestellt. Die Bildverarbeitung wird von den Klassen *ManagePointClouds*, *EditPointCloud*, *Registration* und *SurfaceReconstruction* durchgeführt. Die Klasse *ManagePointClouds* koordiniert den Bildverarbeitungsprozess. Die Klasse *EditPointCloud* bearbeitet die einzelne Punktvolke und entfernt den Hintergrund. Das Zusammenfügen der Punktvolke wird in der Klasse *Registration* durchgeführt. Für die Oberflächenrekonstruktion werden die Methoden der Klasse *SurfaceReconstruction* verwendet.

6.2.3 Bedienoberfläche

Die Bedienoberfläche zeigt nach der Initialisierung den Startbildschirm an. Über das Menü kann der Scan gestartet oder ein aufgenommenes Modell angezeigt werden. Nach Abschluss der einzelnen Anwendungen wird immer der Startbildschirm angezeigt. Wählt eine Benutzer:in *Aufnahme starten* aus, wird eine Live-Vorschau des Aufnahmebereichs angezeigt. Nachdem sich die Benutzer:innen ausgerichtet haben, kann die Anwendung von einer zweiten Person über eine Schaltfläche gestartet werden. Wird der Scanner von einer Person alleine verwendet, erscheint nach dem Start ein Countdown nach dessen Ablauf

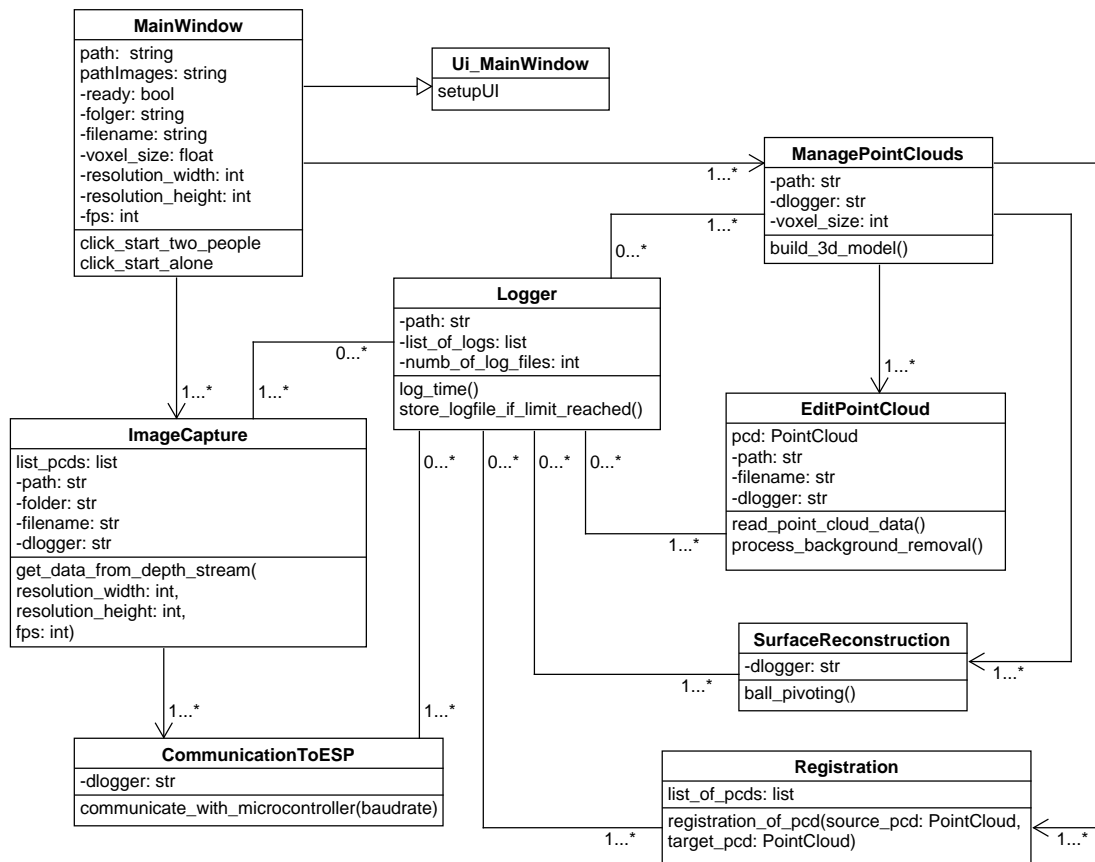


Abbildung 6.7: Übersicht der Klassen

der Scan gestartet wird. Über den Bildschirm werden Informationen angezeigt, dass der Scan durchgeführt wird und dass er abgeschlossen worden ist. Über den Menüpunkt *3D-Modell* auf dem Startbildschirm kann das aufgenommene Modell nach der Erstellung im Bildschirm *Bildvorschau* angezeigt werden.

6.2.4 Aufnahme der Punktwolken

In diesem Abschnitt wird die Aufnahme der Punktwolken beschrieben. Im ersten Teil werden zwei Ansätze zum Aufbau der Kommunikation zwischen dem Windows PC und der Hardware beschrieben. Die Hardware setzt sich aus der Kamera und der Schrittmotorsteuerung zusammen. Im zweiten Teil wird der Aufnahmeprozess beschrieben.

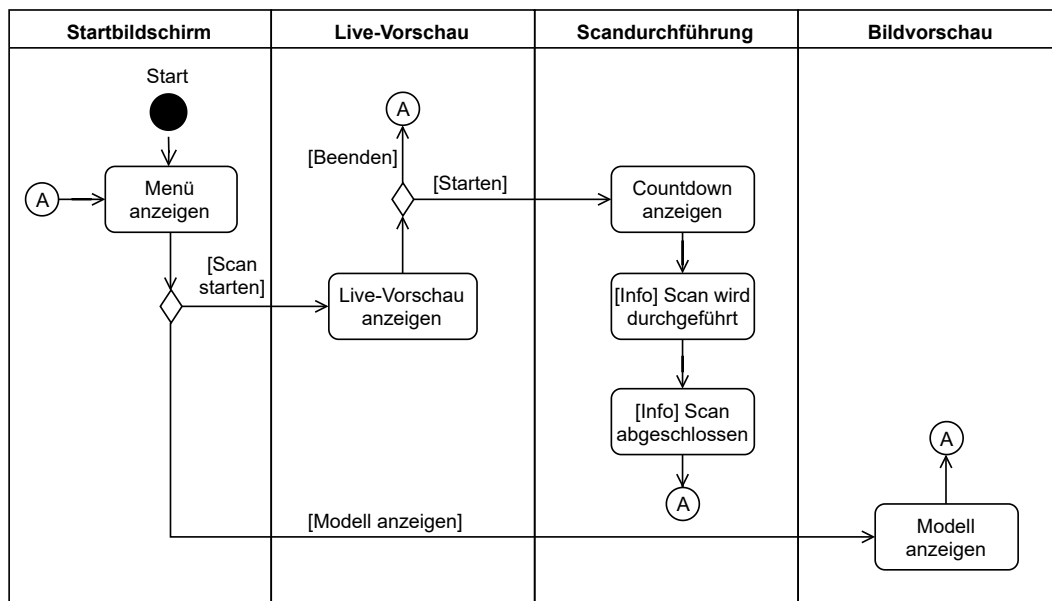


Abbildung 6.8: Bedienoberfläche

6.2.4.1 Kommunikation

In diesem Abschnitt wird beschrieben auf welcher Hardware das Modul Bildaufnahme ausgeführt werden soll. Die Module Bedienoberfläche und Bildverarbeitung werden auf einem Windows PC ausgeführt. Die Bildaufnahme wird in eine Kommunikationsschnittstelle auf dem Windows PC, die Schrittmotorsteuerung und die Kamera, aufgeteilt. Über die Kommunikationsschnittstelle werden Daten mit der Kamera und der Schrittmotorsteuerung ausgetauscht. Im Folgenden wird eine kabelgebundene und eine drahtlose Konfigurationen zur Kommunikation beschrieben.

Drahtlose Konfiguration

In dieser Konfiguration wird der Prozess der Bildaufnahme und der Motorsteuerung von einem Einplatinen-Computer übernommen. Die Kamera ist mit dem Einplatinen-Computer verbunden, der die aufgenommenen Daten zwischenspeichert und über eine drahtlose bidirektionale Verbindung an den Windows PC überträgt. In Abbildung 6.9 ist die Anordnung beispielhaft dargestellt.

Kabelgebundene Konfiguration

Bei der kabelgebundenen Konfiguration ist die Kamera mit dem PC verbunden. Die Daten werden direkt auf dem PC gespeichert. Für die Bildaufnahme wird der Schrittmotor

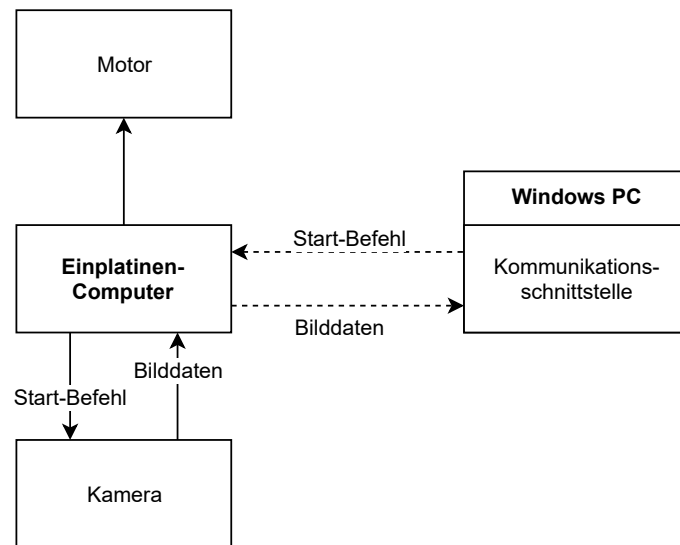


Abbildung 6.9: Drahtlose Konfiguration

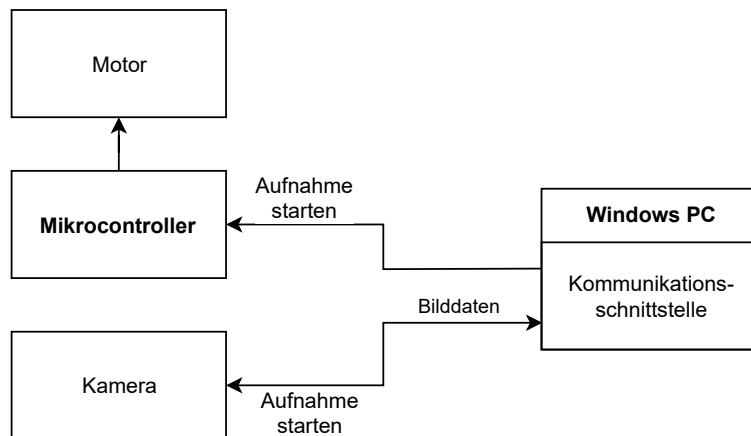


Abbildung 6.10: Kabelgebundene Konfiguration

über einen Mikrocontroller gesteuert. Dieser ist über einer seriellen Schnittstelle ebenfalls mit dem PC verbunden. Die Anordnung ist in Abbildung 6.10 dargestellt.

Bei der kabelgebundenen Konfiguration werden die Kabel und die Anschlüsse an den Hardwarekomponenten bei der Drehung des U-Rahmens mechanisch beansprucht. Bei regelmäßiger Nutzung der Anordnung sind sie einem hohen Verschleiß ausgesetzt. Die drahtlose Konfiguration hat den Vorteil, dass kein Kabel für die serielle Kommunikation zwischen PC und Mikrocontroller sowie PC und Kamera verlegt wird. Nachteilig ist, dass die Daten nach dem Scanvorgang zunächst übertragen werden müssen und die

Bildverarbeitung verzögert startet. Die Einrichtung einer drahtlosen Verbindung wird als aufwändiger angesehen als die Einrichtung einer seriellen Verbindung. Zudem kommen für die drahtlose Konfiguration Anschaffungskosten von etwa 80 € für einen Einplatinen-Computer, wie einen Raspberry Pi 4B 8 GB, hinzu [9]. Für die kabelgebundene Konfiguration mit Mikrocontroller sind die Anschaffungskosten zum Beispiel für einen ESP8266 niedriger und liegen bei etwa 13 € [55]. Aufgrund des geringeren Aufwands und der niedrigeren Kosten sowie der schnelleren Datenübertragung wird die kabelgebundene Konfiguration verwendet. Die Umsetzung erfolgt in der Klasse *CommunicationToESP*

6.2.4.2 Aufnahme

In der Klasse *ImageCapture* wird die Schnittstelle zur Kamera und der Aufnahmeprozess umgesetzt, der in in Abbildung 6.11 dargestellt ist. Nach der Initialisierung des Kamerastreams werden die Kameraverbindung und der USB-Typ geprüft. In folgenden Fällen wird das Programm beendet:

- Wenn kein Gerät gefunden wird.
- Wenn die angeschlossene Intel Kamera nicht der Produktlinie D400 entspricht. Vor dem Beenden wird die Benutzer:in dazu aufgefordert das Gerät zu tauschen.
- Wenn keine USB 3.2-Verbindung verwendet wird. Vor dem Beenden wird die Benutzer:in dazu aufgefordert die Verbindung zu ändern.

Sofern die Produktlinie der D400-Reihe entspricht und die Kamera mit einer USB 3.2-Verbindung angeschlossen ist, wird der Kamerastream gestartet. Für die Erfassung von Punktwolken wartet der Stream auf Frames, die im Anschluss gespeichert werden. In der Zwischenzeit rotiert die Kamera zur nächsten Position an der der Vorgang wiederholt wird. Der Ablauf wiederholt sich bis die festgelegte Anzahl an Aufnahmen erreicht ist. Im Anschluss wird der Stream beendet.

6.2.5 Bildverarbeitung

In diesem Abschnitt wird das Modul Bildverarbeitung beschrieben, das in die Untermodule Punktwolkenvorverarbeitung, Registrierung und Oberflächenrekonstruktion unterteilt ist. Sie basieren auf den in den Abschnitten 5.2.5.1 bis 5.2.5.3 vorgestellten Konzepten.

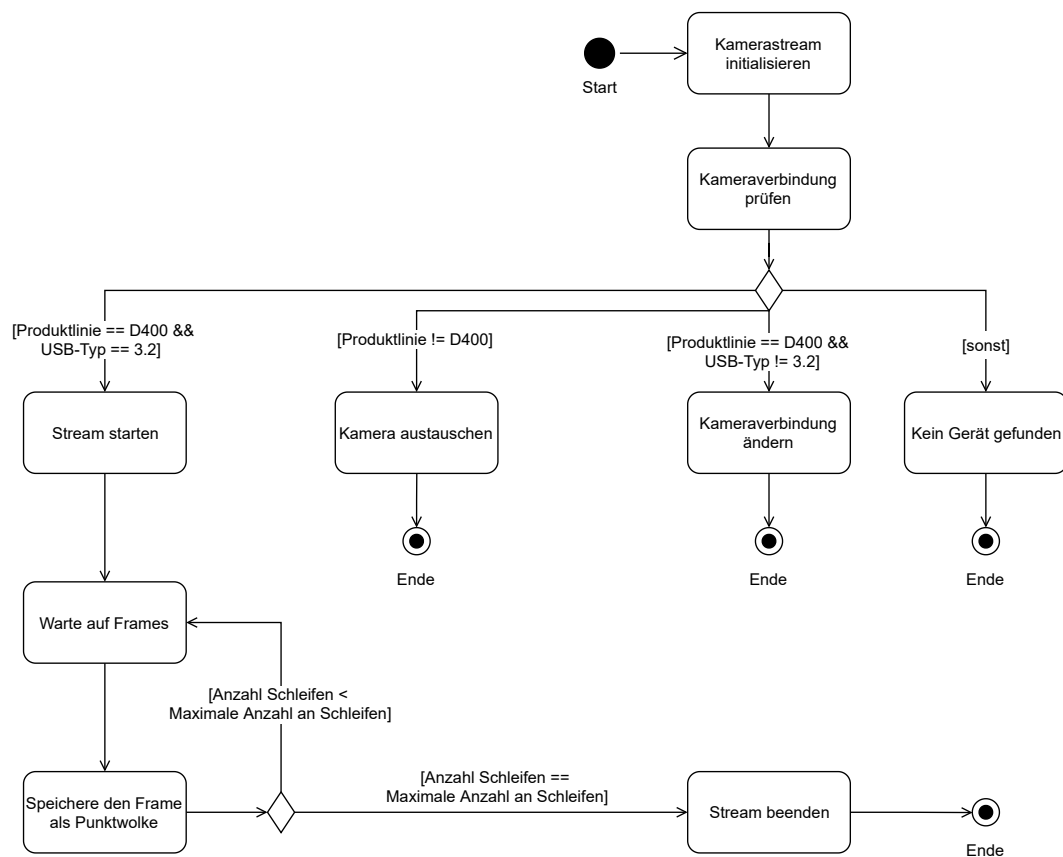


Abbildung 6.11: Ablauf der Aufnahme der Punktwolken

Die Bildverarbeitung ist in Abbildung 6.12 dargestellt und beginnt mit der Punktwolkenvorverarbeitung, die den Hintergrund in den Punktwolken entfernt, sodass nur noch die Daten der Person enthalten sind. Dieser Prozess wird solange ausgeführt bis alle aufgenommenen Punktwolken bearbeitet worden sind. Nach Abschluss der Vorverarbeitung wird die Registrierung mit RANSAC und ICP ausgeführt. Ist die Anzahl der aufgenommenen Punktwolken gleich der Anzahl an durchgeführten Registrierungen, wird für das entstandene 3D-Modell die Oberfläche rekonstruiert. Im letzten Schritt wird das Modell gespeichert. In den folgenden Abschnitten wird das Design der Untermodule beschrieben.

