

Bearbeitungsbeginn: 01.09.2021

Vorgelegt am: 31.03.2022

# Thesis

zur Erlangung des Grades

**Master of Science**

im Studiengang Medieninformatik

an der Fakultät Digitale Medien

***Ron Marvin Metzger***

***Matrikelnummer: 265018***

## VRonGlove

Konzeption, Umsetzung und Implementierung eines Open-Source  
Motion Capture Handschuhs mit haptischem Feedback für VR und  
Motion Capture

Erstbetreuer:	Prof. Christoph Müller
Zweitbetreuer:	David Lochmann



## Abstract

Durch Motion Capture kann man die Bewegungen eines Menschen in Echtzeit auf einen 3D-Charakter übertragen. Da Motion Capture der Hände durch VR-Anwendungen inzwischen für Konsumenten in Frage kommt, erhöht sich die Nachfrage nach preiswerten Lösungen für das Handtracking. Lösungen hierzu sind für Konsumenten jedoch noch zu teuer. Für Unternehmen werden speziell für VR bereits Lösungen mit haptischem Feedback, mit der die Immersion in VR-Anwendungen erhöht werden kann, angeboten. In dieser Arbeit wird deshalb untersucht, ob ein günstiger Motion Capture-Handschuh mit haptischem Feedback erstellt werden kann, der als Alternative zu teuren Systemen gesehen wird. Dazu werden Lösungen für Motion Capture und haptischem Feedback untersucht und verglichen. Anschließend wird der Designprozess eines Handschuhs präsentiert, der Motion Capture der Finger sowie haptisches Feedback miteinander vereint. Dieser wird in der Spiele-Engine Unreal Engine 4 implementiert, wodurch sich die Finger eines 3D-Charakters anhand der Messdaten der Sensoren auf dem Handschuh bewegt und Daten für das Einschränken von Bewegungen durch Motoren berechnet. Das Design wird anschließend durch eine qualitative Umfrage evaluiert.

Motion capture can be used to transfer a human's movements to a 3D character in real time. As motion capture of hands through VR applications has become an option for consumers, the demand for low-cost solutions for hands is increasing. However, solutions for this are still too expensive for consumers. For companies, solutions with haptic feedback, which can increase the immersion in VR applications, are already offered especially for VR. This thesis therefore investigates whether a low-cost motion capture glove with haptic feedback can be created that is seen as an alternative to expensive systems. For this purpose, motion capture and haptic feedback solutions are investigated and compared. The design process of a glove that combines motion capture of the fingers and haptic feedback is then presented. The glove then gets implemented in a game engine called Unreal Engine 4, allowing the fingers of a 3D character to move based on the glove's sensor data and additionally calculating data for constraining finger movement using motors. The design will then be evaluated through a qualitative survey.



## Gender-Hinweis

Aus Gründen der besseren Lesbarkeit wird in dieser Arbeit das generische Maskulinum verwendet. Dieses ist jedoch nicht nur auf männliche sondern auch auf weibliche und andere Geschlechteridentitäten bezogen, sofern dies für die entsprechende Aussage erforderlich ist.



## Inhaltsverzeichnis

Abstract . . . . .	I
Gender-Hinweis . . . . .	III
Inhaltsverzeichnis . . . . .	V
Abbildungsverzeichnis . . . . .	IX
Tabellenverzeichnis . . . . .	XV
Quellcodeverzeichnis . . . . .	XVII
Abkürzungsverzeichnis . . . . .	XIX
1 Einleitung . . . . .	1
1.1 Motivation . . . . .	1
1.2 Fragestellung . . . . .	2
1.3 Grundlegende Vorgehensweise . . . . .	2
1.4 Aufbau der Arbeit . . . . .	3
2 Umgebung . . . . .	5
2.1 VR und Immersion . . . . .	5
2.2 Virtuelle Influencer . . . . .	5
3 Motion Capture . . . . .	9
3.1 Rotoskopieren . . . . .	9
3.2 Visual Motion Capture . . . . .	10
3.3 Sensorisches Motion Capture . . . . .	14
3.3.1 IMU . . . . .	14

---

3.3.2	Motion Capture durch IMU . . . . .	20
3.4	Pose Estimation . . . . .	22
3.4.1	Bildbasiertes Tracking . . . . .	22
3.4.2	Tracking mithilfe von Funkwellen . . . . .	22
3.5	Haptisches Feedback . . . . .	24
4	Stand der Technik . . . . .	29
4.1	Finger Tracking . . . . .	29
4.2	Haptisches Feedback . . . . .	30
4.2.1	Luftdruckwellen . . . . .	31
4.2.2	Vibration . . . . .	31
4.2.3	Force Feedback . . . . .	32
4.2.4	Technik aus der Robotik . . . . .	37
4.3	Stand des Markts für Konsumenten . . . . .	40
5	Design . . . . .	43
5.1	Konstruktion . . . . .	43
5.2	Das Shield . . . . .	44
5.3	Fingertracking . . . . .	45
5.4	Bewertung des ersten Designs . . . . .	52
5.5	Weitere Iterationen des Designs . . . . .	53
5.6	Letzte Reduzierungen des Designs . . . . .	57
6	Implementierung . . . . .	61
6.1	Hardware . . . . .	61
6.2	Schaltplanung . . . . .	62
6.3	Software . . . . .	65
6.3.1	Programmiersprachen . . . . .	65
6.3.2	Blueprints . . . . .	67



6.4	ESP32 Programmierung als Sender . . . . .	69
6.5	Daten auslesen . . . . .	70
6.6	Daten verarbeiten und senden . . . . .	71
6.7	Implementierung in Unreal Engine . . . . .	74
6.8	Force Feedback und Hand Posing . . . . .	76
7	Evaluation . . . . .	79
7.1	Einarbeitung . . . . .	79
7.2	Innovation der Arbeit . . . . .	79
7.3	Evaluierung des Handschuh Designs . . . . .	80
7.3.1	Kosten . . . . .	80
7.3.2	Tragekomfort . . . . .	81
7.3.3	Trackingqualität . . . . .	82
7.3.4	Haptisches Feedback . . . . .	83
7.3.5	Reparierbarkeit . . . . .	83
7.3.6	Komplexität der Konstruktion . . . . .	84
7.4	Evaluation der Software . . . . .	84
7.5	Problematiken . . . . .	86
8	Zusammenfassung und Ausblick . . . . .	87
8.1	Zusammenfassung . . . . .	87
8.2	Ausblick . . . . .	88
	Literaturverzeichnis . . . . .	91
	Eidesstattliche Erklärung . . . . .	105
	Anhang . . . . .	107



## Abbildungsverzeichnis

Abbildung 1:	Eine Visualisierung der <i>Design Science Research</i> Methode. [97] .	3
Abbildung 2:	Szene aus dem Film <i>Ready Player One</i> . Der Hauptcharakter trägt Handschuhe und eine Weste die ein haptisches Feedback übermitteln. [119] . . . . .	6
Abbildung 3:	Virtuelle 3D Influencer und Künstler. . . . .	7
Abbildung 4:	Die Nachzeichnung einer Fotografie, auch Rotoskopie genannt. [131] . . . . .	9
Abbildung 5:	Getrackte Punkte sowie ihre Bewegungen über die Zeit als Linien visualisiert. [47] . . . . .	11
Abbildung 6:	Handschuhe basierend auf das Tracken mit Markern. . . . .	12
Abbildung 7:	Tracking der Hand und Finger für Zwischensequenzen des Spiel <i>Resident Evil: Village</i> . [113] . . . . .	13
Abbildung 8:	Das Sensorenkonstrukt (rechts), mit dem Figuren (links) bewegt werden konnten. [173] (S. 93) . . . . .	13
Abbildung 9:	Inklinationskarte vom Jahr 2015. Linien zeigen die Inklinationswinkel. . . . .	15
Abbildung 10:	Die Verschiebung von Informationen aus Messdaten durch zufälliges Rauschen. [114] . . . . .	17
Abbildung 11:	Messwerte zweier Sensoren, Weiß zeigt wahrscheinlichkeit der Messwerte über Zeit. [114] . . . . .	18
Abbildung 12:	Vorhersage von Messwerten, Blau zeigt aktuelle, Lila zeigt vorhergesagte Daten. [114]. . . . .	18
Abbildung 13:	Erwartbare Störungen und miteinbeziehung dieser in der Voraussage. $k-1$ ist hierbei die aktuelle, $k$ die vorhergesagten nächsten Wertemöglichkeiten. [114] . . . . .	19

Abbildung 14: Das multiplizieren der Messwerte ermöglicht eine genauere Einschätzung der Informationsstreuung. [114] . . . . .	19
Abbildung 15: Mobile Tracker basierend auf Inertial Measurement Unitss (IMUs). 21	
Abbildung 16: Funktionen und Beispiele des OpenPose Frameworks.[53] . . . . .	23
Abbildung 17: (a) zeigt die Regionen, in der Körperteile erwartet werden, (b) zeigt die erkannten Körperteile, nachdem 25 Bilder entnommen wurden. [109] (S. 1) . . . . .	24
Abbildung 18: Umsetzung des Lernprozesses des neuronalen Netzwerks. CSI Data sind Daten aus zurückgeworfenen Funkwellen. So wird die Bewegungen des Menschen aufgenommen und direkt mit den Werten aus den Funkwellen verglichen, um die Daten verständlich zu machen und daraus ein Skelett entwickeln zu können. [139] (S. 2) . . . . .	25
Abbildung 19: Schultertaste des <i>Playstation 5</i> Controllers, der Anhand eines Motors haptisches Feedback ermöglicht.[95] . . . . .	26
Abbildung 20: Größerer ERM Motor zur Erzeugung von Vibration. . . . .	26
Abbildung 21: Der Teslasuit. [71] . . . . .	27
Abbildung 22: Anatomie der Hand mit Knochenamen.[21] . . . . .	29
Abbildung 23: Visualisierung des haptischen Feedbacks der <i>STRATOS</i> Systeme. 32	
Abbildung 24: Auf dem Markt erhältliche Exoskelette. . . . .	33
Abbildung 25: Hochtechnologische Handschuhe mit Force Feedback (FF) und taktilen Feedback (TF). . . . .	35
Abbildung 26: Funktion eines fluiden Muskels.[91] . . . . .	38
Abbildung 27: Von Hugh Herr präsentiertes intelligentes Material. [87] . . . . .	38
Abbildung 28: Eisenpartikel in Magnetorheologische Flüssigkeit (MF), wenn (wie rechts) ein magnetisches Feld vorhanden ist. [33] . . . . .	39
Abbildung 29: Der von Jonathan Blake und Hakan B. Gurocak erstellte MF Handschuh. [118] . . . . .	40
Abbildung 30: Abgenutzter <i>LucidGlove</i> in der vierten Version, nachdem Tests durchgeführt und Servomotoren abgebaut wurden. . . . .	41

Abbildung 31: Erste Designs des Shields. . . . .	44
Abbildung 32: Funktion Hall-Sensor. . . . .	47
Abbildung 33: Ein Testversuch mit einem Hall-Sensor. . . . .	47
Abbildung 34: Erste Version des Potentiometer-Halterung (blau) mit Feststellschraube (grün) für die Mutter (grau) zum Einschrauben des Alpha RV16AF20KB100KM (Alpha) Potentiometers. . . . .	48
Abbildung 35: Zweite Version des Potentiometer-Halterung inkl. Socket (gelb). . . . .	49
Abbildung 36: Erste Version der Brücke (lila) zum Finger. . . . .	49
Abbildung 37: Zweite und dritte Version der Brücke (pink und orange) zum Finger. . . . .	50
Abbildung 38: Haptische Blockade durch einen SG90 Servomotor (grün). Der dritte Motor kann die Klemme nicht mehr nach hinten bewegen, da die Klemme versucht sich nach unten zum Motor zu rotieren. Das Design der Klemme ganz rechts verhindert dies. . . . .	51
Abbildung 39: Erstes vollständiges Fingertracking Konzept. Brücke (blau) sowie Klemmen (pink, orange) wurden für das Mittelglied adaptiert. . . . .	52
Abbildung 40: Erster Handtracking Prototyp. . . . .	52
Abbildung 41: Entwicklung des Ring-Designs. Die ersten Versuche, Sensoren in das gelbe Objekt einzudrücken (links), bis diese nur noch festgehalten werden (rechts). . . . .	54
Abbildung 42: Produktion und Test des Ring Designs. . . . .	54
Abbildung 43: Das Eindringen eines Sensors in ein Plastikstück. . . . .	55
Abbildung 44: Zweite Version des Ring Designs, in der die Bourns-Potentiometer oben platziert werden (oben). Für das Eindringen des Bourns-Potentiometer in das Stück für das Grundglied (orange) ist eine bestimmte Rotation beim 3D-Druck zu beachten (unten). . . . .	56
Abbildung 45: Die dunkelblaue Schiebeleiste ermöglicht das Einschieben von Objekten. Dadurch kann die Schiebeleiste überall verwendet und sogar, im Falle von zwei 3382G-1-252G von Bourns (Bourns), direkt auf einen Handschuh geklebt werden. . . . .	58

Abbildung 46: Dünneres Shield mit Schiebeleisten. Es laufen Kupferkabel zur Stromversorgung entlang um Kabel am Board zu reduzieren. . .	58
Abbildung 47: Funktionsfähiges Handschuhdesign mit an die Finger geklebten halben Ringen, der drei Bourns Sensoren nutzt. Motoren blockieren die Fingerbewegung. . . . .	59
Abbildung 48: Verkabelung der Stromzufuhr. Die äußeren Bahnen, in denen die Motoren und Sensoren angeschlossen sind, sind über die ganze Länge verbunden. Die inneren sind nur über die Breite verbunden.	63
Abbildung 49: Aufbau des CD4051BE ICs (Multiplexers) (von oben). [138] (S. 3) . . . . .	64
Abbildung 50: Verteilung der Datenkabel. . . . .	64
Abbildung 51: Top 10 Programmiersprachen weltweit. . . . .	66
Abbildung 52: Aufbau eines Blueprint. . . . .	68
Abbildung 53: Performance von Blueprints im Editor (Rot), Developer Build (Gelb) und Shipping Version (Blau). [141] (22:58) . . . . .	68
Abbildung 54: Aufbau der Struct für Sensoren. . . . .	70
Abbildung 55: Ablauf des Starts des Programms inklusive Auslesen der Sensordaten in <i>readMultiplexer()</i> . Oben ist die Legende für dieses und kommende Ablaufdiagramme dargestellt. . . . .	71
Abbildung 56: Vor dem versenden der Daten werden diese zunächst gefiltert und anhand der Kalibrierung in ein Rotationsfenster verrechnet.	72
Abbildung 57: Ablauf des Sendens der Daten über Bluetooth an den PC oder über Wifi an den USB-Dongle. Anschließend werden die Motoren bewegt. . . . .	73
Abbildung 58: Ein Byte-Array, der bei der Verwendung von Wifi auf dem USB-Dongle erstellt wird. . . . .	74
Abbildung 59: Ablauf des Programms für den USB-Dongle. Er liest den seriellen Port, sendet die Daten an die Hand, empfängt die Daten der Hand und sendet diese über den seriellen Port weiter. . . . .	74
Abbildung 60: Längeres Byte-Array mit Nullen als Trennung. . . . .	75
Abbildung 61: Automatisches Hand Posing dargestellt an einem Würfel. . . . .	77

Abbildung 62: Ablauf für das Lesen und Speichern der Daten, vom USB-Dongle empfangen werden. . . . .	78
Abbildung 63: Informationen und Slice aus dem Slicerprogramm <i>Cura</i> . . . . .	81
Abbildung 64: Timings des Handschuhs und Dongles, nachdem Daten der Finger gesammelt, zusammen gesetzt, über Wifi übertragen, entschlüsselt und dann an Unreal Engine 4 (UE) versendet wurden. Die letzten drei Stellen zeigen die Millisekunden. Es handelt sich um den UNIX Zeitstempel in Millisekunden, der die Zeit seit dem 01.01.1970 misst. [80] . . . . .	85





## Tabellenverzeichnis

Tabelle 1: Tabelle erwerbbarer Handschuhe. . . . .	42
Tabelle 2: Gezielte (De-)Aktivierung von Switches für Kanäle . . . . .	63
Tabelle 3: Pinout Tabelle. . . . .	65
Tabelle 4: Kostentabelle verwendbarer Hardware. . . . .	82
Tabelle 5: Umfrage übers Design, den Selbstbau und Nutzung. . . . .	83



## Quellcodeverzeichnis

Codeauschnitt 6.1: Finger definieren und Objekte erstellen . . . . .	70
Codeauschnitt 6.2: Erster Sensor des kleinen Fingers wird aus Kanal 5 des ersten Multiplexers ausgelesen . . . . .	70



## Abkürzungsverzeichnis

<b>XR</b>	erweiterte Realität
<b>VR</b>	virtuelle Realität
<b>FF</b>	Force Feedback
<b>HF</b>	haptisches Feedback
<b>TF</b>	taktiler Feedback
<b>UE</b>	Unreal Engine 4
<b>MT</b>	Motion Tracking
<b>MC</b>	Motion Capture
<b>DIY</b>	do it yourself
<b>ERM</b>	elektrische Vibrationsmotoren
<b>MIT</b>	Massachusetts Institute of Technology
<b>MCHmhF</b>	Motion Capture Handschuh mit haptischem Feedback
<b>Hz</b>	Herz
<b>IMU</b>	Inertial Measurement Units
<b>MEMS</b>	mikro-elektro-mechanisches System
<b>MF</b>	Magnetorheologische Flüssigkeit
<b>Ncm</b>	Newtonzentimeter
<b>Nmm</b>	Newtonmillimeter
<b>ms</b>	Millisekunden
<b>FDM</b>	Fused Deposition Modeling
<b>SLA</b>	Stereolithografie
<b>Alpha</b>	Alpha RV16AF20KB100KM
<b>Bourns</b>	3382G-1-252G von Bourns
<b>MHz</b>	Megahertz
<b>IDE</b>	Integrated Development Environment
<b>Multiplexer</b>	CD4051BE IC
<b>DoF</b>	degrees of freedom
<b>VM</b>	virtuelle Maschinen



## 1 Einleitung

In dieser Arbeit werden aktuelle Techniken von Motion Capture (MC) beschrieben und anschließend im Kontext des Fingertrackings genauer beleuchtet. Ein integraler Bestandteil ist zudem das Thema haptisches Feedback (HF), welches insbesondere in Anwendungen mit erweiterter Realität (XR) seinen Platz findet. Hierzu werden aktuelle MC Handschuhe und deren Technik analysiert und miteinander verglichen sowie ihre Vor- und Nachteile hervorgehoben. Zudem werden sie kurz für die Kundenzielgruppe der Konsumenten beurteilt. Im Anschluss wird ein eigenes Design für das Tracken von Fingern mitsamt FF in seinem Entwicklungsprozess beschrieben und evaluiert. Dann wird die eigenständige Datenübertragung und Integration mit der *Unreal Engine 4* (UE) beschrieben sowie ein automatisiertes Hand Posing beim Greifen von virtuellen Objekten erleutert. Diese Arbeit beschränkt sich demnach auf MC und FF, befasst sich dabei allerdings nicht tiefgreifend mit der biologische Empfindung von Hautkontakt, dessen Gefühl und deren realistischen Simulation, sowie des exakten biologischen Aufbaus und Bewegungen von Fingern und Händen. Das Ziel ist, einen vergleichfähigen Motion Capture Handschuh mit haptischem Feedback (MCHmhF) zu konzipieren und diesen in UE einzubinden. Dieses Ziel wurde ausgewählt, um XR Erlebnisse sowie das animieren von virtuelle Charaktere erschwinglicher zu machen.