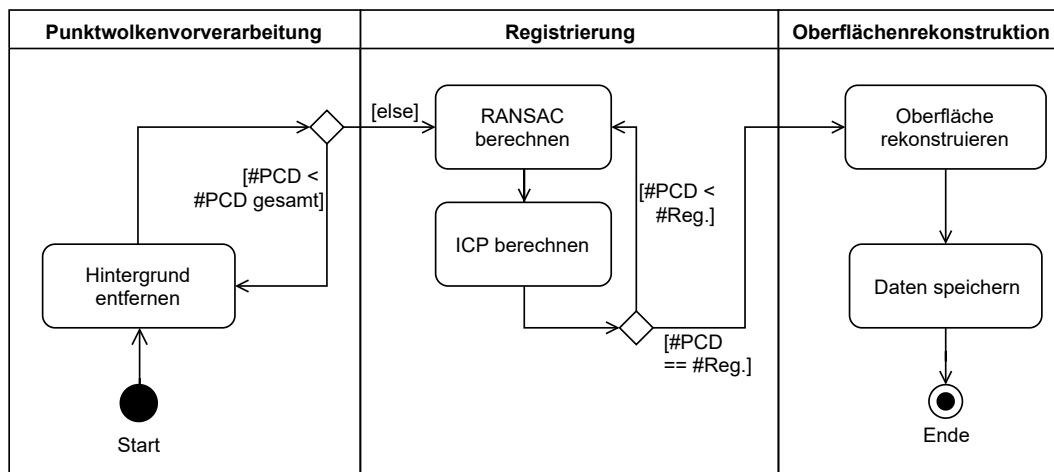


Abbildung 6.12: Übersicht der Module für die Bildverarbeitung

6.2.5.1 Punktwolkenvorverarbeitung

Die Punktwolkenvorverarbeitung wird in der Klasse *EditPointCloud* implementiert. Der Prozess wird in einer Schleife für alle Punktwolken durchgeführt, die sich im übergebenen Dateiverzeichnis befinden. Der Ablauf wird in Abbildung 6.13 dargestellt. Im ersten Schritt wird die Anzahl der Punkte in der Punktwolke reduziert, um die Berechnung zu beschleunigen. Im Anschluss werden der Hintergrund und die Ausreißer entfernt. Als Vorbereitung für den Registrierungsprozess werden in jeder Punktwolke Normalen und FPFH-Deskriptoren berechnet. Die Schritte Hintergrund und Ausreißer entfernen werden im Folgenden genauer beschrieben. Für beide Prozesse wird der unter 5.2.5.1 auf Seite 38 ausgewählte Clusteranalysealgorithmus DBSCAN verwendet, für den die Implementierung aus der scikit-learn Bibliothek verwendet wird.

Hintergrund entfernen

Der DBSCAN-Algorithmus unterteilt die gesamte Punktwolke in Cluster. Um die großen Cluster von Wänden und umgebenden Flächen vollständig zu erfassen, wird eine minimale Clustergröße festgelegt. Der Abstand der Cluster in z-Richtung wird im Bereich zwischen 0 bis -1 angegeben. Je größer der Wert, desto näher befinden sich die Cluster in der Nähe der Kamera. Es wird davon ausgegangen, dass das Cluster mit der Person am nächsten an der Kamera ist. Um es zu identifizieren wird der Mittelwert der z-Parameter jedes Clusters berechnet und das Cluster mit dem größten Mittelwert zurückgegeben.

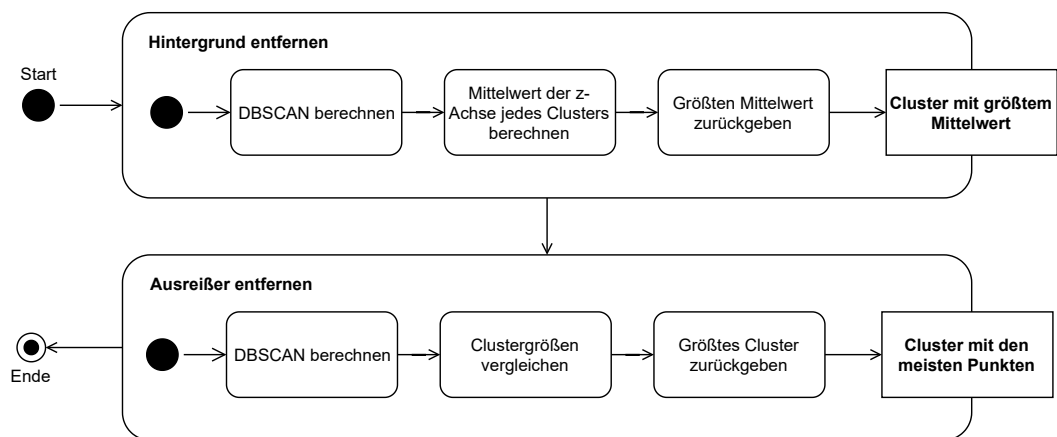


Abbildung 6.13: Ablauf des DBSCAN-Algorithmus

Ausreißer entfernen

In der Umgebung der großen Cluster befinden sich kleine Cluster mit wenigen Punkten, die nicht zu der Punktwolke der Person gehören. Diese werden durch eine erneute Anwendung des DBSCAN-Algorithmus erkannt. Da diese Cluster sehr viel kleiner sind als die Cluster im ersten Schritt, wird die minimale Clustergröße reduziert. Die Clustergrößen der erkannten Cluster wird verglichen und das Cluster mit der größten Anzahl an Punkten zurückgegeben.

6.2.5.2 Registrierung

Für die Registrierung werden zunächst eine Quell- und eine Zielpunktwolke bestimmt. Beide Punktwolken werden anschließend mit dem RANSAC-Algorithmus in Deckung gebracht. Die Durchführung des Algorithmus wird wiederholt bis der Fitnesswert gleich oder größer als der Schwellwert ist, siehe Abbildung 6.14. Der ICP-Algorithmus wird anschließend ebenfalls in einer Schleife ausgeführt. Sobald auch hier der Fitnesswert größer oder gleich dem Schwellwert ist, wird die Ausführung beendet. Ergebnis ist eine registrierte Punktwolke aus Quell- und Zielpunktwolke. Ist die Anzahl der durchgeführten Schleifen zur Registrierung der Punktwolken kleiner als die Anzahl der aufgenommenen Punktwolken, wird die berechnete Punktwolke als neue Quellpunktwolke festgelegt. Für die erneute Registrierung werden Normalen und FPFH-Deskriptoren nach [46] berechnet. Der Registrierungsprozess endet, wenn die Anzahl der durchgeführten Registrierungen gleich der Anzahl aufgenommenen Punktwolken ist.

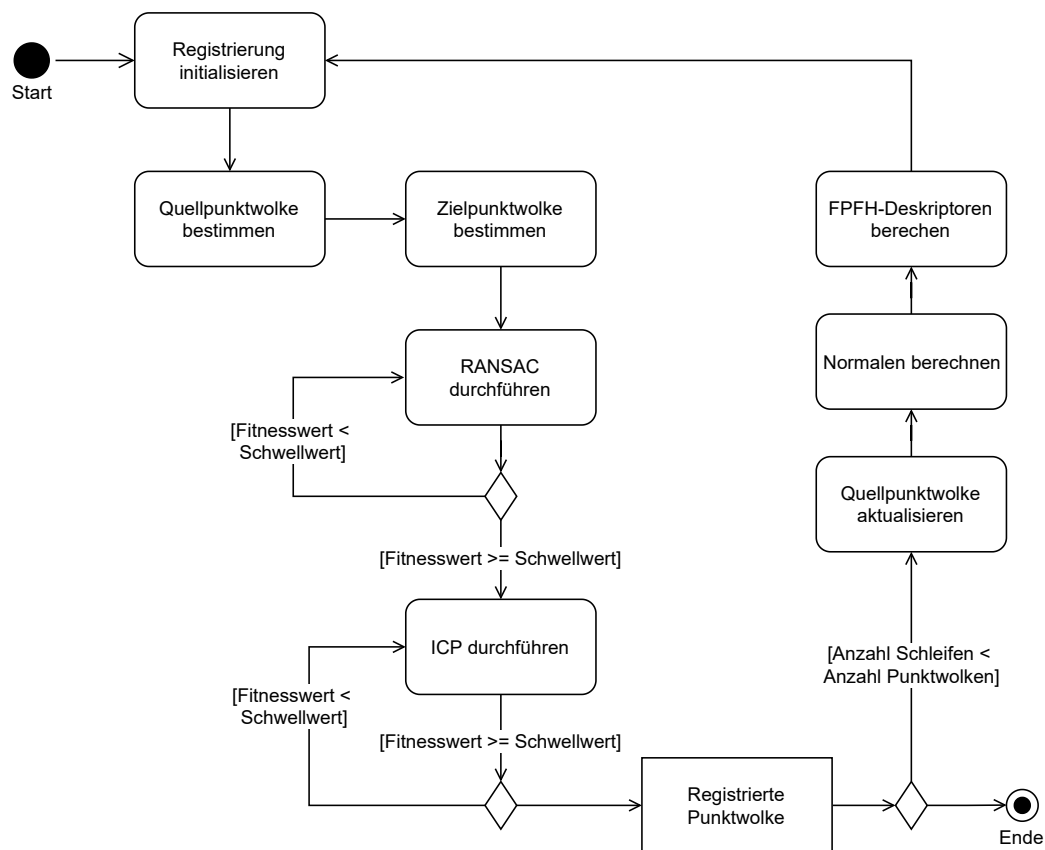


Abbildung 6.14: Ablauf der Registrierung

6.2.5.3 Oberflächenrekonstruktion

Die Oberflächenrekonstruktion wird mit [51] umgesetzt. Mit Ausnahme der Ein- und Ausgabe der Punkt wolke ist der Code an dieser Stelle übernommen worden, um zu zeigen, dass eine Oberflächenrekonstruktion mit der vorhandenen Punkt wolke möglich ist.

Die Rekonstruktion der Oberfläche wird nach dem in Abbildung 6.15 dargestellten Ablauf unter Verwendung des Ball Pivoting Algorithmus durchgeführt. Für die Berechnung des Radius des Balls wird zunächst der Abstand der Punkte der Punkt wolke bestimmt. Nach der Festlegung des Radius wird der Ball Pivoting Algorithmus durchgeführt und die entstandene Punkt wolke in einem CAD-Format gespeichert.

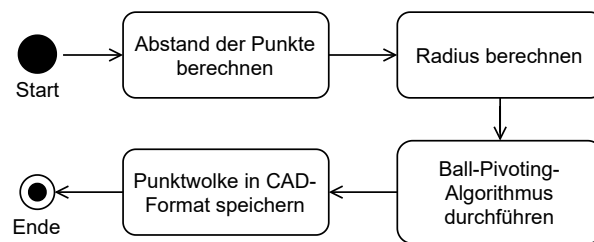


Abbildung 6.15: Ablauf der Oberflächenrekonstruktion

7 Umsetzung

Dieses Kapitel beschreibt die Umsetzung des Konzepts und unterteilt sich in die Abschnitte Hardware und Software.

7.1 Hardware

In diesem Abschnitt werden aufgetretene Schwierigkeiten bei der Verwendung der im Design entwickelten Kamerahalterung und die Entwicklung einer verbesserten Halterung beschrieben. Zudem werden die Motorhalterung und die Montage der Sensoren dargestellt.

7.1.1 Kamerahalterung

Die im Design unter Abschnitt 6.1.1 ab Seite 46 entwickelte Kamerahalterung hat sich nach dem Aufbau aus Aluprofilen, der in Abbildung 7.1 dargestellt ist, als instabil erwiesen. Bei der Bewegung des U-Rahmens durch den Motor schwingt sich die gesamte Konstruktion auf und die Bildaufnahmen verwackeln. In der Form ist der Aufbau für den Zweck der Bildaufnahme ungeeignet. Durch den Einbau weiterer Aluprofile zur Verbindung der Beine sowie durch zusätzliche Kreuze kann die Konstruktion stabilisiert werden. Zur Überprüfung dieser Annahme ist Konstruktion 2 aus den vorhandenen Bauteilen entworfen und aufgebaut worden. Alle entworfenen Konstruktionen werden anhand der Kriterien in Tabelle 7.1 verglichen.

Konstruktion 2

Die zweite Konstruktion ist würfelförmig und hat eine Kantenlänge von 1 m. Sie ist in Abbildung 7.2 dargestellt. An drei von vier Seiten der Konstruktion befinden sich Kreuze zum aussteifen. Zudem sind die Beine durch Aluprofile miteinander verbunden.



Abbildung 7.1: Konstruktion

Im Gegensatz zur ersten Konstruktion müssen die Benutzer:innen in dieser Konstruktion im Aufnahmebereich auf dem Boden sitzen. Die Anforderung HW-A 1 definiert die Gesamtkörpergröße von Personen, die den 3D-Scanner benutzen können sollen. Durch Abweichungen in Bein- und Oberkörperlängen von Personen, kann die Anforderung nicht mehr als vollständig erfüllt angesehen werden. Sitzen die Personen und haben einen Oberkörper der länger als 0,9 m ist, aber eine Gesamtkörpergröße die kleiner als 1,9 m ist, ist die Anordnung nicht passend. Durch die sitzende Position bei der Bildaufnahme und den eingeschränkten Raum im Würfel wird das Verlassen des Aufnahmebereichs erschwert. Zusätzlich befindet sich im Zugangsbereich auf dem Boden ein Aluprofil. Die Gefahr des Stolperns, Stoßens oder Quetschens beim Einstieg und beim Verlassen des Aufnahmebereichs ist größer. Zudem wird eine höhere körperliche Beweglichkeit der Nutzer:innen vorausgesetzt. Das schränkt den Personennutzungskreis zusätzlich ein. Im Gegensatz zu Konstruktion 1 ist diese Konstruktion bei der Bewegung des U-Rahmens stabil und eignet sich für die Aufnahme von Bildern. Wird der U-Rahmen bewegt und zur Bildaufnahme gestoppt, schwingt der Rahmen weiterhin. Ursächlich ist die Achse, die den Motor mit

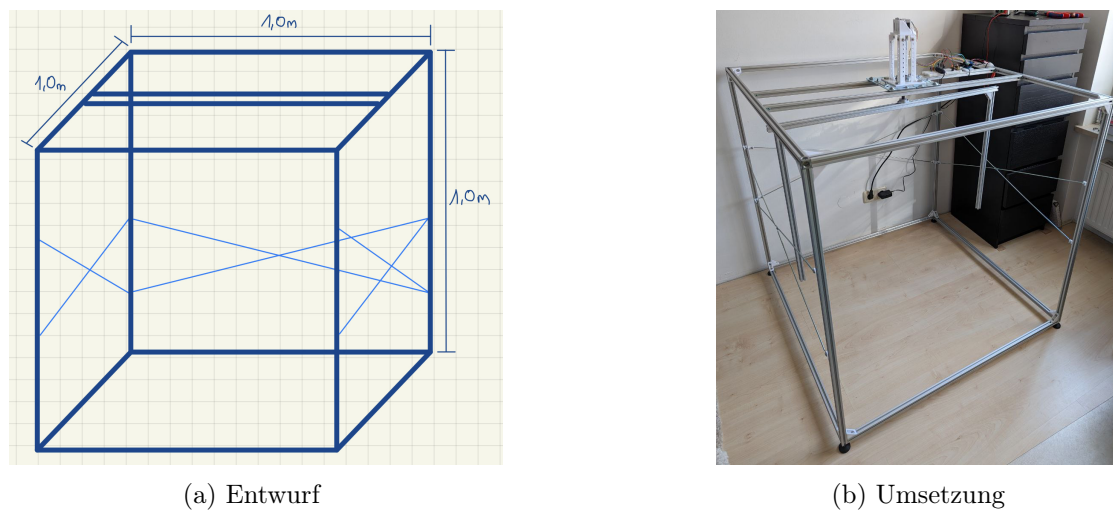


Abbildung 7.2: Konstruktion 2

dem U-Rahmen verbindet. Durch Kürzung der Achslänge kann bereits eine höhere Stabilität erreicht werden. Durch eine Achse mit größerem Radius sind weitere Verbesserungen erwartbar.

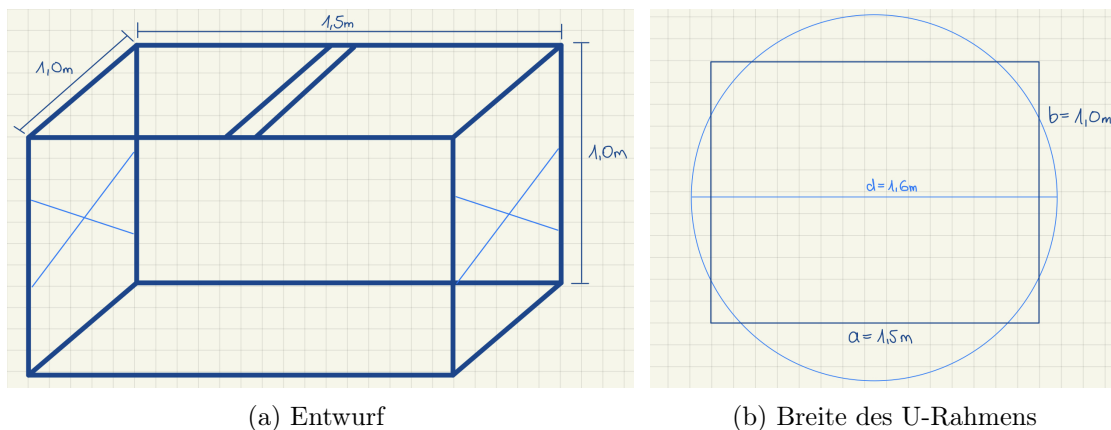
Im Verlauf der Bearbeitungszeit hat sich herausgestellt, dass die Kamera einen größeren Abstand zur Person benötigt und die bestehende Anordnung nicht ausreichend Möglichkeiten dafür bietet. Ursprünglich wurde angenommen, dass ein Abstand von 0,4 m zur Gesichtsfläche ausreichend ist. Aufgrund der ungleichmäßigen Form eines Kopfes und der nicht exakten Positionierung der Personen im Aufnahmebereich ist ein größerer Abstand zwischen Kamera und Person notwendig, weil die Kamera abhängig von der Auflösung erst ab einem definierten Abstand Tiefendaten erfassen kann. Um diese Anforderung zu erfüllen wird Konstruktion 2 erweitert wodurch Konstruktion 3 entsteht ist.