### 1.1 Motivation

MC ist heutzutage hochpräzise und in mehreren technischen Umsetzungen verfügbar. Gängige MC Systeme benötigen viele Kameras mit speziellen Textilien, oder bestehen aus Anzügen. Sie alle haben das Problem, dass sie sehr teuer sind. Das gleiche gilt für Ausrüstung für die virtuelle Realität (VR), welches insbesondere in den letzten Jahren massig aufkam. Die technischen Voraussetzungen sind, um in ein VR Spiel eintauchen zu können, wie es im Film zum gleichnamigen Buch *Ready Player One* dargestellt wird, wenn man sich die technischen Fortschritte anschaut, kein weiter Weg mehr. Jedoch befinden diese sich noch in einer Preisklasse, die sich selbst Entusiasten nur bedingt leisten wollen. Es fehlt an günstigen Versionen dieser Ausrüstungen, die sich in einer Preisklasse befinden, die auch für private End-Konsumenten erschwinglich sind. Insbesondere MC Handschuhe, welche für die Anwendung in VR gedacht sind, sind noch relativ teuer. Zudem besitzen nur wenige Handschuhe Features, wel-

che das Spielen von VR Games ermöglichen oder in besonderer Weise immersiver machen. Zudem sind diese Handschuhe proprietär an eigene Softwares gebunden. Die Hauptmotivation dieser Arbeit liegt darin, eine kosteneffiziente Lösung zu finden, die somit auch einem größeren Menge von Menschen ermöglicht, eine höhere Stufe der Immersion zu erleben. Ein zusätzlicher Faktor ist dabei, dass solche Handschuhe nicht ausschließlich für VR, sondern auch für die Charakteranimation genutzt werden können. Dies liegt daran, dass MC Handschuhe mit dem Hinzufügen von Tasten auch ein Handschuh für Interaktionen in VR sind. Dadurch wird demnach auch der Zugang zu MC Handschuhen vereinfacht.

Ein weiterer Nutzen dieses Handschuhs kann zudem in der Robotik gefunden werden. Dort kann der Handschuh als Interface für die Steuerung von Robotern genutzt werden kann. Die Ergebnisse dieser Arbeit werden nach der Beendigung frei veröffentlicht, sodass Menschen diese reproduzieren und selber weltweit Nutzen können. Die Implementierung erfolgt direkt in UE, in der zudem eine für die Immersion wichtige automatische Positionierung der Finger beim Greifen von Objekten entwickelt wird. Dadurch nähern sich die virtuellen Finger kurz vor dem Kontakt mit einem Objekt deren Oberfläche. Dadurch erscheinen die virtuellen Finger in der Welt integriert.

## 1.2 Fragestellung

Ziel der Arbeit ist ein MC Handschuh zu designen, der auch von technisch weniger affinen Menschen nachgebaut werden kann. Wie sehr kann so ein Handschuh eine Alternative zu kommerziellen Handschuhen sein? Ist die Nachfrage nach günstigen Selbstbau-Handschuhen überhaupt vorhanden und wie kompliziert darf dieser sein, um Menschen nicht davor abzuschrecken, ihn nachzubauen? Zudem ist fraglich ob Konsumenten das haptische Feedback und die Vorteile von Handschuhen für VR zum aktuellen Zeitpunkt verstehen und nutzen wollen.

These 1: Ein *do it yourself* (DIY) Motion Capture Handschuh mit haptischem Feedback ist eine attraktive Alternative zu kommerziellen Versionen.

These 2: Ein DIY Motion Capture Handschuh mit haptischem Feedback kann so designt werden, dass er simpel zum nachbauen ist.

## 1.3 Grundlegende Vorgehensweise

Für diese Arbeit wird auf die *Design Science Research* Methode nach Hevner zurückgegriffen.[134] Diese Methode besteht aus drei Punkten: Zuerst wird die *Umgebung* und das Problem definiert sowie zu lösende Probleme gesammelt. Die *Umgebung*



ist der Nutzungskontext sowie Personen, die das anzustrebende Werk nutzen werden. Anschließend wird im *Design Cycle* das Design erstellt und iteriert. Nach Beendigung der Designiterationen wird diese mit der *Knowledge Base* untersucht und evaluiert. Die *Knowledge Base* besteht aus Institutionen und Personen, welche, soweit Möglich, auch aus der *Umgebung* kommen. Die Ergebnisse der Evaluation werden anschließend wieder in den *Design Cycle* übernommen, wodurch das Design wieder iteriert wird, bis die durch die Evaluation gefundenen Probleme gelöst wurden. Eine Visualisierung des Modells ist in Abb. 1 zu sehen. Als Methode wird für das Designen des Handschuhs zu Anfang bewusst eine blinde Herangehensweise genutzt. So wird für die anfängliche Zeit, entgegen vieler anderer Methoden, bewusst nicht auf den Markt geblickt, um möglichst unbeeinflusst eigene Ideen entwickeln zu können. Hierzu gibt es allerdings eine Ausnahme, die in Kapitel 3.3 beschrieben wird. Erst nach einiger Zeit dient der Markt als Inspiration für weitere Iterationen und Konzepte.

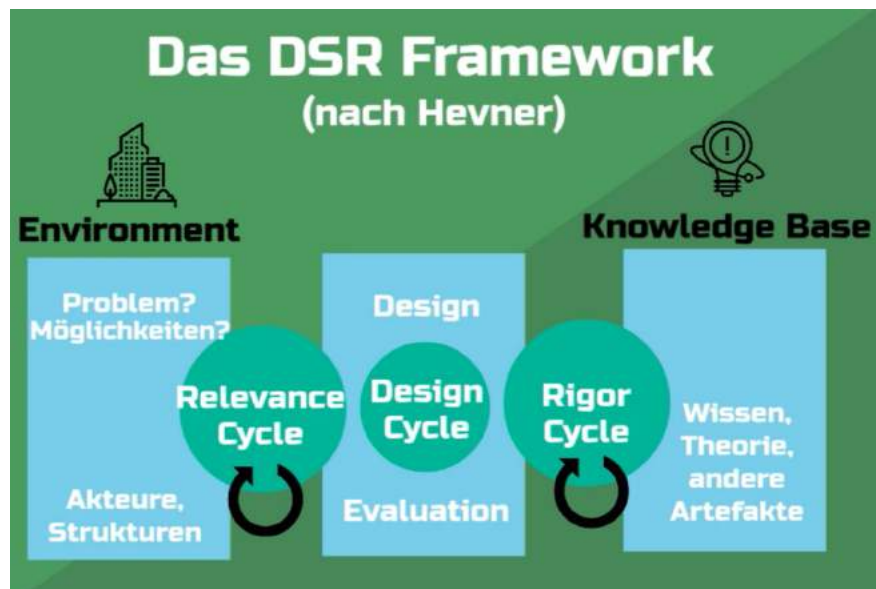


Abbildung 1: Eine Visualisierung der *Design Science Research* Methode. [97]

#### 1.4 Aufbau der Arbeit

In dieser Arbeit wird in Kapitel 2 auf die Zielgruppe geblickt und die *Umgebung* definiert, sowie Probleme erläutert. Anschließend wird in Kapitel 3 die Historie und technische Umsetzungen von MC erläutert. Dort wird auch auf den Stand des haptischen Feedbacks (HF) eingegangen, der für den Körper bereits genutzt wird. Kapitel 4 konzentriert sich anschließend auf den Stand der Technik für das Finger Tracking sowie dem HF für die Hand. Dort werden kommerzielle Produkte verglichen, Vor- und Nachteile benannt sowie ihre Features evaluiert. Anschließend werden in Kapitel 5 die Designiterationen eines eigenen Motion Capture Handschuh mit haptischem

Feedback beschrieben. In Kapitel 6 wird die Datenverarbeitung der Sensordaten des Handschuhs, das Senden und Empfangen der Daten von und zu UE sowie die automatisierte Fingerpositionierung beschrieben. Anschließend wird das Projekt in Kapitel 7 Evaluert. In Kapitel 8 wird alles zusammengefasst sowie einen Ausblick auf weitere Forschungsfragen, Designentwürfe und die Zukunft des Projekts gegeben.

## 2 Umgebung

In dieser Arbeit werden zwei Umgebungen thematisiert, die einen ähnlichen Ursprung haben, wodurch viele Lösungen für eine Umgebung auch der zweiten Umgebung nutzen kann.

### 2.1 VR und Immersion

Durch VR Brillen wie die Oculus Rift, HTC Vive und Valve Index ist es möglich, virtuelle Erlebnisse auch endlich Zuhause zu erleben. Der Preis dieser VR Brillen ist in den letzten Jahren weit gesunken, sodass diese bereits ab 349€ für die Meta Quest 2 auf dem Markt erhältlich und somit für Konsumenten erschwinglich sind.[117] Aktuelle Controller für VR Brillen bestehen zu einem Großteil aus analogen Tasten, die beim Druck die virtuellen Finger schließen lassen. Der Valve Index Controller hingegen bietet bereits Annäherungssensoren, mit denen die Entfernung zu den Fingern gemessen wird, wodurch die virtuellen Finger anhand der echten Finger bewegt werden können. Auch das zerbrechen von Objekten ist durch Sensoren, die die Griffestigkeit messen, möglich.[34] Dennoch gibt es einen Immersionsbruch, wenn Objekte gegriffen werden oder Tätigkeiten ausgeführt werden, diese man allerdings nicht *spürt*.

Der Film zum Buch *Ready Player One* zeigt auf, wie in Zukunft VR erlebt werden könnte. In dem Film trägt der Hauptcharakter eine Weste sowie Handschuhe, mit denen er die Interaktion mit der virtuellen Umgebung spüren kann, wie in Abb. 2 zu sehen ist. Die Darstellung der Immersion in einem VR Erlebnis, wie sie in *Ready Player One* dargestellt wird, dient heutzutage als Benchmark der möglichen Immersion, die es zu erreichen gilt. Ein großer Faktor darin ist unter anderem das haptische Gefühl, dass man Objekte gegriffen hat, in dem die Bewegungen der Finger durch Force Feedback blockiert werden und Vibrationen entstehen. Dieses haptische Gefühl ist allerdings noch nicht für das Nutzen im eigenen Haus erschwinglich.

### 2.2 Virtuelle Influencer

Zeitgleich hat sich auch auf Plattformen wie Twitch ein Trend entwickelt, in der sich Streamer mithilfe ihrer 2D Kamera tracken und ihre Bewegungen auf einen virtuel-



Abbildung 2: Szene aus dem Film *Ready Player One*. Der Hauptcharakter trägt Handschuhe und eine Weste die ein haptisches Feedback übermitteln. [119]

len Charakter übertragen. Nur wenige nutzen komplexere Motion Capture Systeme. Dies ermöglicht den Streamern anonym zu bleiben. Dennoch haben Zuschauer eine Verbindung zu einem Charakter, der seinen Kopf, Mund und Augen bewegt. Auch Firmen sind auf die virtuelle Charakterisierung einer Marke aufgesprungen. So werden auf dem Youtube Kanal von Netflix Anime die neuesten Animes und Trends, die auf Netflix veröffentlicht werden, von *N-ko Mei Kurono*, einem Schaf-Menschen präsentiert.[158] Die wohl bekanntesten Figuren sind *Hatsune Miku* und *Kizuna AI*. *Hatsune Miku* stellt eine virtuelle Gesangstimme aus dem Musikprogramm *Vocaloid* da.[30] Bei Konzerte von *Hatsune Miku* tanzt sie auf der Bühne, wie in Abb. 3 (a) zu sehen ist. *Kizuna AI* ist ein virtueller Influencer, die seit 2016 aktiv ist.[112] Auf dem Youtube Kanal *Ami Yamato* ist bereits seit 2011 ein virtueller 3D Charakter als visualisierung des Künstlers zu sehen.[174]

Ein großer Teil der virtuellen Streamer auf Twitch nutzen allerdings nur sehr einfache Methoden um ihre Charaktere zu animieren. Oft besitzen die Charaktere nur eine Büste, wodurch nie der Oberkörper oder sogar Hände zu sehen sind. Die Streamer *AI Angel* [8] und *Code Miko* [15] hingegen Tracken den ganzen Körper während den Streams in Echtzeit und können sich dadurch auch in einem freien Raum bewegen. Bei *Code Miko* sowie *N-ko Mei Kurono* werden allerdings die Finger nicht getrackt, wodurch diese in einer starren Position während den Bewegungen verweilen.



(a) Konzert der virtuellen Sängerin Hatsune Miku. [156]



(b) Twitch Streamer *Code Miko*. [155]



(c) Vloggerin *Amo Yamoto* während eines Makeup Tutorials. [175]



(d) Virtueller YouTuber *Kizuna AI*. [112]



(e) Virtueller YouTuber *N-ko*. [158]

Abbildung 3: Virtuelle 3D Influencer und Künstler.



## 3 Motion Capture

Um menschliche Bewegungen zu erfassen und als Animationen wiedergeben zu können, wurden viele Möglichkeiten in den letzten Jahren, nicht nur wissenschaftlich begutachtet, sondern auch gewerblich genutzt. Die wohl früheste Version ist das Rotoskopieren.

### 3.1 Rotoskopieren

Die Rotoskopie ist eine Variation der Chronophotographie. Die Chronophotographie beschreibt die Dokumentation von Bewegungen auf Film, um diese für Kunst und Wissenschaft zu dokumentieren. Dabei werden mehrere Bilder der gewünschten Bewegung hergestellt. Die Kunst der Chronophotographie wurde erstmals von Eadweard Muybridge mit der Arbeit *The Horse in Motion* im *Nature* Magazin veröffentlicht.[157]



Abbildung 4: Die Nachzeichnung einer Fotografie, auch Rotoskopie genannt. [131]

Das Rotoskopieren beschreibt die Tätigkeit, Aufnahmen nachzuzeichnen, wie in Abb. 4 dargestellt wird. Um zum Beispiel einen Menschen zeichnerisch zu animieren, kann eine Videoaufnahme erstellt werden, welche anschließend Bild für Bild nachgezeichnet wird. Dies ermöglicht eine sehr realitätsnahe Darstellung der Bewegungen, da diese entsprechend realen Bewegungen abstammen.[133] Das Rotoskopieren wurde am 6. Dezember 1915 von Max Fleischer als Patent eingereicht und schließlich am 9. Oktober 1917 patentiert.[130] Das Debut dieser Technik wurde in seiner Kurzfilm-Serie *Out of the Inkwell* gefeiert.[133] 17 Jahre später, nach der Patentfreigabe,[2] wurde

diese Technik von Disney zum ersten Mal in einem Film implementiert. Der Film war *Schneewittchen und die sieben Zwerge*, welcher nach drei Jahren Entwicklungszeit im Jahr 1937 erschien.[140, 133]

Seit 1997 findet die Rotoskopie seine Anwendung auch in der digitalen Animation. Bob Sabiston entwickelte das Animationsprogramm Rotoshop, mit dem nun Zeichnungen interpoliert werden konnten, womit nicht mehr jedes Bild neu gezeichnet werden musste. Die Interpolation berechnet die Bewegung von Punkten der Zeichnung zwischen den zwei eigenständig erstellen Posen.[133]

Das Rotoskopieren war somit der Beginn um Animationen durch Aufnahmen zu erstellen. Während dem 19. Jahrhundert entwickelte sich das MC in zwei Kategorien weiter. Einmal für das reproduzieren von Bewegungen mithilfe von Kameras und einmal über die Aufnahme von Bewegungen mithilfe von Sensoren.

### 3.2 Visual Motion Capture

Um die Objekte in ihrer Form zu messen, kann die Technik der Photogrammetrie genutzt werden, die Aimé Laussedat 1898 in seinem Buch umfangreich beschreibt. Hierzu werden Punkte eines Objekt aus mindestens zwei Winkeln vermessen und anschließend miteinander verrechnet, um die Dimensionalität dessen beschreiben zu können.[146] Mithilfe einer filmenden Kamera können damit auch Punkte verfolgt und ihre Bewegungen in einem dreidimensionalen Raum über Zeit berechnet werden, wie in Abb. 5 dargestellt wird.

Dadurch ist bereits 1975 das analysieren der Bewegungen von Kniescheiben möglich, nachdem in Knochen ein Millimeter große Stahlkugeln eingebaut und anschließend durch Röntgenstrahlung fotografiert wurden. Dies ermöglichte die Bewegungen von Kniescheiben medizinisch genauer zu untersuchen.[171] 1983 präsentierten Carol Ginsberg und Delle Rae Maxwell beim *Massachusetts Institute of Technology* (MIT) das *Op-Eye* System, welches mithilfe von LEDs an Gelenken des menschlichen Körpers die Bewegungen in Echtzeit auf ein Strichmännchen übertragen konnte. Zudem konnten die Bewegungsdaten gespeichert werden für spätere, komplexere Charaktermodelle. Die LEDs mussten hierfür für jedes Kamerabild identisch referenziert sein, sodass aus allen Blickwinkeln eine Relation zu anderen LEDs gezogen werden konnte.[151] In den frühen 80er Jahren wurde für den Werbefilm *Brilliance* eine Frau gefilmt, welcher Referenzpunkte an den Körper angemalt wurden, um anschließend ihre Bewegungen auf ein 3D Modell zu übertragen. Diese Technik wurde von Randy Roberts *Brute Force Animation* genannt.[160]



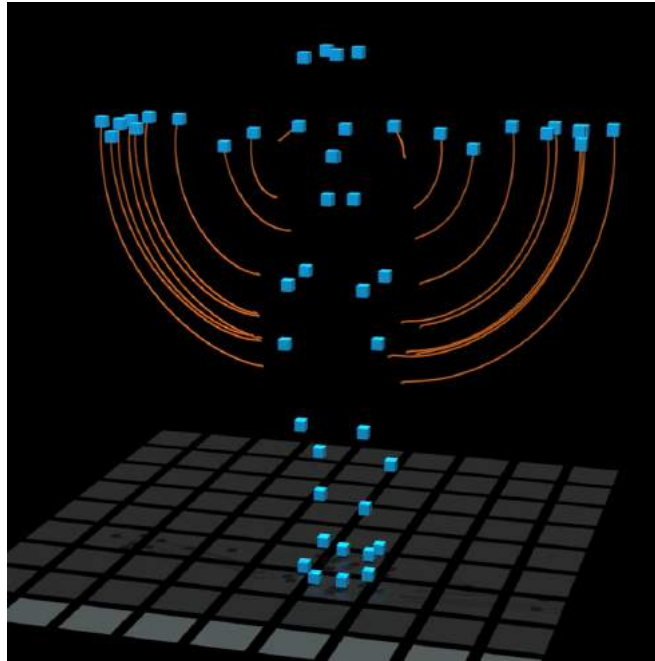


Abbildung 5: Getrackte Punkte sowie ihre Bewegungen über die Zeit als Linien visualisiert. [47]

1990 wurde schließlich das *SELSPOT 3D Motion Measurement System* von Manfred Berger und Johan Weermark entwickelt. Auch hier wurden LEDs für das System als Marker genutzt, die an das Objekt oder den Körper angebracht werden. Die Marker werden dann an einem 3D Modell, an der entsprechenden Stelle, verknüpft. Damit ist es dann möglich, ein Rig, zum Beispiel eines Menschen, nach den Bewegungen der Marker bewegen zu lassen. Kameras, die das Infrarotlicht der LEDs erkennen, erstellen Werte, die die X und Y Koordinaten jeder LED innerhalb des Bildes an einen Computer weitersenden. Da die Kameras zudem das Umgebungslicht herausfiltern, soll das System auch für Projekte im Freien geeignet sein. Dieses System erreichte mit zwei Kameras und fünf LEDs eine Aktualisierungsrate von 1000 Herz (Hz). Mithilfe von Photogrammetrie können so auch hier die Punkte dreidimensional berechnet werden.[116] Damit waren Systeme geschaffen, die das Tracken von Objekten und Menschen mit Kameras in Echtzeit ermöglichen. Die Technik des MC wurde zunächst nur für die Golf Industrie verwendet, um Golfschläge der Sportler zu analysieren und zu verbessern. Der wohl erste Film, der MC verwendete, war *Total Recall* von 1990.[129] Der Film *Der Polarexpress* war schließlich im Jahr 2004 der erste Film, der fast ausschließlich MC nutzte um seine Charaktere zu animieren. Dadurch konnte der Schauspieler Tom Hanks fünf unterschiedliche Charaktere spielen.[176] Diese MC Systeme, welche statische Relativitätspunkte und Kameras besitzen müssen, werden im weiteren Context als *Outside-in* beschrieben. Die Kosten einer Outside-in MC Anlage skalieren je nach gewünschter Qualität der Messungen sowie Größe der Raumvolumens, in der man MC betreiben möchte.



Abbildung 6: Handschuhe basierend auf das Tracken mit Markern.

Die Kosten liegen im Falle von Optitrack, mit den sechs günstigsten Kameras (die Flex 3), für eine 5 x 7 Meter Trackingfläche bei 9754\$. 96 der teuersten Kameras kosten hingegen 657.313\$, die nicht nur die beste MC Genauigkeit der Welt versprechen, sondern auch eine Fläche von  $86 \times 86 \times 9$  Meter abdecken kann.[54] Mit einem Setup von acht Flex 3 Kameras kann auf eine Genauigkeit von unter einem Millimeter getrackt werden.[24] Outside-in Tracking hat allerdings Probleme mit der Verdeckung von Punkten. Wenn also ein Punkt nicht von genug Kameras gesehen wird, dann entstehen entweder keine oder falsche Daten. Optitrack Kameras arbeiten mit Infrarot Licht, das von jeder Kamera aus ausgestrahlt wird. Die Kamera erkennt dadurch nur Punkte, die das Infrarot Licht zurückspiegeln. Entsprechend muss darauf geachtet werden, dass bei der Nutzung keine ungewünschten Spiegelungen des Infrarot Lichts von anderen Kameras entstehen. Diese können zwar in der Software ausmaskiert werden, allerdings entstehen dadurch tote Zonen, in der die Kamera keine Daten erfasst. Bewegende spiegelnde Objekte können allerdings nicht mit toten Zonen ausmaskiert werden.

Für das Tracken der Hände werden Handschuhe angeboten, an denen die Marker platziert werden können. Die Handschuhe von Optitrack sind allerdings nur für das Tracking des Daumens, Mittelfingers und kleinen Fingers ausgelegt, wie in Abb. 6 (a) zu sehen ist. Der Ring- und Mittelfinger werden oftmals aus den Trackingdaten des Zeigefingers und kleinen Fingers interpoliert. Durch das Umschnallen von Trackingpunkten an die anderen Finger können diese ebenso getrackt werden. Der Hersteller *Manus* hingegen bietet einen Handschuh an, welcher das Anbringen von bis zu zehn Markern pro Hand, mithilfe von Klettverschluss, ermöglicht. Dies wird in Abb. 6 (b) abgebildet.

Die Finger unterliegen allerdings oft dem Problem, dass sie vom Körper überdeckt werden. Bei Versuchen im Labor der Hochschule Furtwangen, welches mit zehn Flex 13 Optitrack Kameras ausgestattet ist, war es in einer Domain von 4x5 Metern, ausgerichtet für das Tracken des ganzen Körpers, nicht möglich, Trackingergebnisse aus allen Fingern zu erhalten, die ohne Nachbearbeitung nutzbar wären. Dies konnte dem häufigen Verdecken der Markerpunkte zugeordnet werden, weshalb regelmäßig ein bis vier Marker nicht mehr getrackt werden konnten. Ein Setup mit mehreren Kameras hingegen dürfte diesen Prozess ermöglichen. Beim MC von Szenen, welche nachbearbeitet werden können, da sie nicht in Echtzeit genutzt werden, werden üblicherweise nicht alle Finger getrackt. Ein Beispiel hierfür sind MC Aufnahmen für das Spiel *Resident Evil Village*, siehe Abb. 7.



Abbildung 7: Tracking der Hand und Finger für Zwischensequenzen des Spiel *Resident Evil: Village*. [113]



Abbildung 8: Das Sensorenkonstrukt (rechts), mit dem Figuren (links) bewegt werden konnten. [173] (S. 93)

### 3.3 Sensorisches Motion Capture

Lee Harrison III erschuf in den 60er den Computer *ANIMAC*, mit dem Linien in einem dreidimensionalen Raum erzeugt und zusammengebunden werden konnten. Diese Linien wurden nach Wunsch mit Potentiometern im Raum bewegt. Diese Potentiometer wurden später sogar an Menschen angebracht, wie in Abb. 8 zu sehen ist, und ermöglichten somit zum ersten Mal das digitale Erfassen der menschlichen Bewegung für den Zweck der Animation ohne Photogrammetrie. Dadurch gewann im Jahr 1972 Lee Harrison III die *National Academy of Television Arts and Science* Auszeichnung für *herausragende Leistungen in der technischen Entwicklung*. [173] (S. 92-95)

Tom Calvert, Professor der Simon Fraser Universität, nutzte in den frühen 80er Potentiometer für das Tracken von Knien. Hierfür wurde eine Art Exoskelett an die Beine angebracht, welche mit Potentiometern bestückt waren. Damit konnten die Rotationsbewegungen gemessen und am PC wiedergegeben werden. [150] (S. 5)

Mithilfe von optischen Sensoren sowie Potentiometern wurden Puppen von Dinosauriern erschaffen, welche bewegt wurden. [144] Dies ermöglichte das Animieren von Lebewesen wie bei Stop Motion Filmen, allerdings mit der Besonderheit, die Bewegungen im Nachhinein zu verändern und zu interpolieren. So wurde für Filme wie *Alien 3* und *Jurassic Park* jeweils Puppen erstellt und mit diesen animiert. [128, 78]

Durch die Weiterentwicklung des Computers konnten, im Laufe der Zeit, kleine Computer hergestellt werden, die direkt an Sensoren angeschlossen und in bewegende Objekte eingebaut werden können. Insbesondere durch die Entwicklung von leistungsstarken Mikrokontrollern, eröffnete sich eine komplett neue Art des Motion Trackings. Hierzu werden komplexere technische Module genutzt als Drehpotentiometer, um die Bewegungen zu erfassen. *Inertial Measurement Units* (IMU) sind Module, die das Tracking von Bewegungen ohne externe Module, zur Erstellung einer Relation über Zeit, messen können. Damit ist gemeint, dass IMUs, anders wie bei Visual Motion Tracking, keine Kameras oder Objekte brauchen, um von einem festen Punkt aus Daten zu sammeln. Dies wird ermöglicht durch eine Vielzahl an Sensoren, die in IMUs enthalten sind. Im weiteren Context werden diese als *Inside-out* beschrieben. IMUs besitzen einen Beschleunigungssensor, ein Gyroskop und ein Magnetometer [161, 147]

#### 3.3.1 IMU

Beschleunigungssensoren können die linear Bewegungen und die Schwerkraft messen. Da durch die Schwerkraft auf der Erde Objekte üblicherweise nach unten zum Boden gezogen werden, kann somit ein Beschleunigungssensor in einem stillen Zustand durch die Schwerkraft die Position für unten und oben erkennen. Ein Magnetsensor kann

anschließend, mithilfe des magnetischen Felds der Erde, seine Rotation bestimmen. Da allerdings das magnetische Feld auf der Erde, wie in Abb. 9 zu sehen ist, nicht jederorts identisch ist, müssen die Daten des Magnetsensors mit den Daten des Beschleunigungssensors verrechnet werden, um die Punkte der Pole für die Kalibrierung zu speichern.

Dies geschieht durch eine Bildung des Kreuzprodukts der Daten des Magnetsensors mit den Daten des Beschleunigungssensors, indem das gemessene magnetische Feld mit der Richtung der Anziehungskraft verrechnet wird. Dadurch wird Osten ermittelt. Nun kann Norden ermittelt werden, da Norden das Kreuzprodukt von Osten in Verrechnung mit der Richtung der Schwerkraft ist. [147, 127]

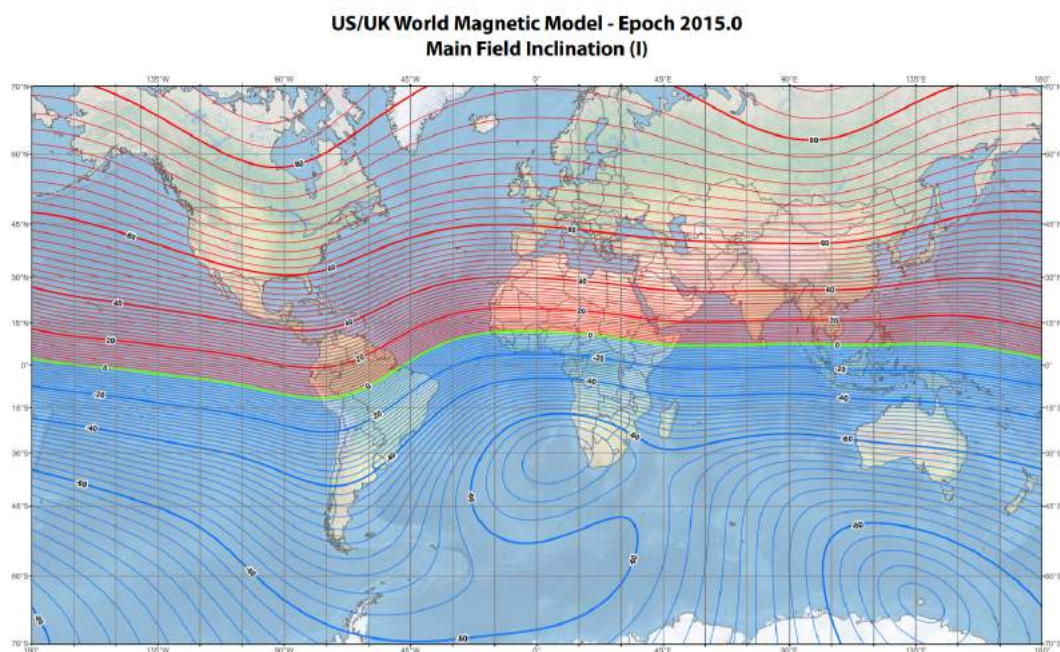


Abbildung 9: Inklinationskarte vom Jahr 2015. Linien zeigen die Inklinationswinkel.

Mit diesen zwei Sensoren zusammen kann nun die Rotationsrichtung gemessen werden sowie seine Bewegung. Da allerdings durch Veränderungen des magnetischen Felds der Magnetsensor schwankt, kann dies zu fehlerhaften Daten führen. Diese Schwankungen können durch Elektronikteile in der nahen Umgebung entstehen, welche bis zu einem bestimmten Maß allerdings gemessen und gegengerechnet werden können, um die Fehlermessungen zu verringern. Beschleunigungssensoren haben allerdings ähnliche Probleme, da zum Beispiel Rotationen, bei denen der Beschleunigungssensor sich nicht im Rotationsmittelpunkt befindet, als Bewegungen erkannt werden. Auch dies kann mit verschiedenen Methoden herausgerechnet werden, allerdings fallen hierbei entweder kurzzeitig Daten aus, oder die Bewegung des Sensors müssen bekannt sein.[127] Aus diesem Grund befindet sich im IMU ein Gyroskop.

Einfach erklärt, ein Gyroskop in einem IMU ist ein Sensor, der unabhängig von Schwerkraft oder Beschleunigung die Rotation misst. Ein digitales *mikro-elektro-mechanisches System* (MEMS) Gyroskop in IMUs funktioniert, anders als mechanische Gyroskope, mithilfe von Vibration. Das Vibrieren lässt eine Fläche still stehen, ähnlich wie es bei einem mechanischen Gyroskop durch den Drehimpuls still stehen bleibt. Durch die Corioliskraft der Rotation des Gyroskops wird anschließend Kraft in Richtung der Z-Achse auf die bewegende Fläche geübt. Diese darausbildende Bewegung sieht ein Beobachter, der fest auf der X-Achse verankert ist. Dieser Bewegung wird dann als elektrischer Strom abgelesen, wodurch der elektrische Strom die Winkelgeschwindigkeit repräsentiert.[162] (S. 13 ff.)