Konstruktion 3

Die dritte Konstruktion ist quaderförmig und besitzt eine Grundfläche von 1,5 m x 1,0 m und eine Höhe von 1,0 m, siehe Abbildung 7.3a und 7.4. Für den U-Rahmen wird ein Aluprofil mit einer Länge von 1,6 m verwendet, wodurch ein größerer Abstand zwischen Kamera und Person ermöglicht wird, siehe Abbildung 7.3b.



Abbildung 7.4: Konstruktion 3



(a) Entwurf

(b) Breite des U-Rahmens

Abbildung 7.3: Konstruktion 3 mit U-Rahmen

Wie aus Tabelle 7.1 hervorgeht erfüllt Konstruktion 1 Anforderungen an Personengröße, Zugänglichkeit, Sicherheit und Lagerung. Insbesondere bei der späteren Verwendung durch Laien sind diese Punkte von großer Wichtigkeit, um die Personen einerseits nicht zu gefährden und das Produkt andererseits für einen großen Nutzungskreis zur Verfügung zu stellen. Da die Stabilität der Konstruktion die Grundlage für eine unverwackelte Bildaufnahme darstellt, werden die Anforderungen an die Personengröße und an das Quetschen, Stoßen, Stolpern für die Entwicklungsphase geringer priorisiert. Durch die Entwicklung von Konstruktion 2 und 3 kann die Stabilität des U-Rahmens und damit die Stabilität der Kamera bei der Bildaufnahme verbessert und die Vibrationen verringert werden.

Tabelle 7.1: Vergleich der Konstruktionen

	1	2	3
Personengröße nach HW-A 1	+	o	o
Lagerung nach HW-A 6	+	+	-
Quetschen, Stoßen, Stolpern	+	o	o
Zugänglichkeit	+	-	o
Stabilität, Vibrationen	-	o	o
Entfernung zur Kamera	o	o	+

Tabelle 7.2: Aluprofilkonstruktion - Angaben nach [14]

Menge	Bezeichnung	Maße	Kosten
4	Aluprorfil 20x20L B-Typ Nut 6	1500 x 20 x 20 mm	34,80 €
1	Aluprorfil 20x20L B-Typ Nut 6	1600 x 20 x 20 mm	9,53 €
6	Aluprorfil 20x20L B-Typ Nut 6	1000 x 20 x 20 mm	34,80 €
2	Aluprorfil 20x20L B-Typ Nut 6	600 x 20 x 20 mm	5,88 €
8	Innenwinkel 20 B-Typ Nut 6		10,96 €
50	Nutenstein	Gewinde: M5	47,00 €
50	Flachrundschraube	M5 x 8 mm	13,00 €
2	Würfelverbinder B-Typ Nut	20 x 20 x 20 mm	11,02 €
		Gesamtkosten	166,99 €

Im vorliegenden Fall ist Konstruktion 3 die geeignetste Lösung. Eine Aufstellung der Gesamtkosten mit den längeren Profilen ist in Tabelle 7.2 zu finden.

7.1.2 Motorhalterung

Die Motorhalterung wird mit Fusion 360 konstruiert und mit einem 3D-Drucker gedruckt. Bei der Verwendung hat sich gezeigt, dass zwischen Schrittmotor und Motorhalterung ein etwa 1,0 mm breiter Spalt ist, wodurch der Motor zu Beginn jeder Rotationsbewegung innerhalb der Halterung einmalig wenige Grad mitrotiert und zusätzlich den U-Rahmen in Schwingung versetzt. Ursächlich für den Spalt sind Ungenauigkeiten des 3D-Druckers beim Drucken von vorgegebenen Maßen. Bei der Konstruktion ist die Breite des Schrittmotors anhand des Datenblatts [30] mit 42,3 mm angenommen worden. Für den Druck ist von einer Abweichung von 0,5 mm ausgegangen worden, weshalb die Halterung mit 41,8 mm kleiner konstruiert worden ist, siehe Abbildung 7.5. Das Problem kann durch Ausfüllen des Zwischenraums gelöst werden.

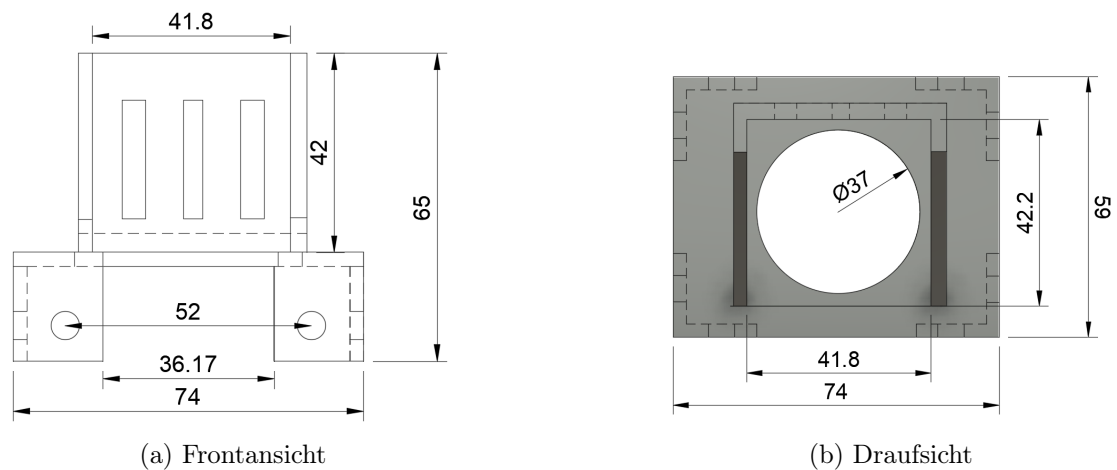


Abbildung 7.5: Motorhalterung - Maße in mm

7.1.3 Montage der Sensoren

Während der Bearbeitung hat sich herausgestellt, dass eine höhenverstellbare Kamera nicht notwendig ist. Bei der Verwendung von einer Kamera, die auf Augenhöhe der zu scannenden Person angebracht wird, werden bei der Aufnahme aus verschiedenen Perspektiven genügend Informationen vom Kopf aufgenommen, siehe Abbildung 7.6. In der Abbildung wird sichtbar, dass bei der Testperson auch Daten unter dem Kinn und unter der Nase durch die seitlichen Perspektiven erfasst werden. Gleichzeitig wird aus der genannten Perspektive die Oberseite des Kopfes erfasst. Eine zweite Kamera, die Aufnahmen von unterhalb des Kinns macht, wie es in Konzept 2 in Kapitel 6.1.1 ab Seite 46 beschrieben wird, ist dennoch für zukünftige Arbeiten empfehlenswert. Grund hierfür sind die anatomischen Unterschiede zwischen den Benutzer:innen und die Ungenauigkeiten bei der Ausrichtung im Scanner.

7.2 Software

In diesem Abschnitt wird die Umsetzung der Software nach dem Konzept aus Kapitel 5.2 beschrieben. Die Software ist unterteilt in die Module Bedienoberfläche, Aufnahme der Punktwolken und Bildverarbeitung.

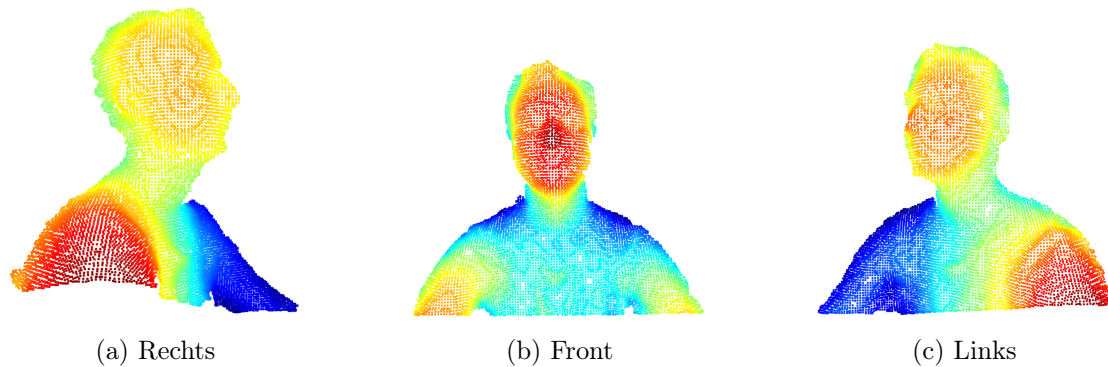


Abbildung 7.6: Darstellung des Gesichts durch Punktwolken aus drei Perspektiven

7.2.1 Bedienoberfläche

Die Bedienoberfläche wird unter Verwendung von [34] mit dem Qt-Designer¹ erstellt. Die Bedienoberfläche implementiert die Funktionen aus Abbildung 6.8 und ist in der Klasse *GUI* implementiert. Alle verwendeten Icons werden mit GIMP² erstellt.

Startbildschirm

Der Startbildschirm wird beim Programmstart angezeigt und stellt im Hauptmenü die drei Auswahlmöglichkeiten *Aufnahme*, *Punktwolken* und *3D-Modell* zur Verfügung, siehe Abbildung 7.7. Der Start der jeweiligen Anwendungen erfolgt ereignisgesteuert durch Klick auf eine Schaltfläche.

Aufnahme

Nach einem Klick auf die Schaltfläche *Aufnahme* auf dem Startbildschirm wird der in Abbildung 7.8 dargestellte Bildschirm angezeigt und die Benutzer:innen können auswählen, ob sie den Scanner alleine oder zu zweit verwenden. Im oberen Bereich soll zukünftig eine Live-Ansicht der Kamera dargestellt werden, die in dieser Arbeit aus zeitlichen Gründen nicht realisiert werden konnte. Nach dem Klick auf *Ich bin alleine* erscheint ein Countdown, der in Abbildung 7.9 dargestellt ist und über die Schaltfläche *Starte den Countdown* gestartet wird. Nach Ablauf beginnt die Bildaufnahme. Bei Auswahl der Schaltfläche *Wir sind zu zweit* wird kein Countdown angezeigt. Die Aufnahme kann zu jedem Zeitpunkt über den Stopp-Button abgebrochen werden. Über den Button mit dem Haus in der rechten Ecke gelangen die Benutzer:innen zurück zum Startbildschirm.

¹<https://doc.qt.io/qt-6/qtdesigner-manual.html> - Zugriffsdatum: 07.09.2022

²<https://www.gimp.org/> - Zugriffsdatum: 02.09.2022



Abbildung 7.7: Startbildschirm



Abbildung 7.8: Aufnahme

Punktwolken und 3D-Modell

Über die Schaltfläche *Punktwolken* auf dem Startbildschirm können die aufgenommenen Punktwolkendaten geöffnet werden. Das erfolgt mit der Methode `open_saved_ply`, die im Explorer den Pfad zu den zuletzt aufgenommenen Punktwolken anzeigt. Über den Explorer der in Abbildung 7.10 dargestellt ist, können beliebige Ordner auf dem PC durchsucht werden. Geöffnet und angezeigt werden können nur Dateien mit der Endung

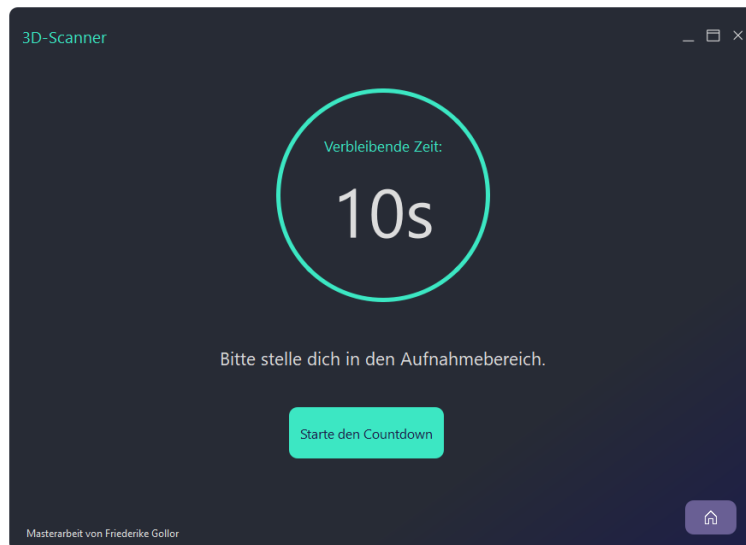


Abbildung 7.9: Countdown

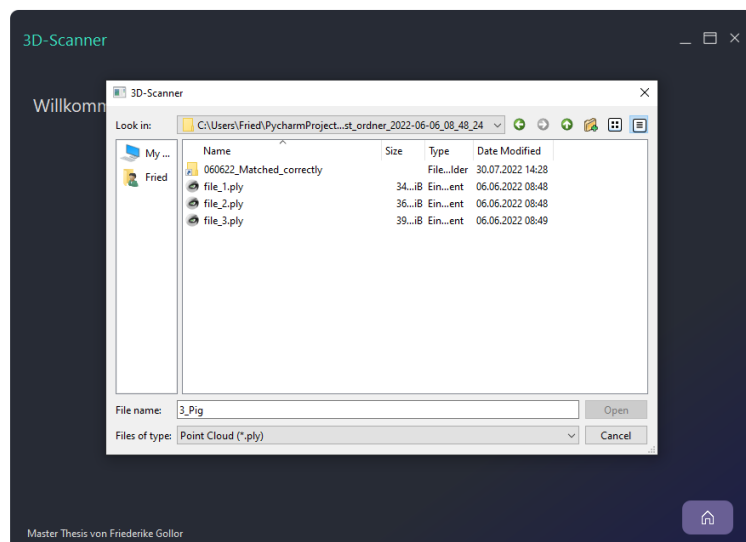
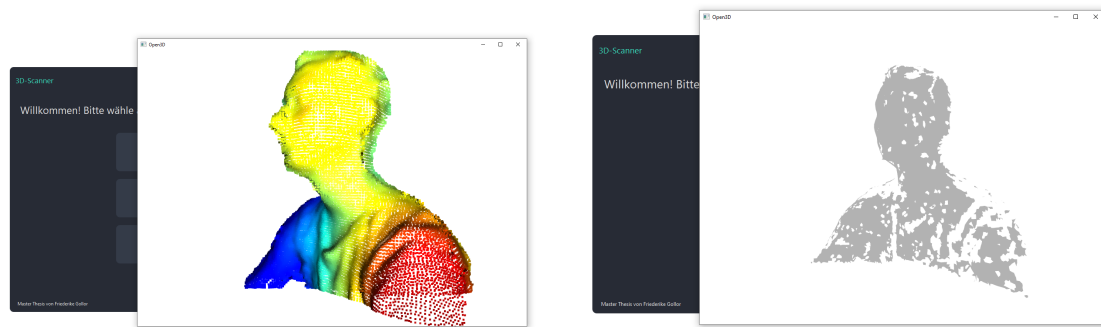


Abbildung 7.10: Exploreransicht

.ply. Mit Klick auf die Schaltfläche *3D-Modell* im Startbildschirm wird der Pfad zum zuletzt erstellten 3D-Modell im Explorer geöffnet und es können Dateien mit der Endung *.ply* mit rekonstruierter Oberfläche, *.stl* und *.obj* geöffnet werden. Für die Anzeige der Punktwolken und der 3D-Modelle werden Funktionen aus der Open3D-Bibliothek verwendet, siehe Abbildung 7.11b.



(a) Punktwolke

(b) Oberfläche

Abbildung 7.11: Visualisierung der Modelle

7.2.2 Aufnahme der Punktwolken

Die Aufnahme der Punktwolken wird in der Klasse *ImageCapture* implementiert. Die Klasse ist die Schnittstelle zur Kamera und verwendet dafür das Framework Pyrealsense aus dem Intel RealSense SDK.

Die Methode *get_data_from_depth_stream* startet und beendet die Kameraaufnahme. Sie initialisiert den Kamerastream mit *initialize_stream* und startet den Stream mit *start_stream* nachdem überprüft wurde, ob eine Kamera verbunden ist.

Readme erstellen

Nach dem Beenden der Bildaufnahme wird eine Readme-Datei erstellt, die Informationen zur Kamera wie die Seriennummer, Firmwareversion, verwendeter Port am Laptop, die verwendete Auflösung, Frames per Pixel und die Namen der aufgenommenen Punktwolken enthält, siehe Abbildung 7.12. Die Informationen dienen der späteren Reproduzierbarkeit der Ergebnisse. Durch die Verwendung eines Dictionarys und der damit einhergehenden Speicherung der Informationen als Key-Value-Pair können die Einträge sowohl von Menschen gelesen werden, als auch von Programmen gezielt ausgewertet werden. Das Readme wird als *json*-Datei unter dem gleichen Pfad wie die Punktwolken gespeichert.

Logging

Für die Auswertung von Fehlern wird die Klasse *Logger* implementiert. Im Programmablauf werden mit der Methode *log_time* Informationen dokumentiert, wie zum Beispiel die Anzahl an durchgeführten Registrierungsversuchen, die erzielten Fitnesswerte

```
1  {
2      "device_product_line": "D400",
3      "device_name": "Intel RealSense D415",
4      "device_usb_type_descriptor": "3.2",
5      "serial_number": "038623021290",
6      "product_id": "OAD3",
7      "firmware_version": "05.12.14.50",
8      "physical_port": "\\.\usb#vid_8086&94699c46e44}\global",
9      "date": "2022-09-04 13:47:57.402980",
10     "resolution_width": 848,
11     "resolution_height": 480,
12     "FPS": 60,
13     "Files": [
14         ".../data\data_2022-09-04_13_47_42\pcd_13_47_42.835.ply",
15         ".../data\data_2022-09-04_13_47_42\pcd_13_47_49.260.ply"
16     ]
17 }
```

Abbildung 7.12: Readme

oder unter welchem Pfad die Punktwolken gespeichert werden. Kommt es im Entwicklungsprozess gehäuft zu Fehlern in der Ausführung des Programms können vergangene Programmabbrüche vergleichend herangezogen werden. Die Log-Datei wird durch das Exceptions-Handling, *deutsch* Ausnahmebehandlung, auch bei Programmabbrüchen angelegt. Beispielhaft wird die Funktion `click_start_two_people` herangezogen, die in Listing 7.1 dargestellt ist, die in der Klasse `MainWindow` die Bildaufnahme ausführt, wenn in der Bedienoberfläche die Schaltfläche `Wir sind zu zweit` gedrückt wurde. Im `try`-Zweig ist der auszuführende Code enthalten, im `except`-Zweig wird der Fehler, wenn es ein `ValueError` ist dem Log hinzugefügt und der `finally`-Zweig enthält die Funktion `store_logs_in_logfile` die die Logdatei zum Ende der Programmausführung anlegt. Die Logdateien werden in dem Ordner mit den aufgenommenen Punktwolken gespeichert.