In dreidimensionalen MEMS Gyroskopen sind drei Gyroskopsensoren orthogonal voneinander angebracht und messen somit jeweils einzeln eine Achse, womit das dreidimensionale Messen möglich ist.[115] Die Messungen der Gyrosensoren werden normalerweise in Grad pro Sekunde angegeben. Für das Berechnen des Rotationsgrades eines Objekts wird zunächst das Objekt still abgestellt. Anschließend muss die Rotation des still stehenden Objekts gespeichert werden, damit ein Ausgangspunkt für kommende Berechnungen steht. Anschließend, bei Bewegungen, können die gemessenen Grade innerhalb des Intervalls auf die kalibrierte Stellung kontinuierlich addiert werden. Dies wird *Dead reckoning* oder auf Deutsch *Koppelnavigation* genannt. Gyroskope haben allerdings auch das Problem, dass diese nicht immer exakt sind. Je nach Integration kommt es zu verschiedenen starken Fehlern, welches einen *Drift* auslösen. Der Drift entsteht, indem mehrere Positionsfehler sich auf Dauer zu einem Drift hinzuaddieren, wodurch eine neue Kalibrierung nach einer bestimmten Zeit nötig wäre.[147, 127]

Alle drei Sensoren in einem IMU haben demnach Fehlerquellen, die aus verschiedenen Gründen geschehen. Um diese Fehlerquellen zu minimieren, werden die Signale gefiltert und vereint.

Wenn Sensordaten gegenseitig verrechnet werden, können ihre speziellen Fehler sich gegenseitig eliminieren. Hierzu gibt es ein paar Ansätze, die gefilterte Messwerte aus mehreren Sensoren erstellen können.

### 3.3.1.1 Der Komplementärfilter

Da ein Gyroskop nur geringe Abweichungen hat, diese allerdings sich mit der Zeit sammeln, kann für jede Messung ein anderer Wert gegengerechnet werden. Da mithilfe des Beschleunigungssensors ungefähr durch die Schwerkraft ermittelt werden kann, wo sich unten befindet, kann dies genutzt werden, um den Gyrosensor auf Dauer neu auszurichten. Dies ist möglich mit einem sogenannten Komplementärfilter. Hierzu gibt es verschiedene Möglichkeiten der Implementierung, die in [136] und [126] näher aus-

geführt werden. Entweder stellt man eine Grenzfrequenz der Daten beider Sensoren ein, wodurch Fehlerquellen nicht mehr durchlaufen, oder man mischt die Daten des Gyrosensors mit den Daten des Beschleunigungssensor in einem Verhältnis miteinander, sodass zum Beispiel die Daten des Gyrosensors immer ein wenig in Richtung der Schwerkraft gelenkt werden, wodurch dessen Messfehler ausgegült werden. Beide Methoden ergeben prinzipiell das gleiche Ergebnis. Bei einem Komplementärfilter müssen, je nach Umsetzung, die Hochpass- und Tiefpassfilter oder Verhältnisse der Mischung der Werte eigenständig angepasst werden.

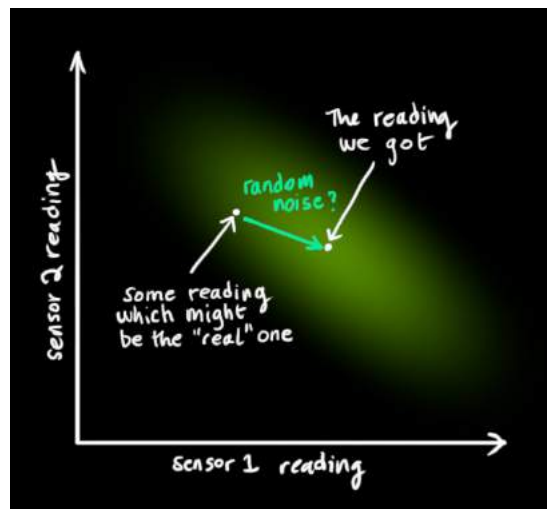


Abbildung 10: Die Verschiebung von Informationen aus Messdaten durch zufälliges Rauschen. [114]

### 3.3.1.2 Der Kalman Filter

Ein Kalman Filter hingegen ermöglicht eine automatische Anpassung der Verhältnisse. Hierfür werden in einem bestimmten Maß alle bisherigen Messungen beachtet. Die Implementierung und Mathematik wird in [166] und [142] genau beschrieben. Einfach erklärt, kann man den Kalman Filter nutzen, wenn es Sensoren gibt, zwischen deren Messungen es eine Korrelation gibt. Beispiele für Messungen können GPS Standortinformationen und Geschwindkeitsdaten eines Autos sein, um herauszufinden, wie weit sich das Auto bewegt hat, seit Antritt der Fahrt. Ebenso können auch indirektere Messwerte genutzt werden, wie die Temperatur der Reifen, da sich diese potentiell erhöht, bei schnellerem Fahren des Autos. Es ist nötig, dass immer nur, wenn ein Messwert sich erhöht, sich der andere Messwert auch verringert, oder auch erhöht. Dies deutet auf eine Korrelation hin. Andere Informationen, die von einem System ausgeführt werden, können ebenso als Daten verwendet werden. Ein Beispiel wäre der Wert, wie sehr ein autonomes Auto auf das Fahrpedal drückt.



Grundlegend haben Sensoren Ungenauigkeiten oder Störungen, wodurch auf den wahren Messwert immer ein zufälliger anderer Wert hinzugefügt wird. Mit dem Versuch aus zwei Sensoren eine Information zu finden, können falsche Daten errechnet werden. Dies wird in Abb. 10 abgebildet.

Im folgenden Beispiel nutzen wir ein Auto mit GPS und Beschleunigungssensor. In diesem Fall, da zwei Sensoren genutzt werden um eine Sache zu messen, können diese Informationen in Abb. 11 dargestellt werden.

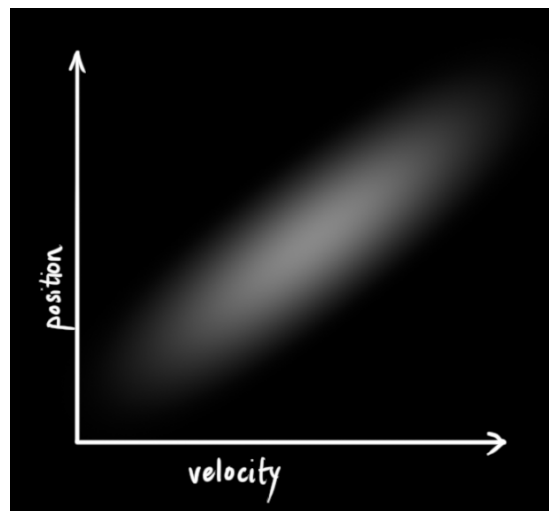


Abbildung 11: Messwerte zweier Sensoren, Weiß zeigt wahrscheinlichkeit der Messwerte über Zeit. [114]

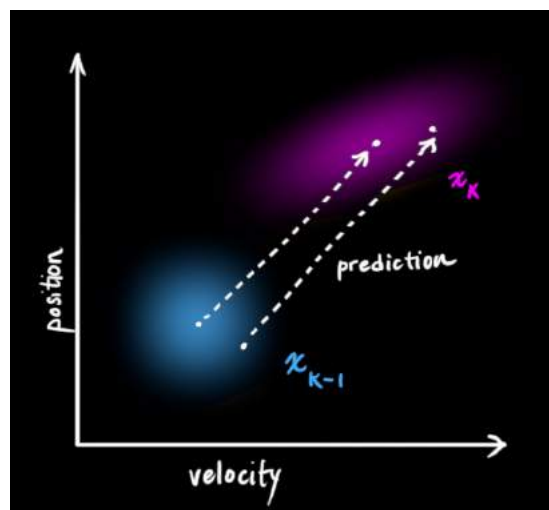


Abbildung 12: Vorhersage von Messwerten, Blau zeigt aktuelle, Lila zeigt vorhergesagte Daten. [114]

Es ist ersichtlich, dass Änderungen an einem Sensor auch Änderungen im zweiten Sensor herbeiführen. Dabei fallen unterschiedliche Werte an. Damit stellen wir fest, dass Werte eines Sensors geschätzt werden können anhand der Daten des anderen



Sensors. Nun kann eine Vorhersage gemacht werden, wohin sich die Messwerte der Sensoren im nächsten Zeitintervall bewegen. Dies wird in Abb. 12 dargestellt.

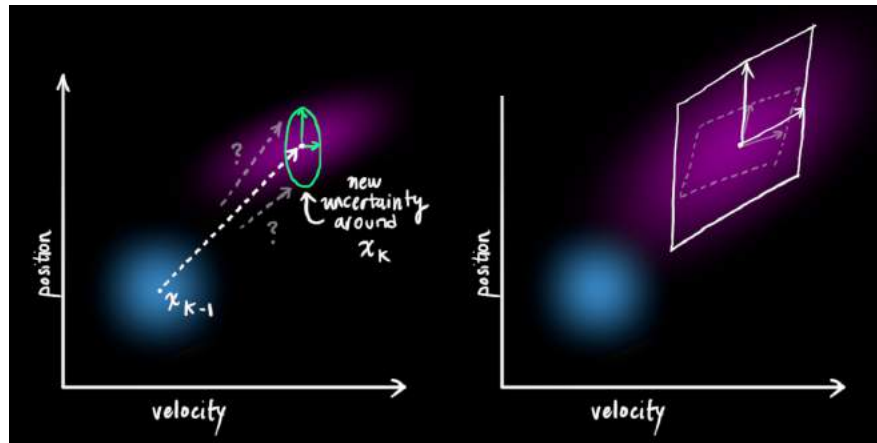


Abbildung 13: Erwartbare Störungen und miteinbeziehung dieser in der Voraussage.  $k-1$  ist hierbei die aktuelle,  $k$  die vorhergesagten nächsten Wertemöglichkeiten. [114]

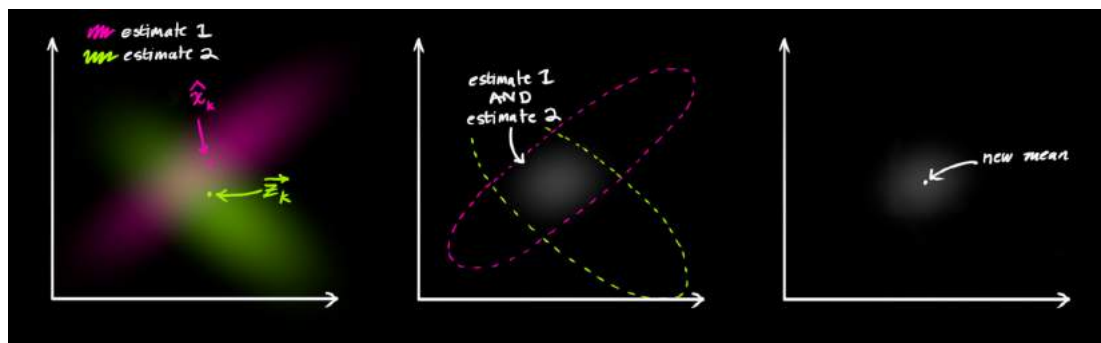


Abbildung 14: Das multiplizieren der Messwerte ermöglicht eine genauere Einschätzung der Informationsstreuung. [114]

Das Auto kann nun unvorhergesehen gestört werden, da es zum Beispiel gegen oder mit dem Wind fahren muss. Diese Störungen häufen sich mit der Zeit, die man als neues Rauschen wahrnehmen muss. Da die vorhersehbaren Messwerte dadurch eine größere Variation erhalten, vergrößert sich die Flächen, in der Werte erwartet werden müssen. Das ist in Abb. 13 ersichtlich.

Da dies ein wiederholender Prozess ist, können die aktuellen Messwerte mit dem verglichen werden, was mit den letzten Werten geschätzt werden konnte. Die Differenz zeigt dabei auf, welche unvorhergesehenen Störungen auf das Objekt wirken und ermöglicht somit deren Filterung. Nun kann eine neue Voraussage erstellt werden, welche die unvorhergesehenen Störungen mit beachtet. Da nun eine Voraussage aus dem letzten Messdaten existiert und die tatsächlichen (mit Rauschen belegte) Messdaten vorhanden sind, kann man die jeweiligen Flächen, in der der wahre Messwert enthalten ist, berechnen. Eine Multiplikation dieser beiden Flächen zeigt eine kleine-

re Fläche auf, dessen Mittelpunkt am ehesten der wahre Messwert ist. Das wird in Abb. 14 dargestellt.[114]

### 3.3.2 Motion Capture durch IMU

Mithilfe der oben beschriebenen IMUs und der möglichen Methoden der Kombination und Filterung von Sensordaten, gibt es auf dem Markt MC Anzüge, die entgegen den Outside-in Versionen keine Kameras oder Sensoren benötigen, um Bewegungen nachzuvollziehen. Diese MC Ansätze bestehen aus IMUs, die an verschiedenen Stellen des Körpers platziert werden müssen, um den ganzen Körper tracken zu können. Dies ermöglicht das Tracken unabhängig des Ortes und ist um ein vielfaches günstiger. Einer dieser Anzüge ist von XSens, dessen Preise bei 3790\$ beginnt. Die Sensoren sind im Anzug eingebaut.[48] Der *XSens MVN Link* ist in Abb. 15 (c) dargestellt.

*Notch* und *Perception Neuron* hingegen verkaufen die Sensoren mitsamt Batterien in einer Hülle, die an Halterungen angebracht und über Bänder an den Körper angezogen werden. Sechs Tracker von Notch kosten 429\$, wobei für das Tracken des ganzen Körpers 18 empfohlen werden, das einen Preis von 1287\$ hat.[50] Eins der Tracker ist in Abb. 15 (a) zu sehen.

*Perception Neuron* hingegen bietet sein günstigstes 18 Tracker System für den ganzen Körper für 2399\$ an, welches in Abb. 15 (b) dargestellt ist. Die professionelle Produktversion von *Perception Neuron* kostet 5999\$, ein paar Handschuhe für das Tracken der Hände zusätzliche 1499\$.[16]

Allerdings gibt es auch Lösungen, die selbst erstellt werden können. Vera Christ präsentiert in ihrer Masterthesis die Umsetzung eines eigenen Tracking Systems mit IMUs, die auf Basis von Arduinos und ähnlich wie die Tracker von *Perception Neuron* und *Notch* funktioniert.[124] Diese sind in Abb. 15 (d) dargestellt.

Diese MC Anzüge verrechnen ihre Daten der IMUs entweder lokal oder am externen PC zusammen. Die gemessenen Rotationsdaten werden dann so umgerechnet, dass die gemessene Rotation auf virtuellen Bones übertragen werden kann, wodurch so die Daten an einen 3D Charakter angelegt werden können.



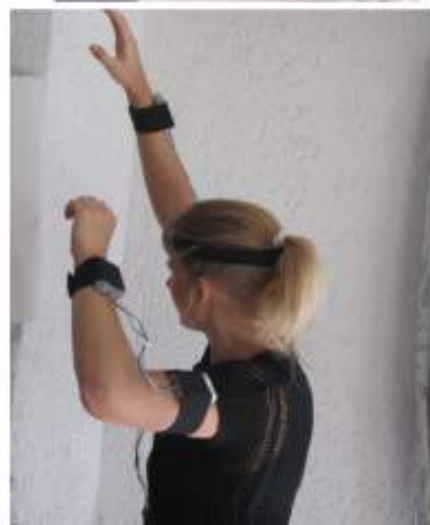
(a) Ein Tracker von *Notch*. [50]



(b) Die Tracker des *Perception Neuron 3*. [56]



(c) Der Xsens MVN Link. [46]



(d) Auf IMUs basierende DIY Tracker, die durch eine Masterarbeit entstanden sind. [124] (S. 64)

Abbildung 15: Mobile Tracker basierend auf IMUs.

### 3.4 Pose Estimation

Mithilfe von neuronalen Netzen haben sich in den letzten Jahrzehnten neue Möglichkeiten entwickelt, Lebewesen zu capturen. So kann man, ähnlich wie beim Kalman Filter, Daten miteinander verbinden, die nun aber nicht direkt in einer Korrelation stehen. Ebenso ist es dadurch möglich, Daten hauptsächlich nur durch einen Sensor zu erhalten, nachdem zuvor qualitative Daten gesammelt wurden. Die neuen Arten des Trackings nutzen zwar Technologien der sensorischen und visuellen Versionen, arbeiten diese allerdings anders auf. Daher werden in dieser Arbeit diese von den klassischen abgegrenzt betrachtet. Wichtig ist hierbei, dass es sich bei der *Pose Estimation* nur um Schätzung einer Pose oder Bewegung handelt und dadurch ein großer Raum von Fehlinterpretationen möglich ist. Es handelt sich hierbei um keine Technik, die eine höhere Genauigkeit erfassen kann als die zuvor genannten.

#### 3.4.1 Bildbasiertes Tracking

Mithilfe heutiger Computerleistung kann jeder sehr preiswert in seiner Wohnung seine Bewegungen Motion-Tracken. Wofür 2012 noch mehrere Kameras benötigt wurden [169], werden seit 2017 mit dem *VNect* nur noch eine einfache Webcam benötigt. Damit ist es möglich, die Bewegungen in Echtzeit durch ein 2D Bild zu erfassen. Die Ergebnisse sind dabei in etwa gleich wie mit einer *Kinect* von Microsoft, welche dies bisher in Echtzeit mithilfe einer Tiefenkamera [106, 12] ermöglichte. Das *VNect* unterliegt dabei nicht den Limitationen einer *Kinect*, da auch bei hellem Licht noch getrackt werden kann, sowie auch Handykameras genutzt werden können. Dennoch ist anzumerken, dass das *VNect* Probleme durch Überdeckungen, seltenen Posen, mehreren Menschen sowie überdeckte Gesichter hat. [152]

Inzwischen gibt es eine Vielzahl von Frameworks, wie *OpenPose* [135, 53], *AlphaPose* [149], *TFPose* [121], *DensePose* [165] (nun Teil von *Detectron2* [164]) und Nvidias *BodyPoseNet* [102]. Jede dieser Frameworks ist dabei in der Lage mehrere Menschen in Echtzeit mit einer 2D Kamera zu tracken. Zudem ist bei manchen das Tracken des Gesicht und der Gesichtszüge eingeschlossen. Die Funktionen von *OpenPose* sind in Abb. 16 dargestellt. Diese Frameworks wurden mithilfe von neuronalen Netzwerken entwickelt. Eine Möglichkeit, wie neuronale Netze lernen können, ist in Abb. 18 abgebildet.

#### 3.4.2 Tracking mithilfe von Funkwellen

Ein ähnliches Phänomen zeigt sich beim Tracken durch Funksignale. 2013 veröffentlichten Fadel Adib und Dina Katabi ein Paper, in dem sie ihre Entdeckung teilten,

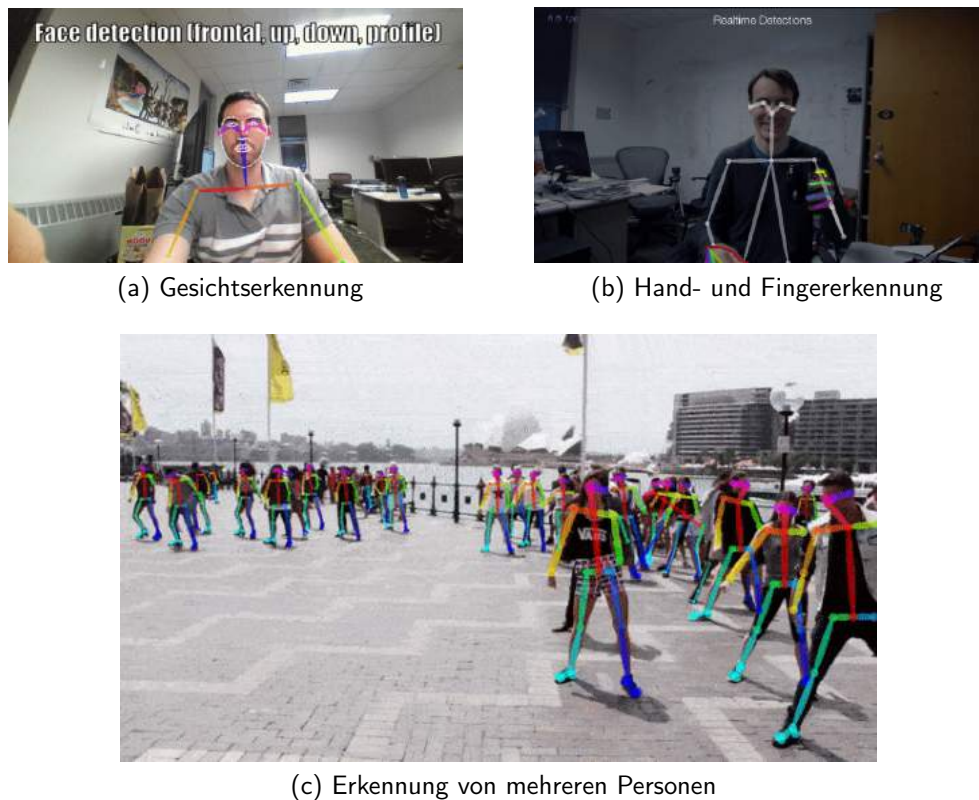


Abbildung 16: Funktionen und Beispiele des OpenPose Frameworks.[53]

dass Wifi Signale genutzt werden können, um Bewegungen von Menschen entdecken zu können. Hierfür wird ein Sender und ein Empfänger benötigt.[110] Dies wurde anschließend zwei Jahre später so weiterentwickelt, dass durch Wände nun auch die menschliche Form korrekt erkannt wurde, sowie Bewegungen von Körperteilen nachvollzogen werden konnte. Die Ergebnisse sind hierbei in der Erkennung der Körperteile innerhalb weniger Bewegungen genauer als bei der *Microsoft Kinect*. Das erkennen des Körpers funktioniert durch Funkfrequenzen, die durch Wände hindurch Funk-signale senden und die vom menschlichen Körper zurückgestrahlten Wellen wieder empfangen. Diese Wellen werden dann zur Erkennung von Körperteilen genutzt.[109] Ergebnisse werden in Abb. 17 dargestellt.

Indem ein Team nun ein neuronales Netzwerk aufgebaut hat, welches Daten aus drei Wifi Sendern und drei Empfängern mit Aufnahmen einer 2D Pose Estimation vergleicht, konnten sie mehrere Posen von verschiedenen Personen gleichzeitig einschätzen. Probleme entstanden hier allerdings, da die Ground Truth, in diesem Fall Posen, die aus 2D Kamera Daten entstanden sind, bereits nicht immer korrekt waren, womit das neurale Netzwerk Daten aus den Wifi Signalen bereits falsch erlernte. Zudem wurde festgestellt, dass manche Körperteile nicht gefunden werden können, welches mit der Frequenz des Signals zusammen hängt. Die genutzten 2.4 GHz haben nämlich eine Wellenlänge von etwa 12.5 cm, womit Objekte die kleiner sind nicht be-

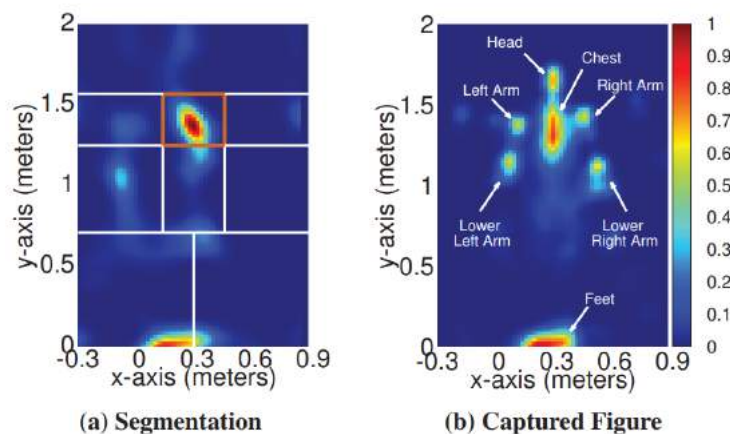


Abbildung 17: (a) zeigt die Regionen, in der Körperteile erwartet werden, (b) zeigt die erkannten Körperteile, nachdem 25 Bilder entnommen wurden. [109] (S. 1)

merkt werden konnten.[172] 2020 wurde das im Kontext des MT erweitert, in dem die Wellenlänge von 2.4 GHz auf 5.825 GHz gewechselt, sowie die Ground Truth mithilfe einer *Outside-in* MC Anlage von Vicon erstellt wurde. Die Umsetzung des Trainings des Netzwerks wird in Abb. 18 dargestellt.

Mithilfe eines eigenen neuronalen Netzwerks namens *WiPose* konnte nur mithilfe von Wifi Funkwellen und ohne Wände zwischen Mensch und Empfänger eine durchschnittliche maximale ungenauigkeit von 28.3mm der Gelenke erfasst werden. Andere Netzwerke hatten hierbei Ergebnisse von 43.6mm. Allerdings ist die Genauigkeit im direkten Vergleich mit *Outside-in* Trackings von *Optitrack* oder *Vicon* nicht sehr hoch, welche eine maximale Ungenauigkeit von 2mm versprechen. Allerdings müssen die Vorteile von Wifi Tracking erwähnt bleiben, da nämlich weder Licht noch teures Equipment benötigt wird, weil in der Testumgebung nur zehn Antennen genutzt wurden, einer für das Senden der Wellen, die anderen neun für das Empfangen der Wellen aus verschiedenen Richtungen.[139] Das Tracken von Fingern mit Funkwellen wurde bisher noch nicht erforscht. Hierfür müssten Wellen genutzt werden, die kleiner als die Dicke der Finger sind, um diese überhaupt erkennbar zu machen.

### 3.5 Haptisches Feedback

Haptische Feedback ist insbesondere ein Thema für Videospiele, welches immer wieder neue Ideen hervorbrachte, um die Immersion in Spielen zu verstärken. Neben Vibrationsmotoren, die in verschiedenen Mengen und Orten über Konsolengenerationen in Controller eingebaut wurden, fügte Sony für die *Playstation 5* eine bis dahin neue Art vom haptischem Feedback für Controller hinzu. Mithilfe eines Motors wird, wie in Abb. 19 gezeigt, ein Zylinder mit einer Spirale gedreht, welches ein Zahnrad blockie-



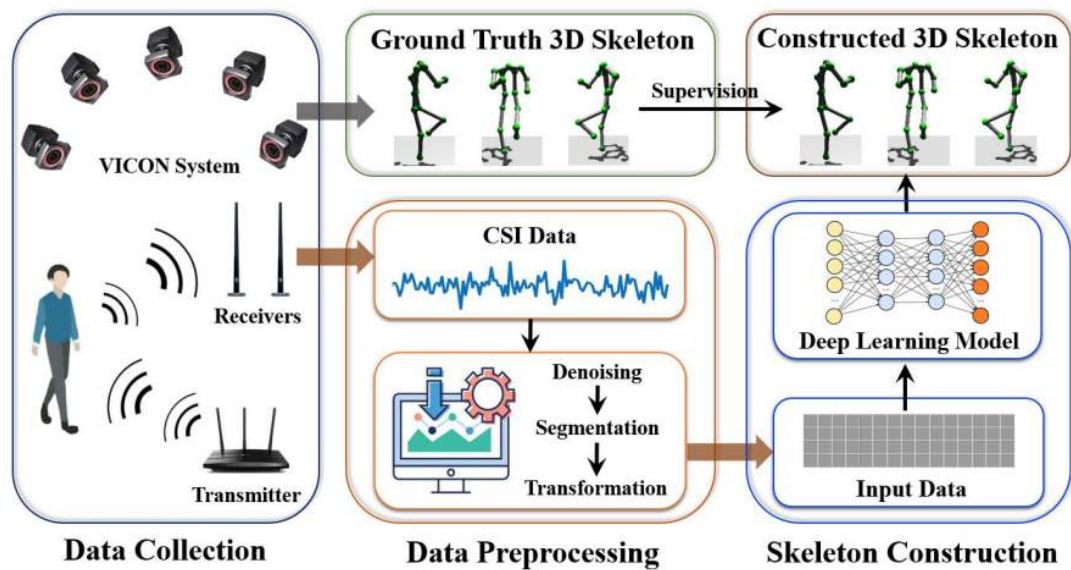


Abbildung 18: Umsetzung des Lernprozesses des neuronalen Netzwerks. CSI Data sind Daten aus zurückgeworfenen Funkwellen. So wird die Bewegungen des Menschen aufgenommen und direkt mit den Werten aus den Funkwellen verglichen, um die Daten verständlich zu machen und daraus ein Skelett entwickeln zu können. [139] (S. 2)

ren kann, das über eine Feder mit der Schultertaste verbunden ist. Beim Drücken der Schultertaste kann der Zylinder durch den Motor gedreht werden, um der Bewegung der Schultertaste Kraft entgegenzusetzen. Damit ist eine Vielzahl von haptischen Simulationen möglich, wie das Zusammendrücken einer Feder, dem Anspannen eines Bogens, oder das Drücken einer Taste.[100]

Auch für VR Anwendungen gibt es bereits Systeme, die ein haptisches Feedback am Körper ermöglichen. Die Firma *bHaptics* bietet für Konsumenten den *TactSuit X16* für 299\$ und *X40* für 499\$, welche dem Namen entsprechend 16 und 40 elektrische Vibrationsmotoren (ERM) beinhaltet.[23] ERMs erzeugen, durch das schnelle Rotieren eines halben Kegels, eine Vibration. Solch ein Motor wird in Abb. 20 dargestellt. Die Vibration entsteht durch eine unregelmäßige Zentripetalkraft, welche durch die unterschiedliche Gewichtung entsteht, da nur ein in der Länge geschnittener halber Zylinder vorhanden ist. Durch eine schnelle Rotation wirkt diese Zentripetalkraft wie ein Vibration, welche spürbar ist.[6]

*Shockwave* ist ein haptischer MC Anzug, der bald erscheinen soll. Dieser setzt ebenso auf ERM, beinhaltet allerdings acht IMUs um den Körper tracken zu können. Ein Preis ist noch nicht bekannt, allerdings wird er, anhand der Preise des Kickstarter Projekts, wohl um die 350\$ kosten.[64, 65]



(a) Offene Schultertaste



(b) Schultertaste weiter auseinandergebaut

Abbildung 19: Schultertaste des *Playstation 5* Controllers, der Anhand eines Motors haptisches Feedback ermöglicht.[95]



Abbildung 20: Größerer ERM Motor zur Erzeugung von Vibration.

Eine Lösung für Unternehmen bietet der *TeslaSuit*, der vollgepackt mit Technik ist. Der *TeslaSuit* ist in Abb. 21 abgebildet. Er beinhaltet für das MC zehn [71, 159] IMU [154] Sensoren. Zudem findet das haptische Feedback über die elektrische Stimulationen der Muskeln statt.[159] Damit soll es möglich sein, das Gefühl von Regentropfen bis hin zu Stichwunden und Kugeltreffer zu simulieren. Es sei allerdings nicht konzipiert dem Nutzer Schmerzen oder Verletzungen zuzufügen. Jeremy Orr, ein Besitzer eines solchen Anzugs, sagte in einem Interview, dass das Gefühl nicht perfekt nachproduziert werden muss. Das Gehirn würde viele Fehler verzeihen und sich die nötigen Informationen selbst zusammenreimen, im Kontext des VR Erlebnisses.[163] Der *Teslasuit* besitzt zudem noch weitere Sensoren um unter anderem das Stresslevel und den emotionalen Status zu überprüfen.[71] Die Anzüge werden bereits von Polizeieinheiten und Piloten zum Training verwendet. Für eine Verwendung für Sport und der medizinischen Rehabilitation wird aktuell noch geforscht.[159]





Abbildung 21: Der Teslasuit. [71]



## 4 Stand der Technik

In diesem Kapitel wird der aktuelle Stand der Technik von Motion Capture Systemen und haptischen Feedbacks für Hände und Finger beleuchtet. Für die weitere Benennung der Fingerknochen werden die in Abb. 22 dargestellten Namen verwendet.



Abbildung 22: Anatomie der Hand mit Knochennamen.[21]

### 4.1 Finger Tracking

Das Tracking der Finger in den in Tabelle 1 aufgelisteten Handschuhen, die erwerbbar sind, ist ersichtlich, dass sich die Technik der Finger grundsätzlich nicht besonders unterscheiden. Das Tracken mit IMUs und Flex Sensoren ist die am meistgenutzte

Technik. Das Tracking durch IMUs wird dabei, ähnlich wie beim Tracken des Körpers, verarbeitet und gefiltert. Die Rotationsdaten der IMU werden an die digitalen Bones übertragen. Flex Sensoren sind Sensorstreifen. Diese Streifen schließen einen Stromkreis mit einem Widerstand. Dieser Widerstand, der dem Stromkreis zugefügt wird, wird durch die leitende Tinte auf dem flexiblen Sensor auf natürliche Weise zugefügt. Beim Biegen dieses Sensors verlängert sich der Weg durch die flexible Tinte, wodurch sich der Widerstand erhöht. Diese Veränderung des Widerstands kann gemessen werden und gibt somit an, wie sehr der flexible Sensor gebogen wird. Durch die Anbringung an einen Finger, kann man dadurch messen, wie stark sich der Finger, oder bei kleinen flexiblen Sensoren, ein Fingergelenk, gedreht hat. Diese Information kann dann wieder genutzt werden um die virtuellen Bones eines Charakters zu animieren. Der Vorteil dieser Flex Sensoren liegt darin, dass diese einfach unbemerkt in einen Stoff eingebaut werden können. Allerdings nutzen sich die Flex Sensoren über eine lange Zeit ab, wenn diese stark gebogen werden.[36]

Der *HaptX Glove DK2* hingegen bietet eine einzigartige Trackingmethode, da der Handschuh ein eigenes magnetisches Feld aufbaut. An den Fingern befinden sich so vermutlich erweiterte Hall-Sensoren, die das magnetische Feld messen, um die Position in Abhängigkeit des magnetischen Feldes zu erkennen. *HaptX* verspricht die Fingerposition genauer als ein Millimeter bestimmen zu können.[32]

Um das Tracking der Handposition zu ermöglichen, besitzen gewerbliche Handschuhe für gewöhnlich Adapter um in Trackingsysteme eingebunden zu werden. So kann für Systeme, die das Outside-In Lighthouse System nutzen, ein Vive Tracker angebracht werden. Im Falle von VR Headsets, die *Inside-Out* mit Kameras nutzen, können die von der VR Brille getrackten Controller an den Handschuhen angebracht werden. Anschließend muss eventuell der Positionsunterschied der Hand zum Tracker kalibriert werden.