Dateimanagement

Für jede neue Aufnahme wird ein Ordner zum Speichern der aufgenommenen Punktwolken, der Logdateien und des Readmes' erstellt. An den vorgegebenen Namen des Ordners wird Datum und Uhrzeit der Aufnahme angehängt. So soll ein Überschreiben von Dateien bei mehreren Aufnahmen vermieden werden. Vor der Erstellung des Ordners wird geprüft, ob unter dem angegebenen Dateipfad bereits ein Ordner mit dem geplanten Namen vorhanden ist. Ist das der Fall, wird der Name erweitert.

```

def click_start_two_people(self):
    try:
        stream = ImageCapture(path=self.path,
                               folder=self.__folder,
                               filename=self.__filename,
                               dlogger=self.__dlogger)
        self.__pathImages = stream.get_data_from_depth_stream(
            self.__resolution_width,
            self.__resolution_height,
            self.__fps)
        self.stackedWidget.setCurrentWidget(self.widget_save_content)
    except ValueError as error:
        self.__dlogger.log_time(error)
        error_type, error_value, error_traceback = sys.exc_info()
        traceback.print_exception(error_type, error_value,
                                   error_traceback)
        self.__dlogger.log_time(error_type)
        self.__dlogger.log_time(error_value)
        self.__dlogger.log_time(error_traceback)
    finally:
        self.__dlogger.store_logs_in_logfile()

```

Listing 7.1: Exception-Handling

7.2.3 Bildverarbeitung

7.2.3.1 Punktwolkenvorverarbeitung

Die Bildverarbeitung wird in der Klasse *EditPointCloud* umgesetzt und nach dem Aktivitätsdiagramm aus Abbildung 6.13 von Seite 60 implementiert. In Listing 7.2 ist die Methode *process_background_removal* dargestellt, die den Hintergrund der Punktwolken mit *cluster_w_bgrnd* und die Ausreißer mit *clusters_w_otlrs* entfernt. Für die Verwendung des DBSCAN-Algorithmus müssen die Punktwolken in ein numpy-Array konvertiert werden. Nach Abschluss der Ausführung wird die Punktwolken wieder in das Open3D-Punktwolkenformat konvertiert.

7.2.3.2 Registrierung

Die Registrierung wird in der Klasse *Registration* mit der Bibliothek Open3D umgesetzt und in der Klasse *ManagePointClouds* aufgerufen. In *ManagePointClouds* werden der Klasse die Source- und Target-Punktwolken übergeben. Nach [46] wird nach der globalen Registrierung mit RANSAC der ICP-Algorithmus zur lokalen Ausrichtung durch-

```

def process_background_removal(self):
    # Convert Open3D format to numpy array
    self.convert_o3d_format_to_numpy_array()

    # Remove background
    cluster_w_bgrnd = self.compute_dbscan(
        eps=0.1, min_samples=1000, clusters=self.pcd)
    get_max_cluster = self.dbscan_max_mean_of_clusters(
        clusters=cluster_w_bgrnd)

    # Remove outliers
    clusters_w_otlrs = self.compute_dbscan(
        eps=0.008, min_samples=10,
        clusters=cluster_w_bgrnd[get_max_cluster])
    get_max_shape = self.dbscan_max_mean_of_shape(
        clusters=clusters_w_otlrs)
    self.pcd = clusters_w_otlrs[get_max_shape]

    # Convert numpy array to Open3D format
    self.convert_numpy_array_to_o3d_format(self.pcd)

    return self.pcd

```

Listing 7.2: Algorithmus zur Clusteranalyse

geführt. Beide Algorithmen berechnen einen Fitnesswert der nach [47] das Verhältnis zwischen der Anzahl an Punkten im Überschneidungsbereich und der Anzahl an Punkten in der Zielpunktwolke angibt. Der Fitnesswert liegt zwischen 0 und 1. Je größer der Wert ist, desto größer ist der Überschneidungsbereich und desto mehr Gemeinsamkeiten zwischen den Punktwolken konnten gefunden werden, siehe Abbildung 7.14b. Ein Fitnesswert von eins gibt an, dass die Punktwolken identisch sind und sich vollständig überschneiden, siehe Abbildung 7.14c. Unter der Annahme, dass der Überschneidungsbereich und die Größen der Zielpunktwolke bei jeder zum Modell hinzugefügten Punktwolke ähnlich groß sind, sind ähnliche Fitnesswerte zu erwarten. Diese Annahme wird bei Verwendung des von Open3D implementierten Fitnesswerts nicht erfüllt. Stattdessen sinkt dieser. In der Open3D-Bibliothek auf GitHub [48] im MasterBranch vom 25.06.2022 mit dem Commit-Hash 55ded67 wird der Fitnesswert statt mit der Zielpunktwolke mit der Quellpunktwolke berechnet, siehe Listing 7.3. Wie in Kapitel 6.2.5.2 auf Seite 60 beschrieben, wird die Quellpunktwolke mit jeder Registrierungsschleife größer, das Verhältnis zum Überschneidungsbereich also kleiner und damit sinkt der Fitnesswert. Um sicherzustellen, dass die registrierten Punktwolken einen großen Überschneidungsbereich besitzen, werden RANSAC und ICP jeweils in while-Schleifen implementiert die ausgeführt werden bis ein Abbruchkriterium erfüllt wird, siehe Abbildung 7.13a. Abbruch-

```

[...]
else {
    size_t corres_number = result.correspondence_set_.size();
    result.fitness_ = (double)corres_number / (double)source.points_.
        size();
    result.inlier_rmse_ = std::sqrt(error2 / (double)corres_number);
}
[...]
```

Listing 7.3: Berechnung des Fitnesswertes in der Open3D-Bibliothek aus [48]

```

[...]
computed_ransac_fitness =
    ((np.asarray(pcds_ransac.correspondence_set).size)/2) / \
    (np.asarray(source_pcd.pcd.points).size /3)
[...]
```

Listing 7.4: Berechnung des Fitnesswertes mit der Zielpunktwolke

kriterium ist der Fitnesswert. Damit ein gleichbleibender Schwellwert verwendet werden kann wird der Fitnesswert nach Listing 7.4 mit der Zielpunktwolke berechnet. Die Registrierung nach der Open3D-Bibliothek erfolgt ohne Schwellwerte und ist in Abbildung 7.13 dargestellt. Wird der RANSAC-Fitnesswert überschritten, wird der ICP-Algorithmus nach der gleichen Vorgehensweise ausgeführt. Ist die Registrierung erfolgreich wird die registrierte Punktwolke als neue Source-Punktwolke zurückgegeben. Für den Fall, dass keine Übereinstimmungen oberhalb der Schwellwerte gefunden werden können, wird mit *check_fitness_values_for_changes* die Änderung des Fitnesswerts in den letzten fünf Iterationen überprüft. Hat sich der Wert nicht verändert wird die Durchführung der Registrierung beendet und die Methode in der Klasse *ManagePointClouds* erneut aufgerufen. Führt auch das zu keinem Ergebnis wird die Programmausführung mit einem *ValueError* abgebrochen.

7.2.3.3 Oberflächenrekonstruktion

Wie in Abschnitt 6.2.5.3 auf Seite 61 geschrieben, wird die Oberflächenrekonstruktion gemäß [51] implementiert und erfolgt mit dem Ball Pivoting Algorithmus. Wie in Abbildung 7.15 dargestellt enthält die Oberfläche Löcher aufgrund von fehlenden Oberflächeninformationen. In Abbildung 7.15a ist die Ausgangspunktwolke dargestellt. Die Abbildung 7.15b zeigt die Rekonstruktion der Oberfläche mit einer Genauigkeit, auch Level of Detail, kurz LoD, von maximal 300 Dreiecken, die die Oberfläche rekonstruieren.

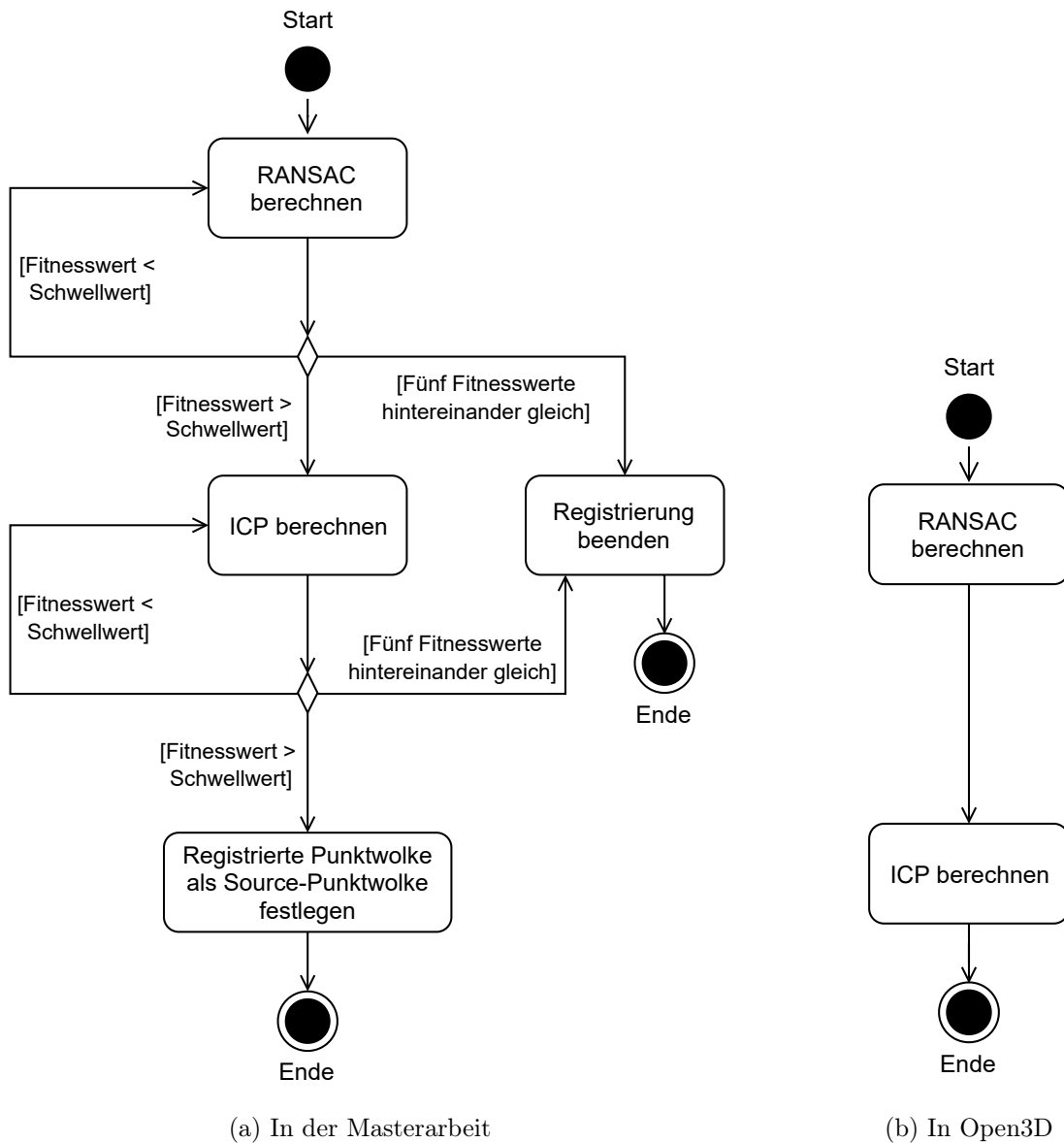


Abbildung 7.13: Umsetzung der Registrierung

Eine höhere Detailtiefe mit maximal 1.000.000 Dreiecken als Oberfläche wird in Abbildung 7.15c dargestellt.

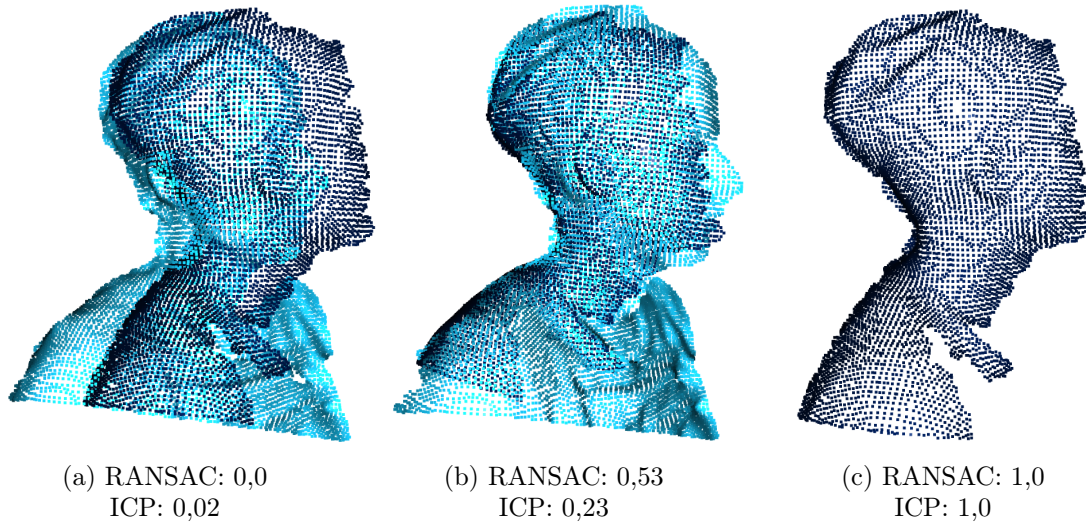


Abbildung 7.14: Vergleich der Fitnesswerte

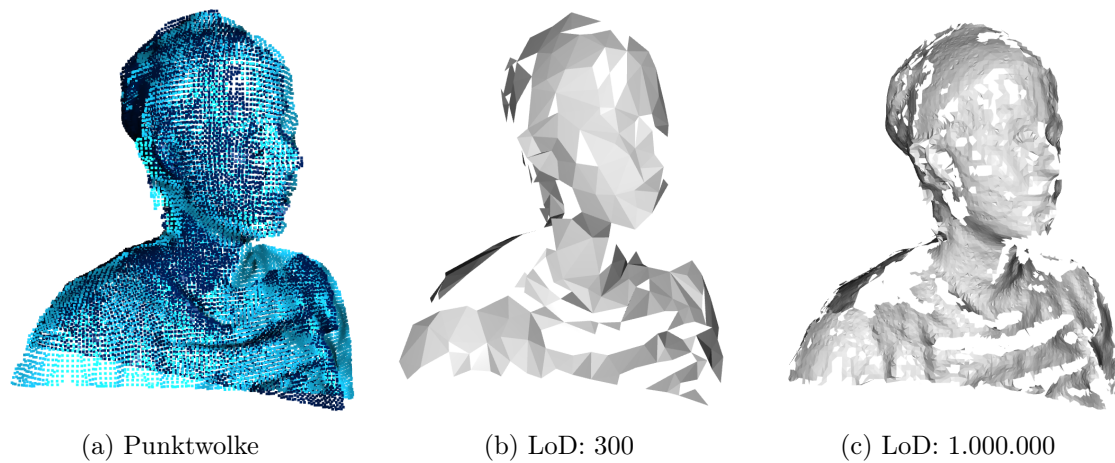


Abbildung 7.15: Vergleich der rekonstruierten Oberflächen mit der Punktwolke

8 Test

In diesem Kapitel wird die entwickelte Hard- und Software anhand der gestellten Anforderungen aus Kapitel 4 ab Seite 16 getestet und anhand der aus [22] entnommenen Kriterien getestet und beurteilt:

- Testgegenstand, *hier* die zu testende Anforderung
- Testdurchführung
- Testkriterien
- Beobachtungen
- Beurteilung

Eine Anforderung gilt als erfüllt, wenn alle in der Anforderung genannten Kriterien ohne Einschränkung umgesetzt worden sind. Teilweise erfüllt ist eine Anforderung, wenn nicht alle Kriterien oder die Kriterien nur in Teilen erfüllt sind. Als nicht erfüllt gilt eine Anforderung, wenn die Kriterien in keinem Punkt erfüllt worden sind.

8.1 Hardware

8.1.1 Allgemein

Der Test der allgemeinen Anforderungen aus Abschnitt 4.6.1.1 ab Seite 24 wird in Tabelle 8.1 beschrieben.

Tabelle 8.1: Anforderungen: Allgemein

HW-A 1 (NF)		Teilweise erfüllt
Durchführung	Abstandsmessung: Fußboden - U-Rahmen abzüglich Toleranzbereich von 2 cm	
Kriterien	Geeignet für Personen zwischen 1,56 m und 1,9	
Beobachtung	Die Konstruktion ist für die Verwendung im Sitzen konstruiert worden. Der Abstand zwischen Fußboden und U-Rahmen beträgt 96 cm. Aufgrund von Unterschieden in der Oberkörperlänge können nicht alle Personen, die die Anforderung an die Gesamtpersonengröße erfüllen, die Konstruktion verwenden, siehe Abschnitt 7.1.1. In der Anordnung finden Personen mit einer Oberkörperlänge bis 94 cm Platz.	
Urteil	Die Konstruktion ist für einen eingeschränkten Personenkreis nutzbar.	