Die *Meta Quest 2* ist sogar in der Lage die Hand nur mit den angebrachten Kameras zu tracken.[94] Mithilfe eines Armbands kann das Drücken auf Oberflächen erkannt werden, womit unter anderem auf einer virtuellen Tastatur geschrieben werden kann.[153]

## 4.2 Haptisches Feedback

Für haptisches Feedback gibt es einige unterschiedliche Konzepte und herangehensweisen, wie das Gefühl von Objektpräsenz anderer Objekte simuliert werden soll. Diese Konzepte basieren zu meist auf diese zwei Möglichkeiten: Feedback über Luftdruckwellen (wie zum Beispiel durch Ton) und Vibrationen durch Motoren. Gesondert wird

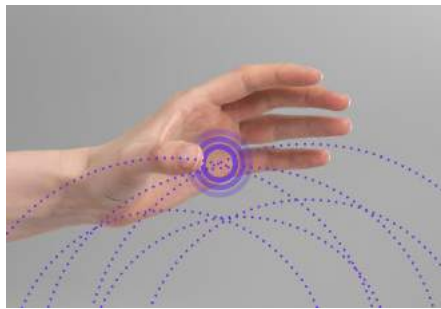
anschließend die Simulation durch kleine Druckpunkte betrachtet. Zudem wird versucht, technische Konzepte aus der Robotertechnik in den Kontext des haptischen Feedback zu übertragen.

#### 4.2.1 Luftdruckwellen

Ein Konzept für taktilen Feedback ist die Übertragung mithilfe von Luftdruckwellen. Das Gefühl von Präsenz und der Oberflächenstruktur wird *taktilen Feedback* genannt. Ultraleap bietet hierfür die *STRATOS Explore* für Entwickler und die technisch identische *STRATOS Inspire* für die tatsächliche Anwendung in Systemen an. Mithilfe von 256 Hochfrequenzwandlern, die in einem 16 zu 16 Array aufgebaut sind, können für den Menschen nicht hörbare Töne abgespielt werden, welche dennoch spürbar sind. Mithilfe einer enthaltenen *Leap Motion*, welche die Bewegungen der Hand und Finger mit einer Tiefenkamera trackt, kann von den *STRATOS* Modellen ein Muster erstellt werden. Durch das verzögerte abfeuern von Wellen und Steuerung deren Stärke können 3D Modelle in Wellenform erzeugt werden, wie in Abb. 23 dargestellt wird. Beim Aufprall dieser Wellen auf der Haut können somit neben einfachen Dingen wie Knöpfe auch eine statische elektrische Ladung, Wasserwellen und komplexere Formen interpretiert werden.[93] Dies bringt allerdings einige Nachteile mit sich. Der Hersteller gibt an, dass die Hände maximal 70cm von den Wandlern entfernt sein dürfen.[69, 70] Zudem ist die Größe des haptischen Feedbacks auch auf die Fläche des *STRATOS* beschränkt, da darüber hinaus keine Wellen mehr erzeugt werden können. Sicherlich sind Maßanfertigungen mit einer größeren Fläche für massenproduzierte Produkte für Großkunden möglich. Letztendlich ist hier aber wichtig zu erwähnen, dass die *STRATOS* Systeme immer relativ statisch zum Nutzer positioniert sein muss (wie in einem Auto, wenn das eingebaute *STRATOS* System sich mit dem Nutzer bewegt). Im Kontext von VR muss sich somit der Spieler mit dem Körper in Richtung des *STRATOS* Systems gedreht halten. Zudem kann nur das Gefühl von Präsenz simuliert werden, die Hand und dessen Bewegungen kann allerdings nicht gestoppt werden.

#### 4.2.2 Vibration

Haptisches Feedback mithilfe von Vibrationsmotoren ist wohl die häufigste Art für eine haptische Integration. Der *Prime X* Handschuh von Manus wird unter anderem mit einer haptischen Erweiterung verkauft, welcher an jeden Finger an der Oberseite ein Vibrationsmotor hinzufügt.[58] Der *Senso Glove DK3* beinhaltet sogar sechs Vibrationsmotoren. [73] Der *Sensorial XR* Handschuh bietet mit zehn Vibrationsmotoren die meisten. Hier besitzt jeder Finger an den Kuppen ein Vibrationsmotor, sowie auf der unteren Handfläche weitere fünf. [83]



(a) Haptik durch Arrays von Ultrasound Lautsprechern [68]



(b) Eine durch die *STRATOS Explore* erstellte Form visualisiert auf Öl.  
[90]

Abbildung 23: Visualisierung des haptischen Feedbacks der *STRATOS* Systeme.

Die Nützlichkeit von Vibrationsmotoren kann allerdings Fragen aufwerfen. Eine Befragung, nachdem Nutzer Aufgaben in VR lösen mussten, zeigt, dass durch Vibration insgesamt am ehesten die gefühlte Präsenz von Objekten gesteigert wird. Die Präsenz ist sogar höher als durch FF, wobei FF die Geschwindigkeit zur Ausübung der Aufgaben erhöhte.[145] Grundsätzlich wird hiermit eine Präsenz an den Nutzer ermöglicht, wenn die Vibrationsmotoren anfangen zu vibrieren, wenn zum Beispiel ein Objekt berührt wird. Allerdings sind diese Vibrationsmotoren, wenn diese in einem weichen Stoff eingewoben werden, nur punktuell, da die Vibrationen nicht über den Stoff verteilt werden. Bei festen Materialien kann sich allerdings die Vibration eher ausbreiten. Der *Sensorial XR* Handschuh wird mit einem *Touching Masterpieces* Projekt beworben, in dem blinde Menschen 3D Modelle von Skulpturen aus der Geschichte erfühlen sollen, um deren Form verstehen zu können.[84] Die blinde Pianistin Ráchel Skleničková findet und identifiziert Teile eines Gesichts, wie die Nase, die Lippen und das Kinn. Sie war ebenso, zu einem anderem Zeitpunkt, in der Lage das Kleid der Statue *Venus von Milo* wiederzuerkennen.[96] (1:00-1:38)

Im Kontext von VR ist allerdings gegeben, dass die virtuelle Hand durch statische Objekte wandern kann. Dadurch kann die Hand unter anderem in Objekte verschwinden. Hierbei kann die Vibration genutzt werden, um dies als Gefühl zu verdeutlichen. Genaue taktile Simulationen wie mit *STRATOS* Systeme sind allerdings nicht möglich, wodurch Effekte wie statische elektrische Ladungen oder Wasserwellen mit Handschuhen mit Vibrationsmotoren eine gefühlte niedrigere Auflösung besitzen. Vibrationen sind nicht in der Lage, das Schließen der Finger mit einer Kraft aufzuhalten, wenn der Nutzer virtuelle Objekte greifen möchte.

#### 4.2.3 Force Feedback

Damit die Finger bei Interaktionen in VR nicht ins Leere greifen, müssen diese in der Bewegung gestoppt werden. Dies wird Force Feedback genannt und ist in den teureren

(a) Das Exoskelett *Dexmo*. [74](b) Der *CyberGrasp*. [19]

Abbildung 24: Auf dem Markt erhältliche Exoskelette.

Handschuhen inbegriffen. Damit ist es möglich die Finger in der Greifbewegung zu stoppen, sodass keine Faust gebildet werden kann. So wird direkt das Gefühl übertragen, ein Objekt ohne Gewicht in der Hand zu halten. Das ist mit Vibrationsmotoren und Luftdruckwellen nicht möglich. Für die Umsetzung von FF gibt es bereits ein paar Produkte auf dem Markt. Diese unterscheiden sich grundsätzlich darin, ob es sich um Exoskelette handelt oder nicht. Ein Exoskelett ist eine Konstruktion, die dem menschlichen Skelett nachempfunden ist und außerhalb des Körpers angebracht wird. Exoskelette können unter anderem genutzt werden, um Lebewesen bei Aktivitäten mechanisch oder sogar mit dem Zusatz von Motoren zu unterstützen.

#### 4.2.3.1 Exoskelette

Der *Dexmo*, der in Abb. 24 (a) dargestellt ist, ist ein tragbares Exoskelett, welches mithilfe von Servomotoren die Bewegung des Fingers einschränken kann. An den Knöcheln befindet sich hierfür ein Konstrukt, welches einen Servomotor sowie einen Drehsensor beinhaltet. Da die Finger über eine Brücke mit dem Servomotor direkt verbunden sind, kann nicht nur die Bewegung des Fingers beim Greifen gestoppt werden, sondern auch der Finger wieder in die flache Hand gezogen werden. Der *Dexmo* besitzt zudem noch Sensoren für das Tracken für das Spreizen der Finger. Für den Daumen werden sogar drei Drehsensoren genutzt, um das Spreizen, Rotieren und Biegen des Daumen tracken zu können. Dennoch wird deutlich, dass für alle Finger vom Zeige- bis zum kleinen Finger nur die Rotation des ganzen Fingers getrackt wird, wodurch die Glieder interpretiert werden müssen. Zudem wird angemerkt, dass die Kraft der Servomotoren an die Fingerkuppen wirken. Damit werden beim Greifen von Objekt alle Fingergelenke gestoppt. Die Motoren selber sind jeweils nicht so stark wie die durchschnittliche Fingerkraft, wodurch, wenn man es drauf ankommen lässt, man den Motoren überwinden kann.[92] (S. 1 ff.) Der *Dexmo* ist aktuell nur in einer Entwickler- und Unternehmensversion vorhanden. Eine Konsumentenversion ist noch

in der Entwicklung. Auf der Webseite steht, dass zwar der Preis noch unbekannt ist, aber erwartet wird, dass es sich um ca. 100\$ handeln soll.[74]

Ein weiteres Exoskelett ist der *CyberGrasp* [20], der in Abb. 24 (b) dargestellt ist, welcher an den *CyberGlove* [18] angehängt wird. Der *CyberGrasp* ist allerdings nicht einfach so kabellos nutzbar, da dicke Kabel an ihm befestigt sind um das Force Feedback zu ermöglichen. Die Kabel führen in ein Gerät, welches lineare Aktuatoren besitzt. Diese Aktuatoren ziehen an Kabel, die über das Exoskelett führen, welche mit den Fingern verbunden sind. Damit wird die Kraft gegen die Finger nicht auf der Hand erzeugt. Dies beschränkt die Nutzbarkeit des *CyberGrasp*, wobei das externe Gerät mit den Aktuatoren auch in einen Rucksack eingelegt werden kann. Dadurch ist er im direkten Vergleich zum *Dexmo* dennoch unhandlicher, kann aber mehr Kraft auf jeden Finger ausüben. Die Kraft pro Finger wird beim *CyberGrasp* mit 12 Newton angegeben. Aufgrund dessen, dass die Kraft durch an die Finger angehängten Kabel ausgeführt wird, kann keine Kraft zum schließen der Hand erstellt werden.[20] Dies führt dazu, dass entgegen des *Dexmo* keine Simulation von pochenden Objekten möglich ist, wie zum Beispiel das halten eines Herzens. Der Hersteller des *CyberGrasp* bietet zusätzlich die Erweiterung *CyberForce* [17] und davon eine weitere Weiterentwicklung, die *Haptic Workstation* [27], an. Die *Haptic Workstation* ist eine Arbeitsstation, an die beide Hände über ein *CyberForce*, welche an den *CyberGrasp* und somit den *CyberGlove* angeschlossen werden. Der *CyberForce* ermöglicht das mechanische Einschränken des Arms.

#### 4.2.3.2 Force Feedback basierte Handschuhen

Um Force Feedback zu ermöglichen ist nicht zwingend ein Exoskelett nötig. Dies beweist der *HaptX Glove DK2*, welcher durch viele Innovationen und Besonderheiten die wohl fortgeschrittenste und beste erwerbbar Version eines Motion Capture Handschuhs mit taktilen und Force Feedback für Unternehmen beinhaltet. Neben einem eigenem System am Handschuh, welches ein eigenes magnetisches Feld aufbaut, um die Fingerposition durch das erstellte magnetische Feld zu tracken, besitzt der *HaptX Glove* zudem noch eine *mikrofluidische Haut*. Diese Haut ist an den Fingerspitzen sowie an der unteren Handfläche positioniert und besteht aus kleinen Räumen, in denen Luft hineingepumpt wird. Dadurch erhebt sich eine kleine runde Fläche, wie in Abb. 25 (b) dargestellt wird, bis zu 2mm hoch, welche auf die Haut Druck ausübt und somit wahres TF ermöglicht.[28] Diese kleinen Flächen existieren mehr als 130 mal in einem Handschuh. Dies ermöglicht das Simulieren von Regentropfen, Konfetti, rauen Oberflächen sowie sogar dünne Spinnenbeine auf der Handoberfläche.[89] Force Feedback wird, ähnlich wie beim *CyberGrasp*, mit einem Metallfaden erzeugt, der an den Fingerkuppen zieht, um das Bewegen der Finger zu stoppen. Zusammen mit der





(a) Der *HaptX* Handschuh. [35]



(b) Die *mikrofluidische Haut* im *HaptX* Handschuh. [28]



(c) Fläche an der Unterseite der Fingerkuppe des *Meta VR Glove*, die sich in seiner Rotation bewegen kann. [99]



(d) Der *Meta VR Glove*. [99]

Abbildung 25: Hochtechnologische Handschuhe mit FF und TF.

*mikrofluidischen Haut* ist somit ein einzigartiges Gefühl von Präsenz von Objekten vorhanden. Ähnlich wie der *CyberGrasp* wird allerdings in der DK2 Fassung noch eine kleine Box benötigt, welche den Luftdruck in der *mikrofluiden Haut* steuert. Dieser kann allerdings auch in einen Rucksack gepackt werden.[28]

Die Firma *Meta*, ehemalig *Facebook*, präsentierte am 16. November 2021 ihre eigenen Fortschritte beim Entwickeln eines Handschuhs für VR. Dort wird neben der Technik auch darauf hingewiesen, dass viel daran geforscht wird, Gewicht durch Sehen, Hören und HF zu simulieren. Meta setzt bei ihrem Handschuh auf das Füllen von Luftpolster an der Unterseite der Hand, um das Greifen zu blockieren. Dies dürfte sogar theoretisch eine Simulation von weichen Objekten ermöglichen. Zusätzlich sind an den Fingerkuppen Mechaniken am Werk, um eine Fläche zu drehen, um auch das seitliche Halten von Objekten an der Fingerkuppe zu simulieren. Diese sind in Abb. 25 (c) abgebildet.[99] Ebenso ist, wie im *HaptX* Handschuh, *mikrofluide Haut* enthalten. *HaptX* verurteilt dies in einem Statement jedoch, da einige Komponenten, inklusive der *mikrofluide Haut* von *HaptX* patentiert sind und *Meta* sich dennoch dieser Idee bedient hat. Eine Zusammenarbeit zur Erforschung einer Konsumentenversion wird aber nicht ausgeschlossen.[29]

Lucas De Bonet hingegen, ist Bachelorstudent in Elektrotechnik und Informatik beim MIT [39] und baut unter dem Namen *Lucas VRTech* die *LucidGloves* als privates Projekt und veröffentlichte seine Fortschritte auf TikTok[40] und Youtube[41] sowie seit dem dritten Prototyp seine Dateien auf GitHub.[42] Dieser Handschuh hat im vierten Prototyp FF erhalten. Das System läuft über einen Faden, der von einer Spule über den Finger zur Fingerkuppe führt. An der Spule befindet sich ein Drehpotentiometer, der die Drehung der Spule misst. Die Spule wird durch einen SG90 Servomotor gestoppt, wodurch die Finger nicht mehr weiter greifen können.[103] Die *LucidGloves* ist ein öffentliches Projekt zum selber nachbauen. Der Preis für die nötigen Teile inklusive Filament für einen 3D Drucker für ein Paar beträgt in etwa 60\$ .[107] Für weitere Prototypen des Handschuhs sind kleinere Sensoren,[104] Temperaturregulation,[101] sowie Sensoren um die Force Feedback Simulation von elastischen Objekten zu ermöglichen, geplant.[105] Die Integration in Spiele geschieht durch einen eigenen Treiber, welcher den Handschuh als Valve Index Controller simuliert. Dadurch ist das Spielen vieler SteamVR Spiele ohne Modifikationen dieser möglich.[51]

#### 4.2.4 Technik aus der Robotik

In der Robotik gibt es technische Innovationen, welche zwar bisher nicht für haptisches Feedback in gewerblichen Handschuhen genutzt wurden, in den allerdings Potential besteht.

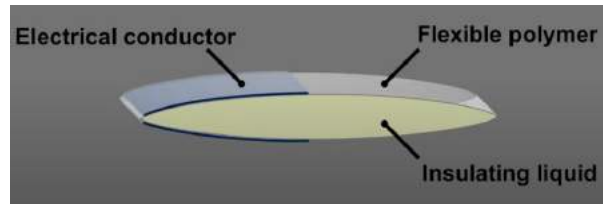
##### 4.2.4.1 Soft Robotics

In den letzten Jahren hat sich in der Robotik die Forschung immer weiter zu sogenanntem *Soft Robotics* weiterentwickelt. Anstatt technische Motoren oder Hydraulik zu nutzen, um künstliche Muskeln zu erstellen, MF um einen Roboter zu bewegen, wird immer mehr auf weiche Stoffe gesetzt. Diese benötigen nichts anderes als eine elektrische Spannung um sich zu erhärten oder zu vergrößern. So gibt es Flüssigkeiten und Stoffe, die zur Verhärtung genutzt werden können. Wilhelm Conrad Röntgen veröffentlichte 1880 einen Artikel über Veränderungen der Form und des Volumens von manipuliertem Kautschuk, welches sich durch Strom veränderte.[167] Diese Forschung wurde 2010 in einer neuen Weise erprobt. Da ein magnetisches Feld durch Elektrizität erstellt wird, können elektrische Leiter mit Strom versorgt werden, wodurch sich diese zusammen ziehen. Wenn sich zwischen beiden elektrischen Leitern eine Flüssigkeit, die wie Öl keinen Strom leitet, in einem Beutel befindet, ziehen sich die Leiter bei der Stromzufuhr zusammen und drücken das Öl weg. Der Beutel zieht sich dadurch zusammen, wodurch Kraft entsteht.[143] Dies wird in Abb. 26 dargestellt.