HW-A 2 (NF)		Nicht erfüllt
Durchführung	Sicht- und Funktionsprüfung der Konstruktion	
Kriterien	Der Sensor bewegt sich elektrisch angetrieben um die Person.	
Beobachtung	Die Kamera wird mit einem U-Rahmen, der von einem Schrittmotor gedreht wird, um die Person im Aufnahmebereich bewegt. Wird keine Kamera befestigt, dreht sich der U-Rahmen um 50 Schrittmotorsteps mit einer Geschwindigkeit von 60 rpm (revolution per minute <i>deutsch</i> : Umdrehungen pro Minute). Beim Stopp an einer Position schwingt der Rahmen nach. Eine Erhöhung der Geschwindigkeit auf >70 rpm führt dazu, dass sich der U-Rahmen in ungleichmäßigen Abständen oder nicht vorwärts bewegt. Erhöht man anstelle der Geschwindigkeit die Anzahl der Steps, um größere Aufnahmewinkel einzustellen oder eine vollständige Umdrehung zu machen, dreht der Motor in unregelmäßig großen Schritten weiter oder er dreht durch. Wird der Motor ohne U-Rahmen angesteuert sind keine der genannten Einschränkungen festzustellen.	
Urteil	Der Schrittmotor ist in Verbindung mit den konstruktiven Gegebenheiten nicht für diese Anwendung geeignet, weshalb eine zuverlässige Nutzung nicht möglich ist. Die Anforderung ist nicht erfüllt.	

HW-A 3 (NF)	Erfüllt
Durchführung	Sichtung der Punktwolken
Kriterien	Einhaltung der gesetzlichen Vorgaben an einen Prototypen aus Abschnitt 4.4
Beobachtung	<p>Personen im Hintergrund sind auf den unverarbeiteten Rohdaten erkennbar, auf dem fertigen Modell nicht mehr. Im Laufe der Verarbeitung wird der Hintergrund entfernt und Personen im Hintergrund sind nicht mehr erkennbar, jedoch nicht identifizierbar. Die Person im Vordergrund ist auf allen Aufnahmen vor dem Downsampling, also dem Reduzieren der Punkte in der Punktwolke, erkennbar und identifizierbar.</p> <p>Aufgrund der Aluprofilkonstruktion besteht die Möglichkeit den Scanbereich an zwei Seiten abzuschränken.</p>
Urteil	Der Prototyp steht nur einem kleinen Personenkreis mit technischen Vorkenntnissen zur Verfügung die mit der Datenaufnahme rechnen. Die Anforderung wird erfüllt.

HW-A 4 (NF)		Erfüllt
Durchführung	Aufstellung der Gesamtprojektkosten	
Kriterien	Budget von 500 € nicht überschreiten	
Beobachtung	<p>Die Gesamtkosten betragen 405,07 € und setzen sich wie folgt zusammen:</p> <p>Schrittmotor und -steuerung nach Tabelle 6.6: 57,48 €</p> <p>Motorbefestigung nach Tabelle 6.5: 24,64 €</p> <p>Kamera und Halterung (nur eine Kamera) nach Tabelle 6.3: 155,95 €</p> <p>Aluprofilkonstruktion nach Tabelle 7.2: 166,99 €</p> <p>Nicht in der Aufstellung enthalten sind die Kosten für das 3D-Drucker Filament zum Drucken von Bauteilen für die Konstruktion, wie beispielsweise Ecken zur Stabilisierung, sowie Kosten für zusätzliche Nutzensteine und Schrauben. Die Kosten dafür sind kleiner als 100 €. Das Budget wird somit nicht überschritten.</p> <p>Ebenfalls nicht enthalten sind fünf ein Meter lange Aluprofile die für die Konstruktion 1 und 2 verwendet worden sind und in Konstruktion 3 durch längere Profile ersetzt worden sind. Die Kosten werden nicht aufgeführt, da die Profile in anderen Projekten weiterverwendet werden können.</p>	
Urteil	Das Budget wird eingehalten.	

HW-A 5 (NF)		Erfüllt
Durchführung	Prüfen auf Vielseitigkeit der Einsatzmöglichkeiten	
Kriterien	Wiederverwendbarkeit des Materials für andere Anwendungen	
Beobachtung	<p>Aus dem vorhandenen Material konnten drei verschiedene Konstruktionen aufgebaut werden. Zudem sind Aluprofile in Masterarbeiten wie der von Malte Dittmar [13] für andere Anwendungen verwendet worden.</p>	
Urteil	Das Material ist wiederverwendbar.	

HW-A 6 (NF)	Teilweise erfüllt
Durchführung	Maße des Materials gegen die angegebenen Maße prüfen
Kriterien	Material darf in Einzelteilen die Maße 0,6 m x 0,6 m x 1,2 m (L x B x H) und das Gewicht von 31,5 kg nicht überschreiten
Beobachtung	Die langen Aluprofile aus Konstruktion 3 überschreiten mit 1,5 m und 1,6 m die angegebenen Maße und können auch nicht diagonal verstaut werden. Alle anderen Bauteile unterschreiten die angegebenen Maße.
Urteil	Die Anforderung wird teilweise erfüllt.

8.1.2 Sicherheit der Hardware

Die Erfüllung oder nicht Erfüllung der Anforderungen an die Sicherheit der Hardware aus Abschnitt 4.6.1.2 ab Seite 25 wird in der folgenden Tabelle 8.2 beschrieben.

Tabelle 8.2: Anforderungen: Sicherheit

HW-SI 1 (NF)	Erfüllt
Durchführung	Prüfung des Datenblatts der betreffenden Komponenten
Kriterien	Anwendung der DIN EN 60825-1 oder IEC 60825-1 (VDE 0837-1)
Beobachtung	Die Intel RealSense D415 Kamera erfüllt laut Datenblatt [26] die Norm IEC 60825-1:2007 und verwendet einen Laser der Klasse 1.
Urteil	Die Gefährdung für Personen ist laut IEC 60825-1:2007 bei einem Laser der Klasse 1 gering.

HW-SI 2 (NF)	Teilweise erfüllt
Durchführung	Zugänglichkeit der Konstruktion prüfen sowie Prüfung der Schutzeinrichtungen auf Funktionalität
Kriterien	Ausschluss mechanischer Gefährdung durch Quetschen, Stoßen, Stolpern oder durch bewegte Teile
Beobachtung	Der Zugang zum Aufnahmebereich ist, wie in Abschnitt 7.1.1 beschrieben, nach oben und unten beschränkt. Der U-Rahmen besitzt keine Absturzsicherung durch einen Splint.
Urteil	Die Konstruktion weist Mängel in Bezug auf die Sicherheit auf und erfüllt die Anforderung teilweise.

HW-SI 3 (NF)	Erfüllt
Durchführung	Sichtprüfung der Konstruktion
Kriterien	Kanten, Spitze oder Oberflächenrauigkeit müssen für die Benutzer:innen durch Schutzeinrichtungen unzugänglich sein
Beobachtung	Die Konstruktion weist eine glatte Oberfläche auf und besitzt runde Ecken.
Urteil	Für die Benutzer:innen besteht kein erhöhtes Verletzungsrisiko.

HW-SI 4 (NF)	Erfüllt
Durchführung	Sichtprüfung der elektrischen Teile der Konstruktion sowie Spannungsmessung an offen zugänglichen Bauteilen
Kriterien	Gleichspannungen bis 120 V werden nach DIN EN 61140: 2016-11 als für den Menschen bei Berührung nicht gefährliche Kleinspannungen definiert. Um die Sicherheit für die Laienanwender:innen zu erhöhen, wird die Richtlinie 2009/48/EG zur Sicherheit von Spielzeug vom Europäischen Parlament und Rat angewendet. Sie legt fest, dass an keinem spannungsführenden zugänglichen Bauteil eine Spannung von 24 V überschritten werden darf. [15]
Beobachtung	Es wurde eine maximale Spannung von 12 V an der Spannungsquelle gemessen.
Urteil	Spannungsführende Teile sind offen für die Benutzer:innen zugänglich, gefährden die Benutzer:innen aufgrund der Unterschreitung der 24 V Grenze aber nicht. Für die Verwendung auf Veranstaltungen wird ein Gehäuse für die Elektronik empfohlen.

HW-SI 5 (F)	Nicht erfüllt
Durchführung	Prüfen der Abschaltvorrichtungen der elektrischen Komponenten
Kriterien	Elektrisch bewegte Teile sind abschaltbar.
Beobachtung	Elektrisch bewegt wird der U-Rahmen durch den Schrittmotor. Dieser ist über die Spannungsquelle durch einen Kippschalter abschaltbar.
Urteil	Für Laienanwender:innen ist die Form der Abschaltung nicht eindeutig. Zur Erfüllung dieser Anforderung muss ein sichtbar gekennzeichnete und leicht zugänglicher Schalter montiert werden.

8.1.3 Sensor

Dieser Abschnitt beschreibt in Tabelle 8.3 den Test der Anforderungen an den Sensor aus Abschnitt 4.6.1.3 ab Seite 26. Zur Bestätigung der Anforderung wird das Datenblatt der ausgewählten Kamera Intel RealSense D415 [26] verwendet.

Tabelle 8.3: Anforderungen: Sensor

HW-SN 1 (NF)		Erfüllt
Durchführung	Prüfung des Datenblatts	
Kriterien	Arbeitsbereich zwischen 0,3 m und 2,2 m bezogen auf den Abstand zwischen Kamera und Objekt	
Beobachtung	Der Abstand ist abhängig von der eingestellten Auflösung der Kamera und beginnt bei einer Auflösung von 424x240 mit 0,16 m und endet bei 1280x720 mit 0,45 m. Die Werte geben den minimalen Abstand zwischen Kamera und Objekt an. Der maximale Abstand wird mit 10 m angegeben, wobei es zu Abweichung von $\pm 2\%$ kommt. Der ideale Arbeitsbereich wird mit bis zu 3 m angegeben.	
Urteil	Der Arbeitsbereich der Kamera stimmt mit den Anforderungen überein.	

HW-SN 2 (F)		Erfüllt
Durchführung	Prüfung des Datenblatts	
Kriterien	Sensor ist für den Innenbereich geeignet	
Beobachtung	Der Sensor ist für den Innenbereich geeignet.	
Urteil	Die Anforderung wird erfüllt.	

HW-SN 3 (F)		Nicht erfüllt
Durchführung	Prüfung des Datenblatts	
Kriterien	Integration weiterer Sensoren in der Anordnung	
Beobachtung	Die Kamera bietet die Möglichkeit zur Integration weiterer Intel Kameras über eine Schnittstelle an. Im Aufbau wird nur eine Kamera verwendet.	
Urteil	Die optionale Anforderung wird nicht umgesetzt.	

8.2 Software

Dieser Abschnitt beschreibt die Überprüfung der Anforderungen an die Software nach Abschnitt 4.6.2 ab Seite 27 in der nachfolgenden Tabelle 8.4.

Tabelle 8.4: Anforderungen: Software

SW 1 (NF)		Erfüllt
Kriterien	Die entwickelte Software ist auf dem Windows-Betriebssystem Version 10 lauffähig.	
Beobachtung	Die Software wird auf einem Laptop mit Windows 10 getestet.	
Urteil	Die Anforderung wird erfüllt.	

SW 2 (NF)		Erfüllt
Durchführung	Prüfung der Programmstruktur	
Kriterien	Das Programm ist modular.	
Beobachtung	Das Programm ist in Klassen unterteilt, die nach Funktionen unterteilt sind. Dabei ist zwischen Klassen, die Basisfunktionen enthalten, wie <i>EditPointCloud</i> , <i>Registration</i> , <i>CommunicationToESP</i> , <i>SurfaceReconstruction</i> , <i>Logger</i> und <i>ImageCapture</i> , und Klassen, die zusammenführende Funktion zur Ausführung des Programms wie <i>ManagePointCloud</i> und <i>MainWindow</i> enthalten, zu unterscheiden. Die Methoden der Basisklassen sind in anderen Kontexten ähnlichen Programmen nutzbar. Mit Ausnahme von <i>ImageCapture</i> und <i>CommunicationToESP</i> sind sie dabei nicht von anderen Klassen abhängig.	
Urteil	Die Software ist modular und nach Funktionen unterteilt. Die Module können wiederverwendet und weiterentwickelt werden.	

SW 3 (F)		Erfüllt
Durchführung	Prüfung auf Vorhandensein einer Bedienoberfläche	
Kriterien	Bedienoberfläche ist vorhanden	
Beobachtung	Beim Start des Programms wird eine Bedienoberfläche geöffnet.	
Urteil	Die Interaktion zwischen Anwender:innen und System ist möglich.	

SW 4 (F)		Erfüllt
Durchführung	Prüfung auf Vorhandensein und Test auf Funktionsfähigkeit	
Kriterien	Start der Anwendung über die Bedienoberfläche	
Beobachtung	Die Aufnahme kann über das Menü der Bedienoberfläche gestartet werden.	
Urteil	Die Anforderung wird erfüllt.	

SW 5 (F)		Nicht erfüllt
Durchführung	Prüfung auf Vorhandensein und Test auf Funktionsfähigkeit	
Kriterien	Beenden der Aufnahme vor Fertigstellung	
Beobachtung	Während der Aufnahme ist keine Interaktion mit der Bedienoberfläche möglich. Eine Funktion zum Abbruch der Aufnahme wird der Benutzer:in nicht angezeigt.	
Urteil	Die Anforderung wird nicht erfüllt.	

SW 6 (F)		Erfüllt
Durchführung	Prüfen der Bedienart der Bedienoberfläche	
Kriterien	Zur Navigation auf der Bedienoberfläche wird eine Maus oder Tastatur verwendet.	
Beobachtung	Die Navigation erfolgt über die Maus.	
Urteil	Die Anforderung wird erfüllt.	

SW 7 (F)		Teilweise erfüllt
Durchführung	Prüfung auf Vorhandensein der Funktion durch Navigation durch die Bedienoberfläche	
Kriterien	<ul style="list-style-type: none"> • Die Bedienung der Anwendung ist durch eine einzelne Person durch Vorhandensein eines 10 sekündigen Countdowns möglich. • Überspringen oder Abwählen des Countdowns ist möglich. 	
Beobachtung	Nach der Auswahl von <i>Aufnahme</i> im Hauptmenü gelangen die Benutzer:innen in ein weiteres Auswahlmenü, in dem sie zwischen <i>Ich bin alleine</i> und <i>Wir sind zu zweit</i> wählen können. Nach dem Klick auf den zweiten Button startet die Aufnahme sofort. Beim Klick auf den ersten Button wird ein Countdown und ein Button mit <i>Starte den Countdown</i> angezeigt. Nach dem Klick auf den Button startet kein für die Benutzer:innen sichtbarer Countdown. Die Aufnahme startet jedoch zeitverzögert.	
Urteil	Die Anforderung wird teilweise erfüllt.	

SW 8 (F)		Nicht erfüllt
Anmerkung	Der Test dieser Anforderung wird in Kapitel 8.2.1.2 beschrieben.	

SW 9 (F)		Erfüllt
Durchführung	Prüfen der Ausgabedatei und öffnen der Datei in einem dafür geeigneten Programm, wie Ultimaker Cura ¹	
Kriterien	Speicherung des Modells im neutralen CAD-Dateiformat <i>stp</i> , <i>.stl</i> oder <i>.obj</i>	
Beobachtung	3D-Modelle werden nach der Oberflächenrekonstruktion als <i>.stl</i> gespeichert und können in CURA geöffnet werden.	
Urteil	Die Anforderung wurde umgesetzt.	

¹<https://ultimaker.com> - Zugriffsdatum: 09.09.2022

SW 10 (F)	Erfüllt
Durchführung	Prüfen der Speicherverwaltung
Kriterien	<ul style="list-style-type: none"> • Durchführung beliebig vieler Aufnahmen • Verwaltung des Speichers durch das Betriebssystem
Beobachtung	Alle Dateien und erstellten Ordner werden mit einem Zeitstempel versehen, womit ein Überschreiben einer Datei aufgrund des gleichen Namens ausgeschlossen werden soll. Zusätzlich prüft das Programm das Vorhandensein des vorgesehenen Dateinamens vor dem Speichern und erweitert diesen wenn der Name bereits vorhanden ist. Diese Funktion ist mit Pytest in der Test-Klasse <i>TestPointCloud.py</i> über den Test <i>test_save_pcd_as_ply</i> geprüft worden. Die Tests waren erfolgreich.
Urteil	Der Funktionsnachweis konnte mit dem Test erbracht werden und die Anforderung damit erfüllt werden.

Tabelle 8.5: Anforderungen: Software (optional)

SW 11 (F)	Nicht erfüllt
Durchführung	Visuelle Prüfung der 3D-Modelle
Kriterien	Erkennbare Texturen wie Haare
Beobachtung	Im 3D-Modell sind keine Texturen erkennbar.
Urteil	Die Anforderung ist nicht erfüllt.

SW 12 (F)	Nicht erfüllt
Durchführung	Prüfung der Bedienoberfläche auf Vorhandensein der Funktion
Kriterien	Löschen von Sensordaten über die Bedienoberfläche
Beobachtung	Es wird keine Funktion zum Löschen von Punktwolken in der Bedienoberfläche ausgewiesen. Auch über die Exploreransicht, die über die Bedienoberfläche geöffnet wird, können Dateien nicht entfernt werden.
Urteil	Die Funktion ist nicht umgesetzt.

SW 13 (F)		Teilweis erfüllt
Durchführung	Prüfung der Bedienoberfläche auf Vorhandensein der Funktion	
Kriterien	<ul style="list-style-type: none"> • Die Benutzer:innen können auswählen, dass ein 3D-Modell erstellt wird. • Die Benutzer:innen können auswählen, dass die aufgenommenen Daten verworfen, also im Speicher gelöscht werden. 	
Beobachtung	Nach der Aufnahme erscheint ein Auswahlmenü, über das die Benutzer:innen ein 3D-Modell erstellen lassen können (Ja) oder die Daten verwerfen können (Nein). Wird <i>Nein</i> gewählt, werden die Daten im Speicher nicht gelöscht.	
Urteil	Die Funktion ist gegeben, die Daten werden jedoch nicht gelöscht. Die Anforderung ist somit nur in Teilen erfüllt.	