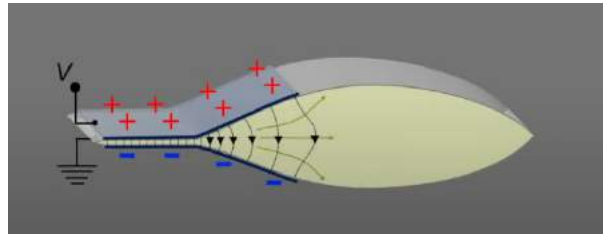
Diese Methode dürfte eine Möglichkeit sein, falls Stromleistung, Stromverbrauch und Größe in einem Handschuh Platz finden, Force Feedback in Handschuhe einzubeziehen. Zusätzlich ist die Verzögerung abhängig von der Größe des fluiden Muskels. So ist denkbar die Beutel so zu platzieren, dass die Beulen, wenn sich die Leiter zusammengezogen haben, unter den Fingergliedern befinden. Ebenso können sie sich auch auf der Handoberfläche befinden um Fäden, die die Fingerkuppen halten, zu ziehen. Eine Simulation von weichen Materialien, die man zusammendrücken kann und dann fester werden, ist hier gegeben.

##### 4.2.4.2 Intelligente Stoffe

Eine möglicherweise geeignetere Lösung bestünde in intelligenten Materialien, die sich durch elektrischen Strom erhärten. Hugh Herr stellt dieses Material, entwickelt vom *SRI International* in Californien, kurz in seinem Ted Talk von 2014 vor. Dieses Material sei flexibel, wenn kein Strom durch das Material läuft. Durch eine Stromzufuhr würde dieses Material sofort erhärten, wie in Abb. 27 dargestellt wird.[87] (7:14 - 7:32)



(a) Aufbau Fluider Muskel.



(b) Fluider Muskel bei Stromzufuhr.

Abbildung 26: Funktion eines fluiden Muskels.[91]



(a) Weiches intelligentes Material



(b) Durch Stromzufuhr erhärtetes intelligentes Material

Abbildung 27: Von Hugh Herr präsentiertes intelligentes Material. [87]

Unklar ist jedoch, ob die Stromzufuhr die Härte verändert und somit auch, eingebaut in einem Handschuh, die Simulation von weichen Objekten ermöglichen kann. Allerdings kann davon ausgegangen werden, dass sich dieses intelligente Material nicht binär in seinen Härtegraden verhält. In einem Patent aus dem Jahr 2005 von *SRI International* in Californien, welches nicht eindeutig dem angesprochenen Material von Hugh Merr zugeordnet werden kann, allerdings für den selben Zweck patentiert wurde, wird allerdings über eine variable Härte berichtet.[86] Artikel, die dieses Material in einem Motion Capture Handschuh nutzen sind nicht vorhanden. Ebenso ist die Verzögerung der Wechsel der Härtegrade unbekannt.

#### 4.2.4.3 Magnetorheologische Flüssigkeit

Mithilfe von magnetorheologische Flüssigkeiten (MF), welche sich durch ein magnetisches Feld erhärten, wurden schon einige Tests im Kontext des haptischen Feedbacks gemacht. MF besteht aus einem Gemisch aus syntethischen Ölen (Polyalphaolefine), Mineralölen und Siliconölen. Hinzugegeben werden Eisenpartikel, die auch in Nahrungsergänzungsmitteln wie Eisentabletten enthalten sind und für den Menschen ungiftig sind. Gewerbliche MF nutzen allerdings Carbonyleisen.[44] Im eigenen Haushalt kann es aber auch mit Olivenöl und Maisstärke in schwächerer Form reproduziert werden. Ohne magnetisches Feld können sich die Eisenpartikel frei in der Flüssigkeit bewegen. Durch das Hinzufügen eines magnetischen Feldes reihen sich anschließend die Eisenpartikel ein, wodurch sich die Flüssigkeit erhärtet.[33] Dies wird in Abb. 28 dargestellt. MF wurde 1988 von Dr. Frank E. Filisko entdeckt.[120]

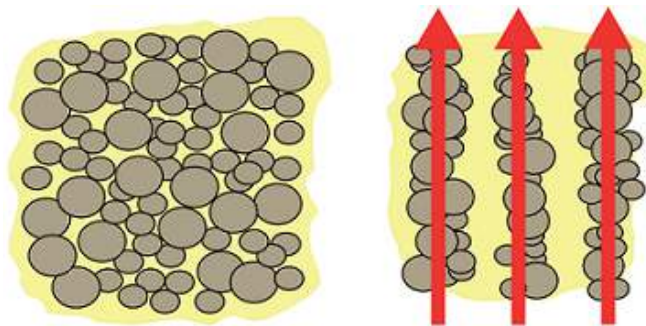
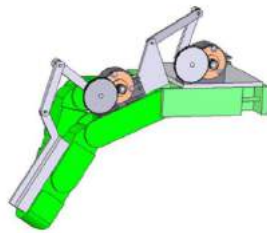


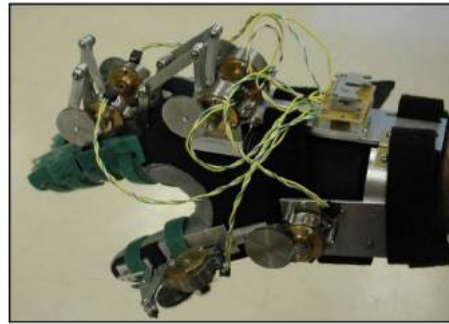
Abbildung 28: Eisenpartikel in MF, wenn (wie rechts) ein magnetisches Feld vorhanden ist. [33]

2006 wurde ein Paper veröffentlicht über das haptische Feedback mit MF in einem Lenkrad, um beim Fahren in einer Simulation das Lenken aus der Fahrbahn hinaus zu blockieren. Anzumerken ist die geringe Spannung von 0.75 Ampers, die für die Erhärtung nötig ist.[111] Ein weiteres Paper befasst sich mit einem 2D Joystick mit MF zur Steuerung von Anwendungen. Dort wird von einer variablen Stärke durch MF von 50 Newtonzentimeter (Ncm) bis 1000 Ncm berichtet. Genutzt wurden hierfür weniger als 2.5 V.[148] Dadurch wird deutlich, dass bereits wenig Strom für eine starke Erhärtung von MF vonnöten ist, was den Einbau in einen kabellosen Handschuh ermöglicht.

2009 veröffentlichten Jonathan Blake und Hakan B. Gurocak ein Paper, in dem sie einen haptischen Handschuh mit MF präsentierten. Sie entwickelten dabei eine kleine 84g schwere Fingerbremse, welche 899 Newtonmillimeter (Nmm) Drehmoment abbremmen kann. Zudem ist der Handschuh durch diese Bauweise autark und benötigt nicht, wie der *CyberGrasp*, eine externe Einheit die Kraft erzeugt. Der Handschuh wird in Abb. 29 dargestellt. In Tests wurden zwei Bremsen entwickelt, wovon die Stärkere mit 821 Nmm Drehmoment Abbremsstärke eine Verzögerung von 100



(a) 3D Visualisierung der MF Bremsen.



(b) Handschuh für 2 Finger

Abbildung 29: Der von Jonathan Blake und Hakan B. Gurocak erstellte MF Handschuh. [118]

Millisekunden (ms) hatte, die Kleinere mit 399 Nmm bot dagegen eine Verzögerung von 67 ms. Zudem wurde das Quetschen einer Spiralfeder erfolgreich simuliert, so dass die Härte der Feder durch das FF eingeschätzt werden konnte.[118] Dies bezeugt, dass eine Simulation von weichen Objekten im Kontext eines FF Handschuhs simuliert werden kann, da die Härte in einem hohen Grad variabel ist. Dadurch ist MF eine tatsächliche zukünftige Technologien für taktiles und genaues Force Feedback in Handschuhen.

### 4.3 Stand des Markts für Konsumenten

In der Tabelle 1 sind zum Kaufen verfügbare Motion Capture Handschuhe mit ihren Eigenschaften gelistet. Die Handschuhe, bei denen Preise nicht öffentlich verfügbar sind, sind üblicherweise nur für Unternehmer oder Entwickler gedacht. Einzig der *Plexus VR* und *LucidGloves* Handschuh sind in einem für Konsumenten attraktiven Preisbereich. Der Lucid Glove Handschuh muss allerdings selbst hergestellt werden, da er noch nicht vom Erfinder kommerziell verkauft wird. Der *LucidGlove* wurde in seiner vierten Version schließlich zu Evaluationszwecken sowie dem privaten Nutzen nachgebaut. Dessen niedrige Stabilität zeigte sich allerdings schnell, da er bereits in kürzester Zeit in seine Einzelteile zerfiel, wie in Abb. 30 deutlich zu erkennen ist. Zudem ist der Bau einer Sensoreinheit mit Spule durchaus unnötig komplex und frickelig. Es muss aber auch angemerkt werden, dass diese Probleme aus den niedrigen technischen Erfahrungen des Authors stammen können. Entsprechend muss dies nicht zwingend das Ergebnis eines schlechten Designs sein. Auch wird angemerkt, dass der Handschuh in der öffentlich Zugänglichen Version selbst vom Stand 08.03.2022 nur das Tracking des ganzen Fingers sowie dessen FF besitzt. Das Spreizen der Finger oder weitere Trackingpunkte sind noch nicht enthalten. Da der *LucidGlove*, abgesehen von der Software, für den Autor nicht ausgereift genug erscheint und sonst keine öffentlich

Zugänglichen DIY Handschuhe verfügbar sind, wird im weiteren Verlauf ein eigener Handschuh entwickelt. Der in Abb. 30 dargestellte *LucidGlove* diente bei seinem Bau zudem als erste Übung für nötige Fähigkeiten, die für das Designen eines eigenen Motion Capture Handschuh nötig sind.

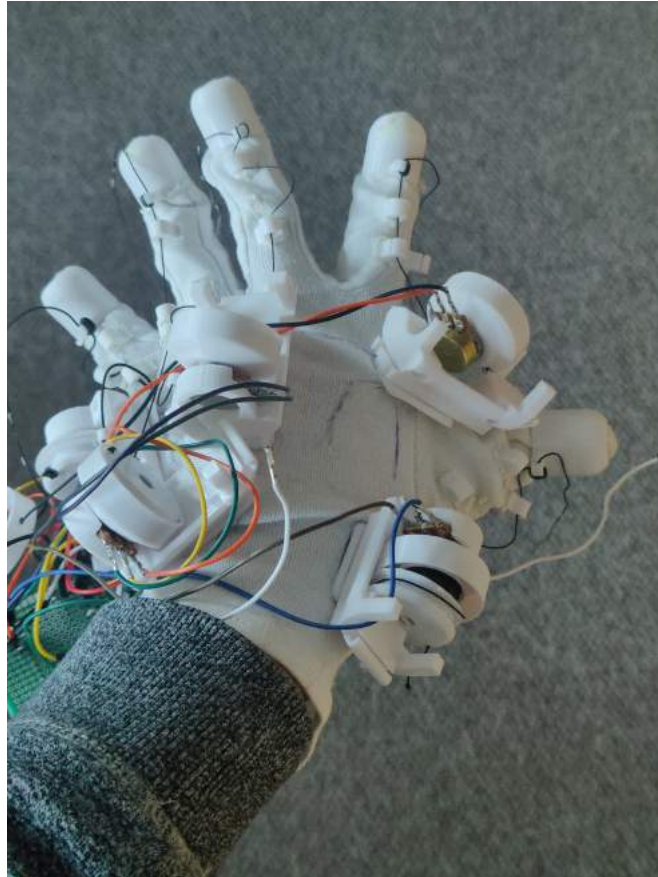


Abbildung 30: Abgenutzter *LucidGlove* in der vierten Version, nachdem Tests durchgeführt und Servomotoren abgebaut wurden.

Name	Tracking	Sensorenanzahl	Haptisches Feedback	Delay	Aktual.-rate pro Sekunde	Preis
HaptX[28]	Hall-Sensoren	N/A	Taktil und Force	<1/4 Sekunde	N/A	N/A
LucidGloves[107]	Potentiometer	5	Force	N/A	N/A	60\$
CyberGlove 3[18]	Flex Sensoren	18 oder 22	Force mit CyberGrasp	N/A	120	N/A
HI5 VR Glove[31]	IMU	7	Taktil	<5MS	180	999\$
Plexus VR[57]	Flex Senoren	5	Taktil	N/A	180	249\$
Rokoko Smart Glove[60]	IMU	7	Nein	N/A	100	1250€
Manus Prime X Haptic[45]	IMU und Flex Sensor	N/A	Taktil	N/A	N/A	3999€
AiQ Cobra Glove[14]	Flex Sensoren	7, 13 oder 16	Nein	N/A	N/A	N/A
Senso Glove DK3[73]	IMU	8	Taktil	15ms	400	999\$
Sensorial XR Glove[83]	IMU	7	Taktil	N/A	N/A	1500€
Dexmo DK1[74]	N/A	N/A	Force	N/A	N/A	N/A

Tabelle 1: Tabelle erwerbbarer Handschuhe.



## 5 Design

Für den Bau eines eigenen Handtracking Handschuhs mussten zunächst einige Parameter bestimmt werden. Da es keine finanzielle Unterstützung Dritter gab, mussten günstige Alternativen für das Prototyping sowie für benötigte Sensoren gefunden werden. Zudem sind nur die von der Hochschule Furtwangen vorhandenen und zugelassenen Maschinen nutzbar.

Das Design soll die Bewegungen der Finger wirkungsvoll und genau messen können und dennoch einen angenehmen Tragekomfort besitzen. Das Ziel ist einen Handschuh zu designen, der als Paar weniger als 90€ kostet, einfach zu reparieren und leicht im Gewicht ist sowie eine nahezu identische Trackingqualität wie kommerzielle Handschuhe besitzt. Für den Bau des Handschuhs ist zwingend ein Lötkolben, Lötzinn und Kabel für das Verbinden von Sensoren, sowie ein 3D-Drucker mit Thermoplastik wie PLA für das Drucken von Objekten, nötig.

In den folgenden Kapiteln wird das Design sowie der Vorgang des Handtrackens in seinen größeren Iterationen beschrieben und dargestellt. Detailanpassungen, sowie Versionen, deren Konzepte in keinen anderen Versionen übersetzt wurden, werden hier ausgelassen.

### 5.1 Konstruktion

Um erstellte Designs testen zu können, wurde eine zeit- und kostengünstige Möglichkeit gesucht, mit der die 3D-Modelle materialisiert werden könnten. Handwerkliche Möglichkeiten wären mit Materialien wie Holz oder Metall zu arbeiten. Diese würde man sägen, lasern und schleifen, um die gewünschten Formen zu erstellen. Allerdings sind diese Arbeiten recht zeitintensive und je nach Metall oder Holzart, auch sehr teuer. Metall könnte zwar gegossen werden, dafür sind allerdings vorgefertigte Formen nötig, die zudem einen höheren Schmelzpunkt haben als die Metalle, die darin geschmolzen werden. Dies bietet zudem ein hohes Gefahrenpotential. Aus diesem Grund wurde die Technik des 3D-Drucks in Betracht gezogen. Bei 3D-Druck wird ein Material vertikal in eine gewünschte Form gedruckt. In den 3D-Drucker-Typen Fused Deposition Modeling (FDM) wird das Filament an der Spitze erhitzt, sodass es weicher wird, wodurch es aus einem Loch als Faden hinaus gedrückt wird und auf einer

Druckplatte erhärtet. Nachdem der Drucker eine Ebene gedruckt hat, wird die zweite Ebene auf die erste Ebene aufgedruckt, bis alle Ebenen gedruckt und das Objekt fertig ist. Die direkte Alternative zu FDM Druckern sind Drucker, die ein Material mit bestimmtem Licht erhärten. So tauchen z.B. Stereolithografie (SLA) Drucker ihre Druckplatte in ein flüssiges Material, woraufhin gezielt Licht in das Material geleuchtet wird, welches das Material langsam erhärten lässt. Anschließend fährt die Druckerplatte langsam nach oben, während neue und alte Ebenen weiter mit Licht beleuchtet werden, um sie weiter zu härten. Da das Material vom SLA Druckern vorsichtig benutzt werden muss, da es giftig für die Haut sein kann und nach dem Druck gewaschen werden muss, wurde sich für die Materialisierung der 3D-Objekte für einen FDM Drucker entschieden. Da PLA günstig ist, als anfängerfreundlich gilt und zu den härteren Thermoplasten gehört, wurde sich dafür entschieden.