SW 14 (F)		Erfüllt
Durchführung	Durchführen des Aufnahmeprozesses und Prüfung auf Vorhandensein der Funktion	
Kriterien	Anzeige des 3D-Modells nach Fertigstellung	
Beobachtung	Das 3D-Modell wird automatisch angezeigt.	
Urteil	Die Anforderung wird erfüllt.	

8.2.1 Bildverarbeitung

In diesem Abschnitt wird die Anforderung SW 8 zur Erstellung eines 3D-Modells eines Kopfes aus einer 360° Perspektive getestet. Die Erstellung setzt sich aus den drei Arbeitsschritten Punktwolkenvorverarbeitung, Registrierung und Oberflächenrekonstruktion zusammen. Für den Nachweis der Funktion werden die Module einzeln getestet.

8.2.1.1 Punktwolkenvorverarbeitung

Anforderung

Die Punktwolkenvorverarbeitung trennt die Person im Vordergrund vom Hintergrund und entfernt Rauschen.

Durchführung

Der Test erfolgt anhand einer mit einer Auflösung von 848x480 aufgenommenen Punktwolke bei der visuell überprüft wird, inwieweit sich die Ausgangspunktwolke nach der Anwendung des Algorithmus zum Entfernen des Hintergrunds und des Rauschens verändert.

Kriterien

1. Reduktion der Anzahl an Punkten in der Punktwolke durch das Entfernen des Hintergrunds
2. Visuell keine vereinzelt Punkte oder Cluster im direkten Umfeld der Punktwolke

Beobachtung

Abbildung 8.1 zeigt die von der Kamera aufgenommene Punktwolke mit einer Größe von 317.079 Punkten. Die Farben geben die Entfernung von der Kamera an. Die Person im Vordergrund in blau ist am nächsten an der Kamera, der Hintergrund in gelb-rot ist im Vergleich dazu weiter entfernt. Vor der Weiterverarbeitung wird die Anzahl der Punkte reduziert. Die nachfolgende Punktwolkenvorverarbeitung entfernt zunächst den Hintergrund. Das Ergebnis ist in Abbildung 8.2a dargestellt. Erkennbar ist hier, dass sich kleine Cluster um die Punktwolke herum befinden. Im nächsten Schritt, der in Abbildung 8.2b dargestellt ist, sind diese nicht mehr erkennbar. Die Punktwolke besteht im letzten Schritt aus 29.840 Punkten.

Urteil

Der verwendete Algorithmus eignet sich für den genannten Zweck und erfüllt alle Kriterien.

8.2.1.2 Registrierung

Da Messinstrumente und Vergleiche mit anderen ähnlichen Systemen im Rahmen dieser Masterarbeit nicht zur Verfügung stehen, sind die Ergebnisse der Registrierung nur visuell überprüfbar. Eine Aussage über die Qualität ist nicht möglich.

Anforderung

Punktwolken aus einer 360°-Perspektive werden mit dem in Abschnitt 6.2.5.2 auf Seite 60 beschriebenen Algorithmus zu einem 3D-Modell zusammengesetzt.

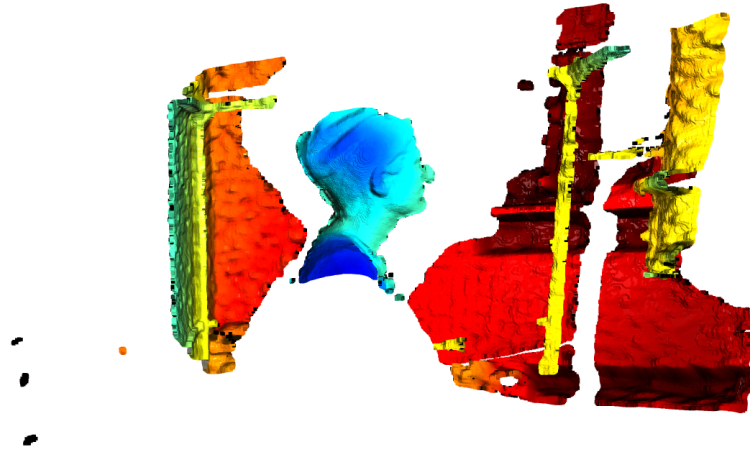


Abbildung 8.1: Punktwolke mit Hintergrund

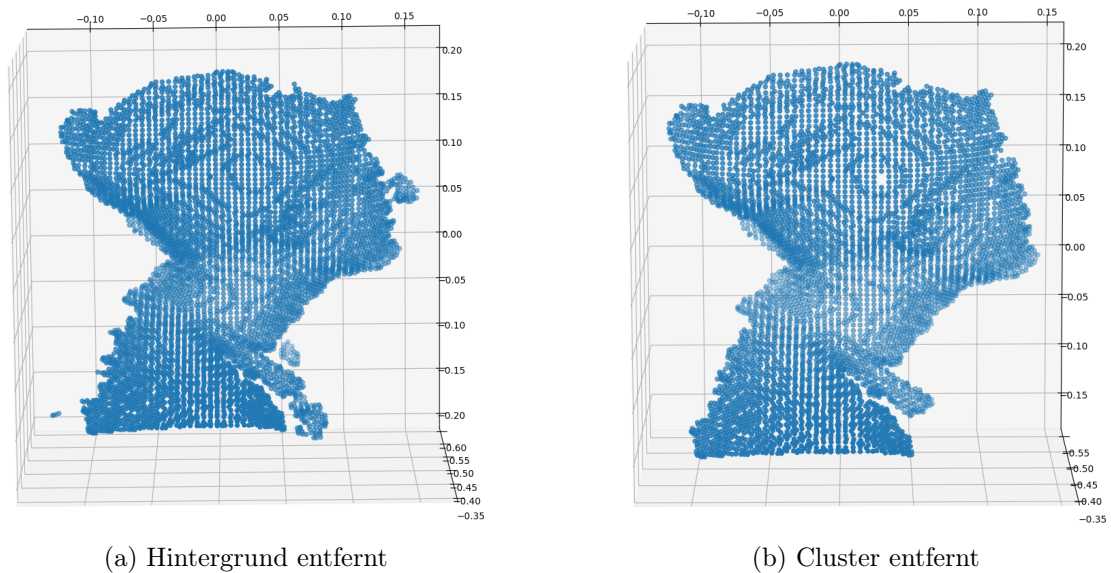


Abbildung 8.2: Hintergrund entfernen

Durchführung

Im Folgenden wird die Registrierung in den Punkten 2. und 3. mit Punktwolken getestet, die mit einer Auflösung von 848x480 aufgenommen worden sind. Die Punktwolken werden aufgrund von fehlenden Messmitteln visuell verglichen.

1. **Erbringung des Nachweises, dass die Registrierung in der Grundfunktion lauffähig ist:** Es werden zwei quaderförmige Punktwolken generiert, die mit

dem Registrierungsalgorithmus zusammengeführt werden sollen. Die Punktwolken unterscheiden sich nur in der Anzahl der Punkte. Die Abstände der Punkte ist identisch.

2. **Prüfung des Einflusses des Fitnesswerts auf das visuell überprüfbare Registrierungsergebnis:** Der zweite Test wird mit zwei Punktwolken durchgeführt die mit der Kamera aufgenommen worden sind. Es wird visuell geprüft, ob Kanten und Leerräume zwischen den Punktwolken entstehen. Dafür werden dieselben Punktwolken einmal mit einem RANSAC-Fitnesswert von $< 0,5$ und mit einem $> 0,5$ registriert. Die Punktwolken sind in einem Abstand von 45° aufgenommen worden.
3. **Einfluss des Winkels zwischen den Aufnahmen auf das visuell überprüfbare Registrierungsergebnis und den Fitnesswert:** Im dritten Test werden Punktwolken in drei Winkelabständen aufgenommen und vom Programm registriert. Die Anzahl an übereinandergelegten Punktwolken, sowie die Entwicklung der Fitnesswerte im Verlauf des Prozesses sollen beobachtet werden. Die Punktwolken werden im Winkel von etwa

- a) 90°
- b) 45°
- c) 20°

aufgenommen. Es wird überprüft, ob folgende Annahme zutrifft:

Die Höhe des erreichten Fitnesswerts richtet sich nach dem relativen Anteil des Überschneidungsbereichs an der Größe der Zielpunktwolke. Es wird angenommen, dass bei Punktwolken, die in einem größeren Winkelabstand aufgenommen werden ein niedrigerer Fitnesswert erreicht wird. Da die Größen der Punktwolken nicht einheitlich sind, bedeutet niedriger in diesem Zusammenhang eine Differenz im zehntel Bereich.

Kriterien

1. Die Punktwolken werden an den Überschneidungsbereichen zusammengesetzt.
2. Der Versatz zwischen den Punktwolken ist gering, sodass eine gleichmäßige Oberfläche entsteht.
3. *Nur für Test 3 gilt:* Es entsteht ein geschlossenes 3D-Modell eines Kopfes.

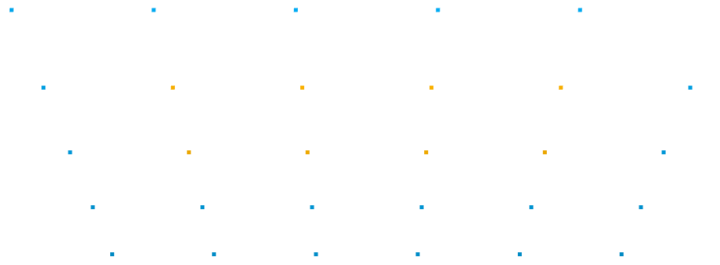


Abbildung 8.3: Test 1

Beobachtung

1. Test: Die kleinere Punktwolke wird mit der Größeren zusammengeführt, siehe Abbildung 8.3. In blau ist die Quellpunktwolke und in gelb die Zielpunktwolke dargestellt. Bei acht Überschneidungspunkten und einer Zielpunktwolke mit acht Punkten ergibt sich ein Fitnesswert von 1, also eine vollständige Überschneidung der Punktwolken.
2. Test: Bei einem RANSAC-Fitnesswert von 0,29 und eine ICP-Fitnesswert von 0,14 entstehen Spalten zwischen der Quell- und Zielpunktwolke, siehe Abbildung 8.4b. Dieser Spalt ist nicht an allen Überschneidungspunkten gleichermaßen ausgeprägt, wie zum Beispiel an der Seite des Kopfes, siehe 8.5c. Liegt der RANSAC-Fitnesswert bei 0,74 und der ICP-Fitnesswert bei 0,24 entsteht kein Spalt in der Ausprägung wie beim ersten Bild, siehe Abbildung 8.7b. Wie in Abbildung 8.4a am Kinn erkennbar gibt es vereinzelt Stellen an denen die Punktwolken nicht aufeinanderliegen.
3. Test:
 - a) Die ersten beiden Punktwolken werden nach sechs Versuchen mit einem RANSAC-Fitnesswert von 0,83 und einem ICP-Fitnesswert von 0,35 registriert. Die Kanten der Punktwolken enthalten Ausreißercluster die nach oben gebogen sind und kein Körperteil abbilden, wie an der Nase und dem Auge in Abbildung 8.6a erkennbar. Die Punktwolken liegen eng aufeinander und es sind in Abbildung 8.6b und 8.6c kaum Kanten und kein Versatz erkennbar. Der Prozess bricht nach zwei Punktwolken ab.
 - b) Die ersten beiden Punktwolken werden nach vier Versuchen mit einem RANSAC-Fitnesswert von 0,73 und einem ICP-Fitnesswert von 0,27 registriert. Endet

eine Punktwolke auf der Fläche der anderen Punktwolke, entstehen Kanten an denen die endende Punktwolken nach oben gebogen ist, siehe Abbildung 8.7. Mit der dritten Punktwolke konnten keine Übereinstimmungen gefunden werden, sodass die Registrierung 15-mal ergebnislos mit einem RANSAC-Fitnesswert von 0,0 wiederholt und dann abgebrochen wurde. Zudem wurde beobachtet, dass die Registrierung erschwert wird, wenn die Testperson lange Haare oder einen Zopf hat, da abstehende Haarsträhnen zusätzliches Rauschen verursachen können.

- c) Der Prozess bricht nach 15 erfolglosen Registrierungsversuchen mit einem RANSAC-Fitnesswert von 0,0 ab.

Die Annahme, dass bei Punktwolken die in einem geringeren Winkelabstand aufgenommen worden sind ein größerer Fitnesswert erreicht wird, als bei Punktwolken mit einem größeren Winkelabstand kann mit den vorliegenden zwei Tests bestätigt werden.

Urteil

Die Kriterien eins und zwei werden erfüllt. Da nicht mehr als zwei Punktwolken registriert werden können, wird das dritte Kriterium nicht erfüllt. Anforderung SW 8 wird nicht erfüllt.

8.2.1.3 Oberflächenrekonstruktion

Anforderung

Rekonstruktion einer Oberfläche

Durchführung

Es wird der Ball-Pivoting-Algorithmus aus Abschnitt 6.2.5.3 mit einem Level of Detail von 1.000.000 angewendet. Es wird ein Modell aus zwei registrierten Punktwolken verwendet, die mit einer Auflösung von 848x480 aufgenommen worden sind. Die Überprüfung erfolgt visuell.

Kriterien

1. Zwischen den Punkten wird eine Oberfläche rekonstruiert
2. Glatte und kantenfreie Oberfläche

3. Geschlossene Oberfläche
4. Details im Gesicht wie Augen, Mund, Nase und Ohren sind erkennbar

Beobachtung

Die Oberfläche enthält viele offene Stellen, insbesondere an der im vorherigen Test beschriebenen Kante wird keine Oberfläche rekonstruiert. Betrachtet man vergleichend die zwei übereinandergelegten Ausgangspunktwolken, ist dort eine geschlossene Oberfläche aus Punkten erkennbar, siehe Abbildung 8.8. An Stellen mit rekonstruierter Oberfläche sind die Dreiecke erkennbar, wodurch eine kantige Oberfläche entsteht. Details im Gesicht sind rudimentär erkennbar. Dabei ist zu beachten, dass die Ausgangspunktwolke diese Details ebenfalls nur rudimentär abbildet.

Urteil

Die Kriterien werden teilweise erfüllt. Insbesondere das letzte Kriterium ist von der Detailtiefe der Punktwolke abhängig.

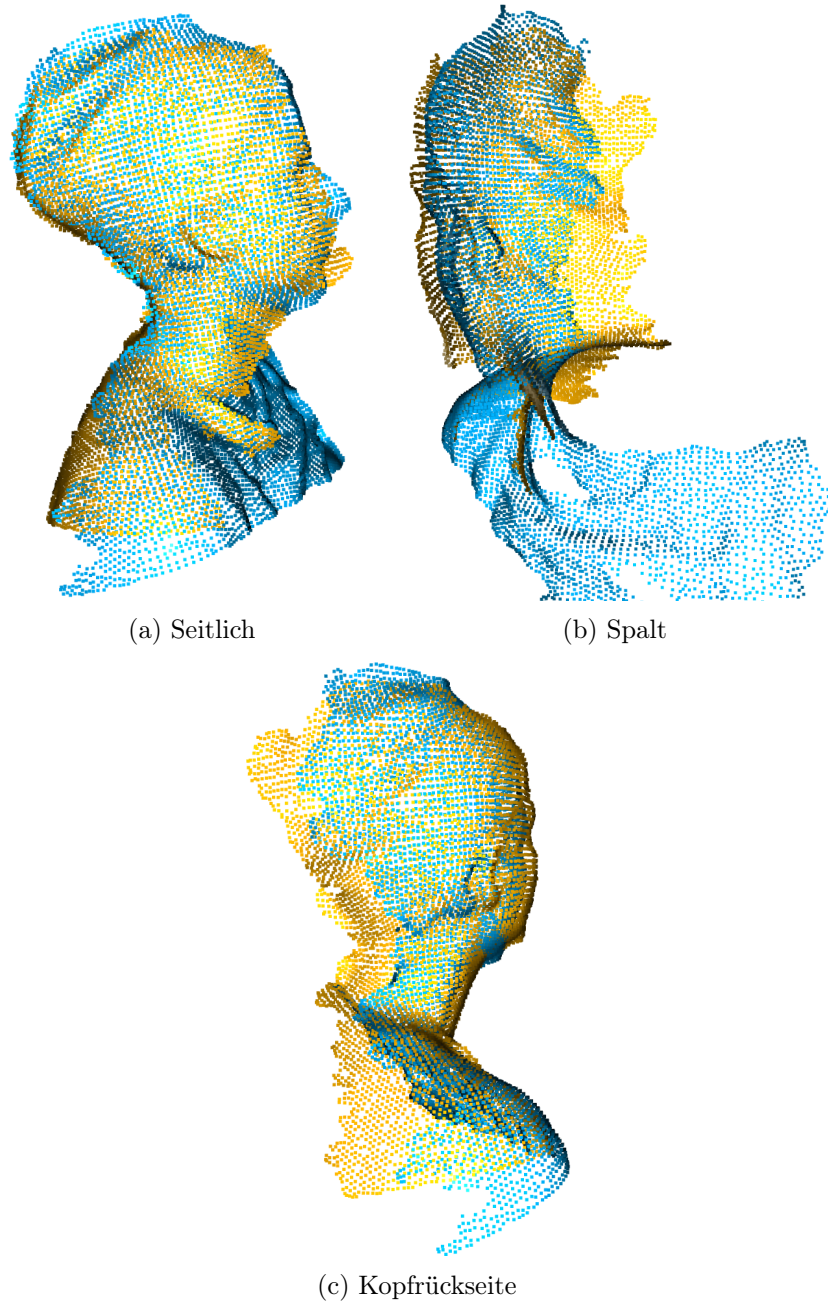


Abbildung 8.4: Registrierung (Fitnesswert RANSAC: 0,29 und ICP: 0,14)

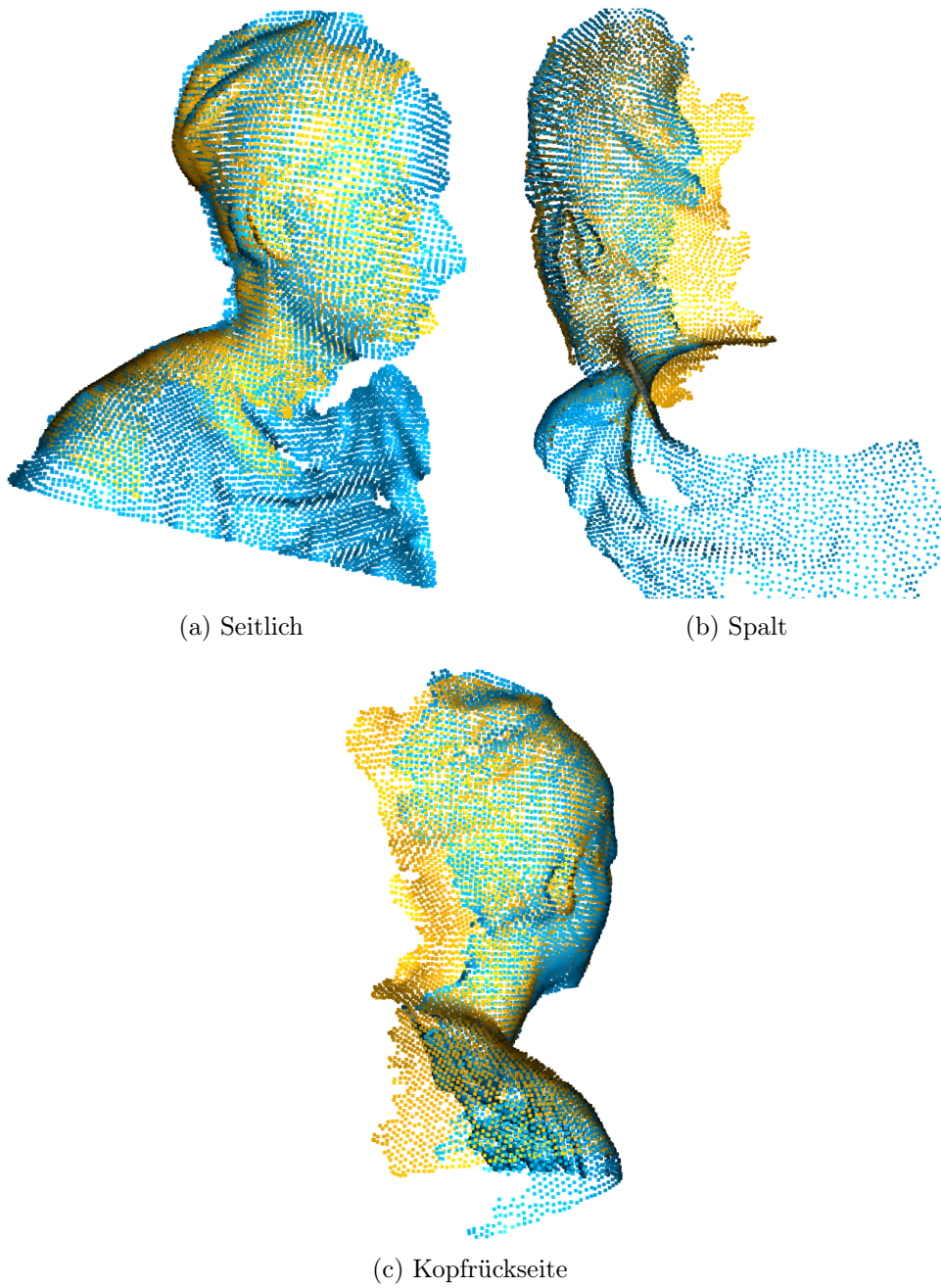


Abbildung 8.5: Registrierung (Fitnesswert RANSAC: 0,74 und ICP: 0,24)

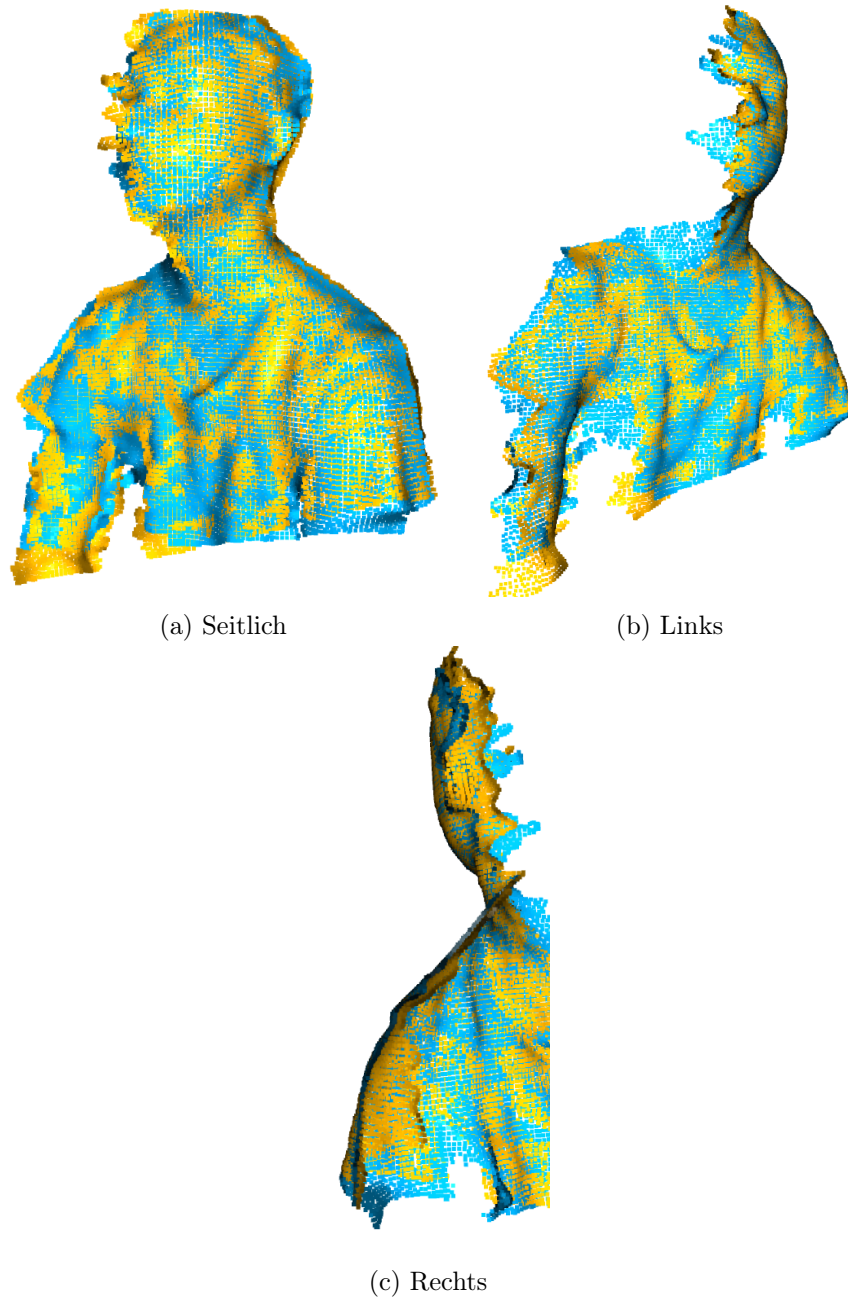


Abbildung 8.6: Test 3a: 20° (Fitnesswert RANSAC: 0,83 und ICP: 0,35)

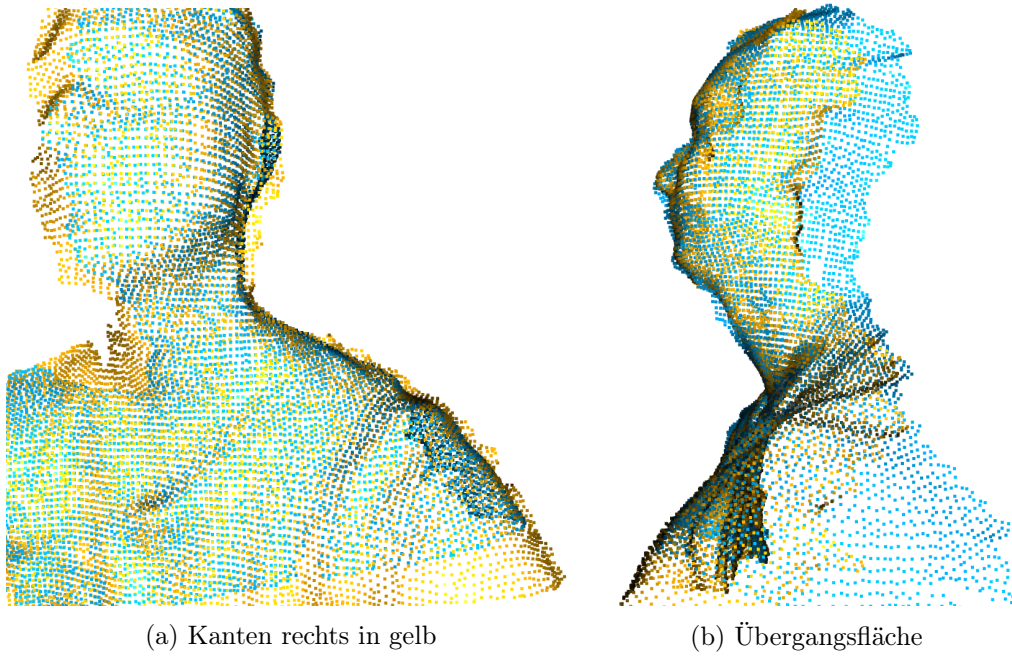


Abbildung 8.7: Test 3b: 45° (Fitnesswert RANSAC: 0,73 und ICP: 0,27)

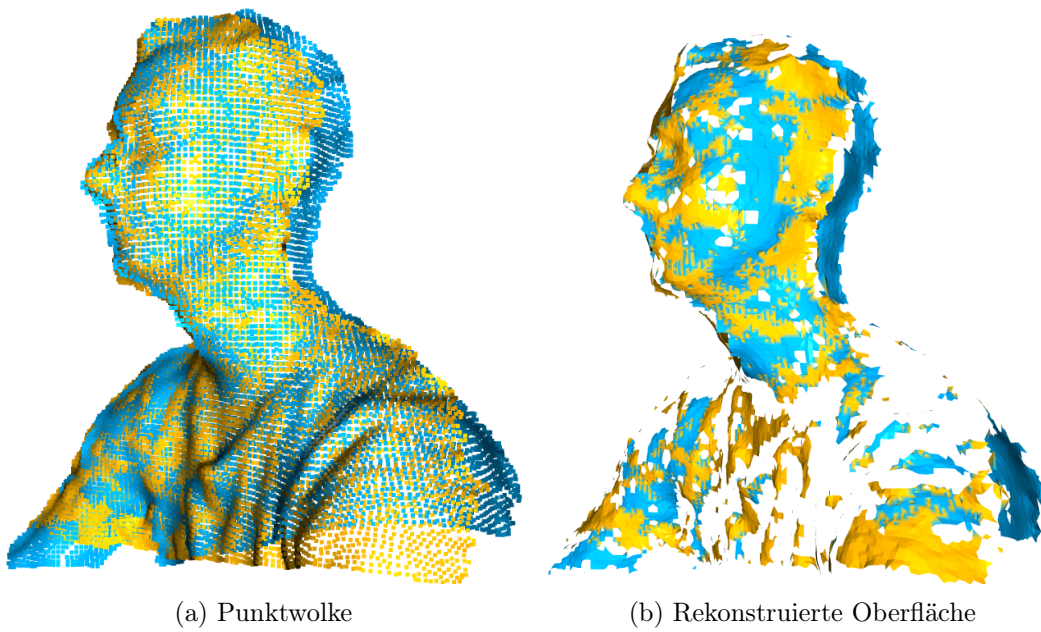


Abbildung 8.8: Oberflächenrekonstruktion

9 Fazit und Ausblick

Im Rahmen dieser Arbeit wurde ein System zur Erstellung von 3D-Modellen von Köpfen von Personen entwickelt. Dafür ist ein funktionsfähiger Prototyp des Gesamtsystems entwickelt worden, der sowohl die Bildaufnahme als auch die Bildverarbeitung bis zur Erstellung des 3D-Modells realisiert und hard- und softwaretechnisch umgesetzt worden ist. Entscheidend für den Erfolg des Projekts war die Definition des Projektziels sowie der Anforderungen zu Beginn. Diese Herangehensweise ermöglichte einen Überblick und eine Abgrenzung der zu bearbeitenden Themenbereiche.

Um das initiale Projektziel zu erreichen und ein vollständiges 3D-Modell zu berechnen, sind Änderungen der vorhandenen Umsetzung in den Bereichen Punktwolkenvorverarbeitung, Registrierung und Oberflächenrekonstruktion notwendig. So ist die Registrierung grundsätzlich funktionsfähig wie die Tests im vorangegangenen Kapitel zeigen, brechen jedoch einheitlich nach zwei registrierten Punktwolken ab, ohne das Überschneidungsbereiche gefunden werden können. Ursächlich sind hier besonders die unsauberen und gebogenen Kanten der Punktwolken. Bei der Oberflächenrekonstruktion entstehen an diesen Stellen zusätzlich Löcher. Für die Erkennung und Glättung von Randpunkten können Algorithmen, wie sie in [39] vorgestellt werden, verwendet werden.

Um Ausführungszeiten im gesamten Programm zu beschleunigen können einerseits Hardwareprodukte wie Embedded Computerboards, zum Beispiel die der Nvidia Jetson-Serie, verwendet werden oder andererseits Änderungen in der Software umgesetzt werden. Gerade die Bildaufnahme und die Entfernung von Rauschen und Hintergrund haben lange Ausführungszeiten. Ein erster Schritt wäre hier die Parallelisierung von Prozessen. Außerdem wird jede aufgenommene Punktwolke im Aufnahmeprozess als Datei gespeichert bevor eine neue Punktwolke aufgenommen werden kann, da die Speicherung des Frames im Open3D-Punktwolkenformat oder als numpy-Array während der Aufnahme im Rahmen der Arbeit nicht umgesetzt werden konnte. Der Aufnahmeprozess kann zusätzlich durch die Verwendung von zwei oder mehr Kameras beschleunigt werden.

Die Hardwarekonstruktion hat sich während des praktischen Einsatzes als instabil und der Motor als zu leistungsschwach herausgestellt. Für zukünftige Projekte wird insbesondere

eine Stabilisierung der Motorachse und des U-Rahmens sowie eine Neudimensionierung des Schrittmotors empfohlen. Mit der vorhandenen Konstruktion sind Aufnahmen nur im manuellen Betrieb ohne den Schrittmotor zuverlässig möglich.

Weiterhin kann die Bedienoberfläche um weitere Funktionen wie die Eingabe von Ordnernamen für die Punktwolken oder Einstellung von Kameraoptionen wie beispielsweise die Auflösung erweitert werden.

Der entwickelte 3D-Scanner erfüllt die Anforderung an die Grundfunktionen, ist dabei jedoch stark abhängig von äußeren Einstellungen. Ist die Kamera im Aufnahmebereich nicht auf die Person eingestellt, werden Teile der Konstruktion erfasst und registriert oder der Kopf ist nicht vollständig auf der Aufnahme abgebildet. Sind diese Parameter passend eingestellt, erstellt das System zuverlässig ein 3D-Modell aus zwei Punktwolken mit einer Oberfläche.

In zukünftigen Projekten kann der 3D-Scanner zu einem Ganzkörperscanner für Personen erweitert werden. Es ist weiterhin denkbar neben Personen auch Objekte zu scannen und zu vermessen. So wäre ein Einsatz in der Instandhaltung für schwer oder nicht mehr beschaffbare Bauteile möglich. Außerdem ist der Scanner für die Anwendung ohne Hardwarekonstruktion nutzbar, wenn er im Abstand von etwa 50 cm zum Objekt oder zur Person gehalten wird und für die Bildaufnahme um die Person bewegt wird. Damit ist der Scanner auch für mobilere Anwendungen verwendbar.

Literaturverzeichnis

- [1] ALBON, Chris: *Machine learning with Python cookbook: Practical solutions from preprocessing to deep learning*. First edition. Beijing and Boston and Farnham and Sebastopol and Tokyo : O'Reilly, 2018. – ISBN 9781491989388
- [2] BERNARDINI, F. ; MITTLEMAN, J. ; RUSHMEIER, H. ; SILVA, C. ; TAUBIN, G.: The ball-pivoting algorithm for surface reconstruction. In: *IEEE Transactions on Visualization and Computer Graphics* 5 (1999), Nr. 4, S. 349–359
- [3] BEYERER, Jürgen ; PUENTE LEÓN, Fernando ; FRESE, Christian ; MEYER, Johannes: *Automatische Sichtprüfung: Grundlagen, Methoden und Praxis der Bildgewinnung und Bildauswertung*. 2. Aufl. 2016. Berlin, Heidelberg : Springer Berlin Heidelberg, 2016. – ISBN 9783662477861
- [4] BIBER, P. ; STRASSER, W.: The normal distributions transform: a new approach to laser scan matching. In: *Proceedings 2003 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2003) (Cat. No.03CH37453)* Bd. 3, 2003, S. 2743–2748 vol.3
- [5] BRAUN, Philipp ; SLIWINSKI, Martin ; HINCKELDEYN, Johannes ; KREUTZFELDT, Jochen: Challenges of CAD conversion to 3D development environments with respect to kinematic dependencies. In: *Proceedings of The 61st SIMS Conference on Simulation and Modelling SIMS 2020, September 22-24, Virtual Conference, Finland*, Linköping University Electronic Press, 2021 (Linköping Electronic Conference Proceedings), S. 215–221
- [6] BUNDESANSTALT FÜR ARBEITSSCHUTZ UND ARBEITSMEDIZIN (Hrsg.): *Damit nichts ins Auge geht ... Schutz vor Laserstrahlung*. 2. 2010. – ISBN 978-3-88261-678-1
- [7] CHEN, Y. ; MEDIONI, G.: Object modeling by registration of multiple range images. In: *Proceedings. 1991 IEEE International Conference on Robotics and Automation*, 1991, S. 2724–2729 vol.3

- [8] CLAUSEN, Jens: *Branch and Bound Algorithms - Principles and Examples*. – URL https://janders.eecg.utoronto.ca/1387/readings/b_and_b.pdf. – Zugriffsdatum: 07.07.2022
- [9] CONRAD ELECTRONIC SE: *Conrad Electronic SE*. – URL <https://www.conrad.de/>. – Zugriffsdatum: 24.08.2022
- [10] DEVELOPERS scikit-learn: *Clustering*. 2007-2022. – URL <https://scikit-learn.org/stable/modules/clustering.html#k-means>. – Zugriffsdatum: 24.06.2022
- [11] DIN DEUTSCHES INSTITUT FÜR NORMUNG E.V: *Projektmanagement – Projektmanagementsysteme – Teil 5: Begriffe*
- [12] DIN-NORMENAUSSCHUSS ERGONOMIE: DIN 33402-2, Ergonomie – Körpermaße des Menschen – Teil 2: Werte.
- [13] DITTMAR, Malte: *Kollisionsvermeidung eines mobilen Roboters mittels Schätzung der Bewegungsrichtung und Geschwindigkeit von Objekten aus 3D-Bilddaten*, HAW Hamburg, Masterarbeit, 2022
- [14] DOLD MECHATRONIK: *DOLD Mechatronik - Alles für Maker, Ingenieure und Visionäre*. – URL <https://www.dold-mechatronik.de/>. – Zugriffsdatum: 24.08.2022
- [15] EUROPÄISCHEN UNION: *Richtlinie 2009/48/EG des Europäischen Parlaments und des Rates vom 18. Juni 2009 über die Sicherheit von Spielzeug*. 2009
- [16] FISCHLER, Martin A. ; BOLLES, Robert C.: Random Sample Consensus: A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography. In: *Commun. ACM* 24 (1981), jun, Nr. 6, S. 381–395
- [17] GELFAND, N. ; MITRA, N. J. ; GUIBAS, L. J. ; POTTMANN, H.: Robust Global Registration. In: *Symposium on Geometry Processing*, 2005, S. 197–206
- [18] GIANCOLA, Silvio ; VALENTI, Matteo ; SALA, Remo: *A Survey on 3D Cameras: Metrological Comparison of Time-of-Flight, Structured-Light and Active Stereoscopy Technologies*. Cham : Springer International Publishing, 2018 (SpringerBriefs in Computer Science). – URL <http://dx.doi.org/10.1007/978-3-319-91761-0>. – ISBN 9783319917603

- [19] GITHUB: *PYPL PopularitY of Programming Language*. – URL <https://pypl.github.io/PYPL.html?country=DE>. – Zugriffsdatum: 07.07.2022
- [20] HARTLEY, Richard ; ZISSERMAN, Andrew: *Multiple View Geometry in Computer Vision*. 2. Cambridge University Press, 2004
- [21] HARTWIG, Matthias ; LL.M., Bénédicte Martin ; SCHUMACHER, Oskar: Rechtliche Rahmenbedingungen für den Einsatz von autonomen Robotern in Assistenzfunktionen: Studie im Rahmen des BMBF-Begleitprojekts ARAIG Teilvorhaben „Ethische und rechtliche Aspekte der Service-Robotik“. (2020)
- [22] HENSEL, Marc: Informationen zu Abschlussarbeiten Formalitäten, Aufbau, Hinweise: Survival Pack für Bachelor- und Masterarbeiten / HAW Hamburg. 18.08.2022. – Information
- [23] INTEL® REALSENSE™ : *LiDAR Camera L515 - Datasheet*. – URL <https://docs.rs-online.com/f31c/A700000006942953.pdf>. – Zugriffsdatum: 05.07.2022
- [24] INTEL® REALSENSE™ : *Multi-Camera configurations - D400 Series Stereo Cameras*. – URL <https://dev.intelrealsense.com/docs/multiple-depth-cameras-configuration>. – Zugriffsdatum: 06.07.2022
- [25] INTEL® REALSENSE™ : *Multi-Camera configurations with the Intel® RealSense™ LiDAR Camera L515*. – URL <https://dev.intelrealsense.com/docs/lidar-camera-l515-multi-camera-setup>. – Zugriffsdatum: 06.07.2022
- [26] INTEL® REALSENSE™ : *Product Family D400 Series - Datasheet*. 010. – <https://web.archive.org/web/20210906063210/https://www.intelrealsense.com/download/15409/?ga=2.49108512.701557908.1630909942-1916947096.1630909933> – *Zugriffsdatum* : 20.08.2022
- [27] INTEL® REALSENSE™ : *SDK 2.0 (v2.50.0)*. – URL <https://github.com/IntelRealSense/librealsense/releases>. – Zugriffsdatum: 06.07.2022
- [28] INTEL® REALSENSE™ DEVELOPER DOCUMENTATION: *Tuning depth cameras for best performance*. 2022-03-31T15:19:03.000Z. – URL <https://dev.intelrealsense.com/docs/tuning-depth-cameras-for-best-performance>. – Zugriffsdatum: 31.03.2021

- [29] JAUER, Philipp: *Physikalisch motivierte Punktwolkenregistrierung - Entwurf und Evaluierung eines neuen Konzepts zur effizienten Registrierung von Punktwolken*, Universität zu Lübeck, Dissertation
- [30] JOY-IT: *NEMA17-05GM*. – URL <https://asset.conrad.com/media10/add/160267/c1/-/de/002148836DS02/datenblatt-2148836-joy-it-schrittmotor-nema17-05gm-joy-it-168-nm-168-a-wellen-durchmesser-8-mm.pdf>. – Zugriffsdatum: 16.08.2022
- [31] KHAN, Kamran ; REHMAN, Saif U. ; AZIZ, Kamran ; FONG, Simon ; SARASVADY, S.: DBSCAN: Past, present and future. In: *The Fifth International Conference on the Applications of Digital Information and Web Technologies (ICADIWT 2014)*, 2014, S. 232–238
- [32] LIU, Shan ; ZHANG, Min ; KADAM, Pranav ; KUO, C.-C. J.: *3D Point Cloud Analysis: Traditional, Deep Learning, and Explainable Machine Learning Methods*. 1st ed. 2021. Cham : Springer International Publishing and Imprint Springer, 2021 (Springer eBook Collection). – ISBN 9783030891800
- [33] LOW, Kok-Lim: *Linear Least-Squares Optimization for Point-to-Plane ICP Surface Registration*. Chapel Hill, University of North Carolina, 2004
- [34] MAGALHAES, Wanderson: *Python PySide2 Custom Title Bar [Modern GUI]*. – URL https://github.com/Wanderson-Magalhaes/Python_PySide2_Custom_Title_Bar. – Zugriffsdatum: 01.08.2022
- [35] MAGNUSSON, Martin ; NUCHTER, Andreas ; LORKEN, Christopher ; LILIENTHAL, Achim J. ; HERTZBERG, Joachim: Evaluation of 3D registration reliability and speed - A comparison of ICP and NDT. In: *2009 IEEE International Conference on Robotics and Automation*, 2009, S. 3907–3912
- [36] MICROSOFT: *EU Declaration of Conformity*. – URL <https://query.prod.cms.rt.microsoft.com/cms/api/am/binary/RE47eL4>. – Zugriffsdatum: 06.07.2022
- [37] MICROSOFT: *Synchronize multiple Azure Kinect DK devices*. – URL <https://docs.microsoft.com/en-us/azure/kinect-dk/multi-camera-sync>. – Zugriffsdatum: 05.07.2022

- [38] MICROSOFT: *Azure Kinect DK hardware specifications*. 05.07.2022. – URL <https://docs.microsoft.com/en-us/azure/kinect-dk/hardware-specification>. – Zugriffsdatum: 05.07.2022
- [39] MINEO, Carmelo ; PIERCE, Stephen G. ; SUMMAN, Rahul: Novel algorithms for 3D surface point cloud boundary detection and edge reconstruction. In: *Journal of Computational Design and Engineering* 6 (2018), Nr. 1, S. 81–91. – ISSN 2288-5048
- [40] NEXR TECHNOLOGIES SE: *HM und NEXR starten mit der ersten virtuellen Umkleidekabine in die Zukunft*. – URL <https://www.nexr-technologies.com/de/virtual-fitting-hm-success-case/>. – Zugriffsdatum: 04.08.2022
- [41] NEXR TECHNOLOGIES SE: *NeXR Technologies SE: GOLD's GYM setzt NeXR-Scanning-Technologie zur Visualisierung von Trainingsfortschritten ein*. – URL <https://www.dgap.de/dgap/News/corporate/nexr-technologies-golds-gym-setzt-nexrscanningtechnologie-zur-visualisierung-von-trainingsfortschritten-ein/?newsID=1421968>. – Zugriffsdatum: 05.08.2022
- [42] NEXR TECHNOLOGIES SE: *ONPOINT MOTION CAPTURE STUDIO*. – URL <https://www.nexr-technologies.com/de/motion-capture-service/>. – Zugriffsdatum: 04.08.2022
- [43] NEXR TECHNOLOGIES SE: *Portabler 3D-Körperscanner*. – URL <https://www.nexr-technologies.com/de/scanning-systems/>. – Zugriffsdatum: 07.09.2022
- [44] NEXR TECHNOLOGIES SE: *Virtual Fitting | HM Stores | 3D Fullbody Scan by NeXR Technologies*. – URL <https://www.youtube.com/watch?v=cCcgMRRzsM>. – Zugriffsdatum: 04.08.2022
- [45] OPEN3D: *File IO*. – URL http://www.open3d.org/docs/0.11.0/tutorial/geometry/file_io.html. – Zugriffsdatum: 02.08.2022
- [46] OPEN3D: *Global registration*. – URL http://www.open3d.org/docs/0.11.0/tutorial/pipelines/global_registration.html. – Zugriffsdatum: 01.08.2022
- [47] OPEN3D: *ICP Registration*. – URL http://www.open3d.org/html/tutorial/Basic/icp_registration.html. – Zugriffsdatum: 28.07.2022

- [48] OPEN3D: *Registration.cpp*. – URL <https://github.com/isl-org/Open3D/blob/v0.11.0/cpp/open3d/pipelines/registration/Registration.cpp>. – Zugriffsdatum: 27.08.2022
- [49] OUSTER INC.: *Ouster - Datasheet OS2*. 2022. – URL <https://data.ouster.io/downloads/datasheets/datasheet-rev06-v2p3-os2.pdf>. – Zugriffsdatum: 16.08.2022
- [50] POHL, Klaus ; RUPP, Chris: *Basiswissen Requirements Engineering: Aus- und Weiterbildung nach IREB-Standard zum Certified Professional for Requirements Engineering Foundation Level*. 5., überarbeitete und aktualisierte Auflage. Heidelberg : dpunkt.verlag, 2021. – URL <https://search.ebscohost.com/login.aspx?direct=true&scope=site&db=nlebk&db=nlabk&AN=2913478>. – ISBN 9783969102473
- [51] POUX, Florent: *5-Step Guide to generate 3D meshes from point clouds with Python*. – URL <https://towardsdatascience.com/5-step-guide-to-generate-3d-meshes-from-point-clouds-with-python-36bad397d8ba>. – Zugriffsdatum: 30.07.2022
- [52] RAGURAM, Rahul ; FRAHM, Jan-Michael ; POLLEFEYS, Marc: A Comparative Analysis of RANSAC Techniques Leading to Adaptive Real-Time Random Sample Consensus. In: FORSYTH, David (Hrsg.) ; TORR, Philip (Hrsg.) ; ZISSERMAN, Andrew (Hrsg.): *Computer Vision – ECCV 2008*. Berlin, Heidelberg : Springer Berlin Heidelberg, 2008, S. 500–513. – ISBN 978-3-540-88688-4
- [53] RAO R, Madhusudhan ; RADHAKRISHNA, Deepa ; S, Usha: Development of a Robot-mounted 3D Scanner and Multi-view Registration Techniques for Industrial Applications. In: *Procedia Computer Science* 133 (2018), S. 256–267. – International Conference on Robotics and Smart Manufacturing (RoSMa2018). – ISSN 1877-0509
- [54] RAPPONOTTI, B. ; SNOWDEN, M. ; ZENG, A.: *Point Cloud to Mesh, Ball-Pivoting Algorithm*. – URL <https://cs184team.github.io/cs184-final/writeup.html>. – Zugriffsdatum: 03.08.2022
- [55] REICHELTELEKTRONIK GMBH Co. KG: *ESP8266 NodeMCU ESP8266 WiFi-Modul*. – URL <https://www.reichelt.de/nodemcu-esp8266-wifi-modul-debo-jt-esp8266-p219900.html>. – Zugriffsdatum: 19.08.2022

- [56] RUSINKIEWICZ, S. ; LEVOY, M.: Efficient variants of the ICP algorithm. In: *Proceedings Third International Conference on 3-D Digital Imaging and Modeling*, 2001, S. 145–152
- [57] RUSU, Radu B.: *Semantic 3D Object Maps for Everyday Manipulation in Human Living Environments*, Technischen Universität München, Dissertation
- [58] RUSU, Radu B.: *Springer Tracts in Advanced Robotics Ser. Bd. v.85: Semantic 3D Object Maps for Everyday Robot Manipulation*. Berlin, Heidelberg : Springer Berlin / Heidelberg, 2013. – ISBN 9783642354793
- [59] RUSU, Radu B. ; BLODOW, Nico ; BEETZ, Michael: Fast Point Feature Histograms (FPFH) for 3D registration. In: *2009 IEEE International Conference on Robotics and Automation*, 2009, S. 3212–3217
- [60] STATISTISCHES BUNDESAMT: *Körpermaße nach Altersgruppen und Geschlecht*. 2019-03-21T15:54:25+0100. – URL <https://www.destatis.de/DE/Themen/Gesellschaft-Umwelt/Gesundheit/Gesundheitszustand-Relevantes-Verhalten/Tabellen/liste-koerpermasse.html;jsessionid=72AC9078907B178E116DA84DC2358B03.live722>. – Zugriffsdatum: 2022-03-21T10:42:54
- [61] STEREO LABS: *Get Started with ZED*. – URL <https://www.stereolabs.com/docs/get-started-with-zed/>. – Zugriffsdatum: 06.07.2022
- [62] STEREO LABS: *Setting Up Multiple 3D Cameras*. – URL <https://www.stereolabs.com/docs/video/multi-camera/>. – Zugriffsdatum: 06.07.2022
- [63] STEREO LABS: *ZED 2 Camera and SDK Overview*. – URL <https://cdn.stereolabs.com/assets/datasheets/zed2-camera-datasheet.pdf>. – Zugriffsdatum: 06.07.2022
- [64] STORE, Microsoft: *Azure Kinect DK*. – URL <https://www.microsoft.com/de-de/d/azure-kinect-dk/8pp5vxmd9nhq?activetab=pivot:overviewtab>. – Zugriffsdatum: 06.07.2022
- [65] SZELISKI, Richard: *Computer Vision: Algorithms and Applications*. London : Springer London, 2011 (SpringerLink Bücher). – ISBN 978-1-84882-935-0
- [66] THE QT COMPANY: *Qt for Open Source Development*. – URL <https://www.qt.io/download-open-source>. – Zugriffsdatum: 15.07.2022

- [67] THE QT COMPANY: *Using Qt for Educational Programs*. – URL <https://www.qt.io/pricing/qt-for-educational-program>. – Zugriffsdatum: 15.07.2022
- [68] ULAŞ, Cihan ; TEMELTAŞ, Hakan: 3D Multi-Layered Normal Distribution Transform for Fast and Long Range Scan Matching. In: *Journal of Intelligent & Robotic Systems* 71 (2013), Nr. 1, S. 85–108. – ISSN 1573-0409
- [69] WEISS, Alexander ; HIRSHBERG, David ; BLACK, Michael J.: Home 3D body scans from noisy image and range data. In: *2011 International Conference on Computer Vision*, 2011, S. 1951–1958
- [70] WWW.OPENPLATFORM.CC: *User Manual For L298N Motor Driver Board (ST1112)*. – URL <https://asset.conrad.com/media10/add/160267/c1/-/en/001525437ML01/bedienungsanleitung-1525437-iduino-st-1112-motortreiber-1-st.pdf>. – Zugriffsdatum: 16.08.2022
- [71] YEMM, Graham: *The Financial Times essential guide to leading your team: How to set goals, measure performance and reward talent*. Harlow, England : Pearson, 2012 (Financial Times essential guides). – ISBN 9780273772446
- [72] ZHOU, Qian-Yi ; PARK, Jaesik ; KOLTUN, Vladlen: Fast Global Registration. In: LEIBE, Bastian (Hrsg.) ; MATAS, Jiri (Hrsg.) ; SEBE, Nicu (Hrsg.) ; WELLING, Max (Hrsg.): *Computer Vision – ECCV 2016*. Cham : Springer International Publishing, 2016, S. 766–782. – ISBN 978-3-319-46475-6

A Anhang

Die Anhänge sind auf einer CD in elektronischer Form abgelegt und können bei Prof. Dr.-Ing. Marc Hensel oder Prof. Dr. rer. nat. Ulrike Herster eingesehen werden.

Anhang A1: Masterthesis als PDF-Datei

Anhang A2: Quellcode des Programms mit Beispielpunktwolken

Anhang A3: Videos von der Programmausführung

Anhang A4: CAD-Modelle der Kamerahalterung

Erklärung zur selbstständigen Bearbeitung einer Abschlussarbeit

Hiermit versichere ich, dass ich die vorliegende Arbeit ohne fremde Hilfe selbständig verfasst und nur die angegebenen Hilfsmittel benutzt habe. Wörtlich oder dem Sinn nach aus anderen Werken entnommene Stellen sind unter Angabe der Quellen kenntlich gemacht.

Ort

Datum

Unterschrift im Original