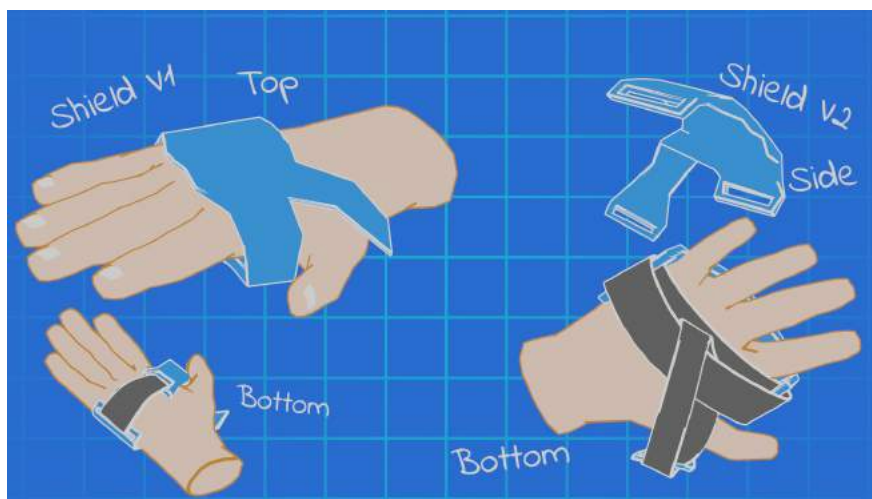


Abbildung 31: Erste Designs des Shields.

## 5.2 Das Shield

Da der Handschuh möglichst gering die Bewegungen des Nutzers einschränken sollte, mussten hierfür funktionelle Lösungen gefunden werden. Dies beginnt bei der Erstellung eines für die Hand angepassten Objekts, das über den Rücken der Hand läuft. Dieses Objekt dient der Befestigung von Sensoren und Motoren, um das Tracken der Bewegungen der Finger überhaupt zu ermöglichen. Hierfür wurde mit einer Kamera die Hand des Nutzers aus verschiedenen Richtungen fotografiert, um mithilfe von Photogrammetrie eine 3D-Visualisierung der Hand und Finger zu erhalten. Anschließend wurde die Handbreite und Länge in Zentimeter gemessen, um die Hand im 3D-Programm Blender in die korrekte Größe zu skalieren. Das 3D-Model der Hand dient somit als erster Anhaltspunkt für ein ergonomisches Design der Halterungen für Sensoren und Motoren, im weiteren Kontext *Shield* genannt. Die 3D-Hand ermöglicht

somit die Modellierung des Shields entlang des Handrückens, mitsamt freiem Platz für Knochen, die an der Hand durch diverse Bewegungen herausstechen können. Um den Shield an der Hand befestigen zu können, wurden wie in Abb. 31 dargestellt in der ersten Version Schlaufen an zwei Seiten des Handschuhs erstellt, um mit Materialien wie einem Klettverschluss, das Shield um die Hand herum zu binden und somit zu befestigen. Da das Shield sich allerdings beim Bilden einer Faust bewegt, wurde in einer zweiten Version eine dritte Schlaufe gebildet, wodurch das Shield von drei Seiten an der Hand befestigt wird und somit weniger anfällig durch Bewegungen ist.

### 5.3 Fingertracking

Da sich der Finger in einer gesunden Hand nicht nur nach unten und oben, sondern auch nach links und rechts bewegen kann, müssen Sensoren beide Bewegungen messen können. Hierfür musste ein Design gefunden werden, welches zwei Sensoren am selben Ort, einer für vertikale, einer für horizontale Bewegungen, messen kann. Zudem bewegt sich das Endglied, wenn es nicht durch ein Objekt aufgehalten wird, immer abhängig des Mittelglieds, wodurch nur eines der beiden Glieder getrackt werden muss, um auch die Rotation des anderen Glieds zu besitzen. Da die Bewegungen der Knochen abhängig von den Gelenken sind, können somit verschiedene Konzepte von Drehmessungen ausprobiert werden:

1. Dreh-Potentiometer, die Drehungen messen. Diese besitzen einen Stift, der bei der Drehung einen Widerstand verstärkt oder mindert, welche für die Bestimmung der Drehung genutzt werden kann.
2. Lineare Potentiometer, die ähnlich wie die Dreh-Potentiometer einen Widerstand verstärkt und mindert, abhängig von einer linearen Bewegungen in eine Richtung.
3. Hall-Sensoren, die magnetische Felder messen. Damit kann ein Abstand zu magnetisierten metallischen Objekten gemessen werden.
4. Dehnungsmessstreifen (oder auch Flex Sensoren) sind Streifen, die Winkelrotationen auf den Flächen, an denen sie angebracht sind, messen können. Diese sind allerdings nicht für schnelle und spitze Bewegungen geschaffen, wodurch sie im Nutzungskontext für Finger eine sehr hohe Abnutzung aufweisen. In eigenen Tests haben sich die Dehnungsmessstreifen schnell abgenutzt.

Die Sensoren werden an einem ESP32 angeschlossen, der in Kapitel 6 genauer erklärt wird. Ein ESP32 ist eine kleine Computereinheit, die Daten aus Sensoren lesen und verwenden kann. Für die Nutzung von linearen Potentiometer müssen, um drehende

Bewegungen messen zu können, die Bewegungen übersetzt werden. Ein Faden kann dafür genutzt werden. Lineare Potentiometer müssen technisch bedingt mindestens so lange sein wie die Bewegungen, die gemessen werden sollen. Aufgrund dieser technischen Einschränkung wurde die Möglichkeit nicht weiter verfolgt.

Hall-Sensoren sind sehr kleine Sensoren, die nur einen starken Magneten benötigen um Abstände zu messen. Allerdings ist die Magnetkraft nicht mit der Entfernung linear. Das bedeutet, dass die Verdoppelung der Distanz vom Hall-Sensor zum Magneten nicht den gemessenen Wert halbiert. Hierfür ist eine logarithmische Anpassung der Werte nötig, um eine Rotation in nützliche Werte zu messen. Die magnetische Flussdichte eines Zylindermagneten wird folgend berechnet.

$$B = \frac{B_r}{2} \left( \frac{D + z}{\sqrt{R^2 + (D + z)^2}} - \frac{z}{\sqrt{R^2 + z^2}} \right) \quad (5.1)$$

$B_r$  steht für das Remanenzfeld, welches von den physikalischen Gegebenheiten des Magneten abhängig ist.  $z$  ist der Abstand des Objekts, an dem die magnetische Kraft gemessen wird.  $D$  entspricht der Dicke des Magnets und  $R$  für den Radius. Die Maßeinheit der Werte sind identisch. Diese Formel kann auf  $z$ , also die Distanz, umgerechnet werden, um schließlich gemessene Werte eines Hall-Sensors in lineare Werte umrechnen zu können. Die berechnete magnetische Flussdichte wird allerdings schwächer, je weniger parallel das Objekt zur Magnetoberfläche liegt. Um nun mit einem Hall-Sensor die Bewegung des Finger nachvollziehen zu können, muss an den Finger ein Magnet angebracht werden. Da allerdings in den Fingerbewegungen, wie in Abb. 32 zu erkennen ist, der Magnet seine Rotation und somit sein Winkel zum Hall-Sensor dauernd verändert, fluktuert ebenso der magnetische Fluss durch die Rotation und Entfernung. Zudem wurden die Werte, die vom ESP32 ausgegeben werden, empirisch untersucht. Dabei wurden mehrmals Werte in 2mm Schritten ermittelt, in eine Liste eingetragen sowie anschließend analysiert. Die Ergebnisse waren dabei nicht qualitativ ausgeglichen genug, um eine logarithmische Kurve zur linearen Gegenrechnung der Fingerbewegung bestimmen zu können. Ein Grund hierfür kann eine zu hohe Ungenauigkeit der Hall-Sensoren sein.

Auch ist die Qualität der Daten nicht immer gleich. Zwischen dem Geradehalten des Fingers und einer kleinen Bewegung zum Hall-Sensor gehen die gemessenen Daten im Rauschen unter, sodass dazwischen nicht unterschieden werden kann. Eine Filterung durch einen Komplementär- oder Kalman Filter ist nicht möglich, da pro Finger nur eine Information vorhanden ist. Auch mit einem Hoch- und Tiefpassfilter, die für das Filtern von eindimensionalen Daten funktioniert, waren die Daten nicht aussagekräftig genug. Ein Hoch- und Tiefpassfilter funktioniert dadurch, dass aus einer Anzahl

der letzten gemessenen Daten die wahre, nicht vom Rauschen veränderte Information liegen muss. Daher werden die letzten Daten, ähnlich wie beim der ersten Hälfte eines Intervalls einer Sinuskurve, bewertet, wodurch die Gewichtung der äußeren Daten niedriger ist als die in der Mitte. So gesehen handelt es sich um ein gewichteter Durchschnitt der letzten Sensordaten. Auch eine Filterung die den Durchschnitt oder Median der letzten paar gemessenen Daten ausgibt führte zu keiner nutzbaren Verbesserung. Aus diesen Gründen werden Hall-Sensoren im weiteren Verlauf nicht verwendet. So wurden, auch aufgrund der zuerst logischen Schlussfolgerung, Rotationen mit Dreh-Messern zu messen, die Dreh-Potentiometer ausgewählt.

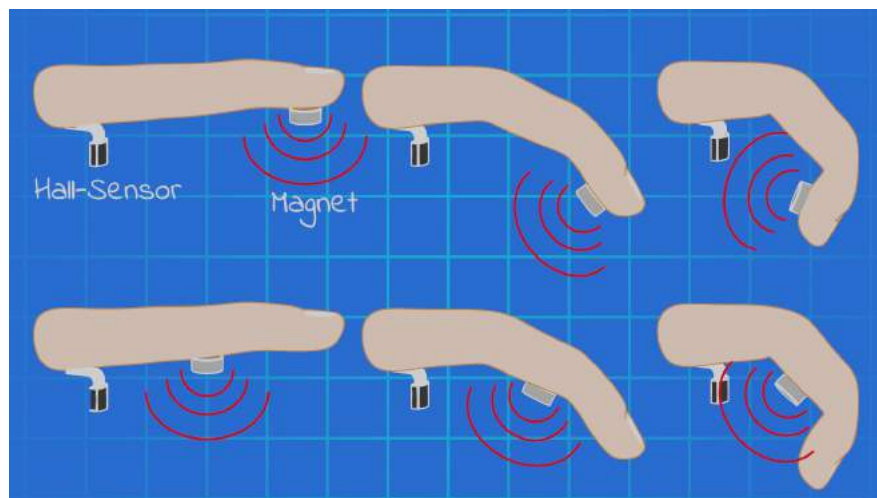


Abbildung 32: Funktion Hall-Sensor.



Abbildung 33: Ein Testversuch mit einem Hall-Sensor.

Aufgrund dessen, dass die Potentiometer fest an ein Objekt angebracht werden müssen, wurden Alpha Potentiometer [61] in Betracht gezogen, da diese neben der großen Verfügbarkeit in Fachgeschäften und Onlineshops, auch eine Einlassung haben, um Messungen von Drehungen von Objekten zu ermöglichen. Da es nötig ist den Potentiometer zu befestigen, gleichzeitig aber auch locker zu lassen um eine Drehung zu messen, ohne dass sich der Potentiometer selbst dreht, wurde ein Design entwickelt, in dem der Potentiometer in einer freien Fassung festgeschraubt werden kann. Zudem wurde eine Fassung gebaut, in der die Achse des Potentiometers eingeführt werden kann. Wie in Abb. 34 zu erkennen ist, wird in der grünen Scheibe eine Mutter eingeführt, in der ein Potentiometer dann eingeschraubt werden kann. Die grüne Scheibe ist lose im Potentiometer-Halterung miteingedrückt und wird mit der Öffnung an einer Seite für das Einschrauben zugänglich.

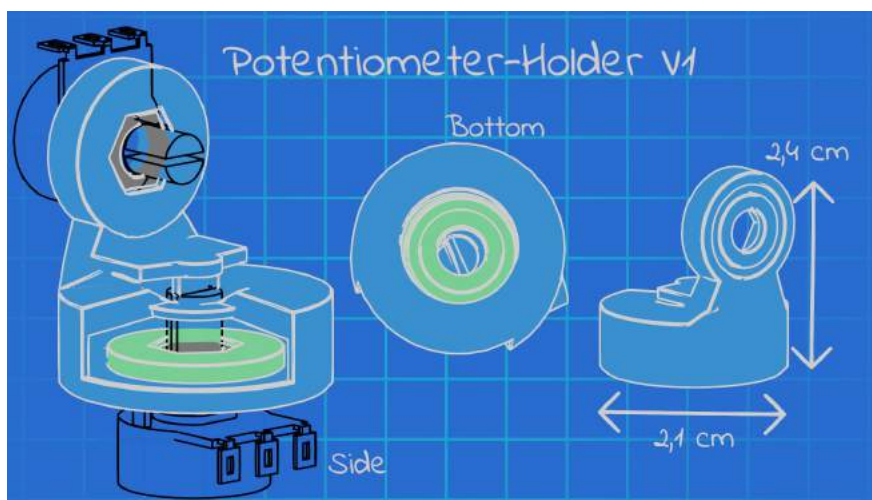


Abbildung 34: Erste Version des Potentiometer-Halterung (blau) mit Feststellschraube (grün) für die Mutter (grau) zum Einschrauben des Alpha Potentiometers.

Auf der oberen Seite ist zudem ebenso Platz um einen zweiten Potentiometer einzudrehen. Somit kann, wenn die *Potentiometer-Halterung* über dem Handgelenk angesetzt wird, das Spreizen und die vertikale Bewegungen der Grundglieder gemessen werden. Da allerdings die Alpha zu groß sind, um die vertikale Bewegungen messen zu können, da sie sich auf der Handoberfläche gegenseitig stören, mussten andere Sensoren gefunden werden. Die Bourns [3] sind Potentiometer, die keine Achse besitzen. Aufgrund dessen können diese in kleine Produkte miteingearbeitet werden, da eine eigene Millimeter kleine Achse eingesetzt werden kann. Um diese Potentiometer in die *Potentiometer-Halterung* nach dem 3D-Druck festzuhalten, muss er eingeklemmt werden. Dafür wurden die Maße der Potentiometers entnommen und, wie in Abb. 35 zu erkennen, eine Fassung gebaut, die mit einem Socket zusammen den Potentiometer, ohne großen Bewegungsspielraum, festklemmt. Der nun gewonnene Platz hat

zudem das Potential an den Potentiometer-Halterung einen kleinen und stromsparenden SG90 Servomotor [63] zu befestigen. Der Servomotor soll als Motor dienen, der gegen die Finger arbeitet, wenn Objekte gegriffen werden. Dafür wurden zwei Blöcke angebaut, die mit ihren Löchern das Einschrauben des Motors ermöglichen.

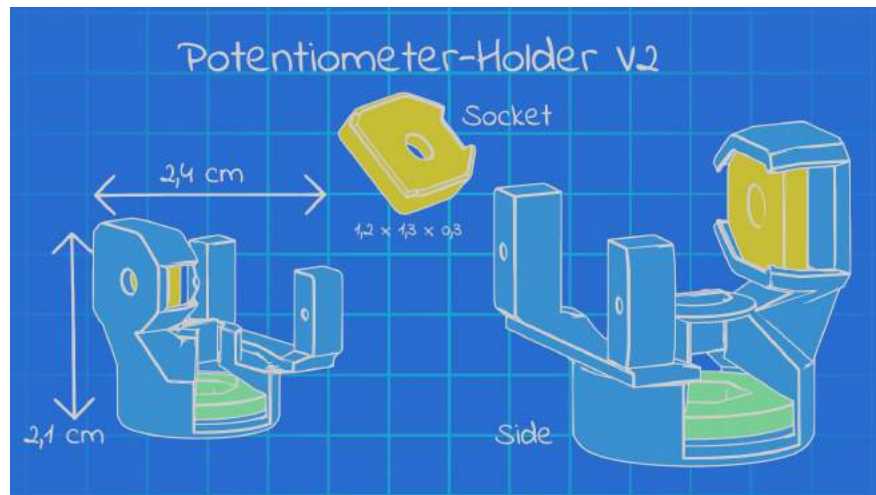


Abbildung 35: Zweite Version des Potentiometer-Halterung inkl. Socket (gelb).

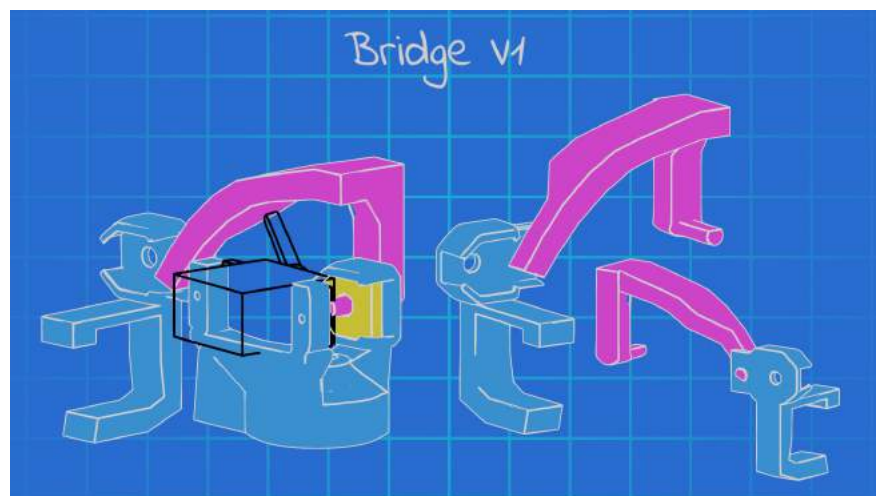


Abbildung 36: Erste Version der Brücke (lila) zum Finger.

Für erste Tests des Designs wurde anschließend eine erste Halterung für den Finger gebaut, die durch den Finger bewegt wird. Zudem musste eine Brücke zwischen dem Potentiometer-Halterung und dem Finger erstellt werden, siehe Abb. 36. Die *Brücke* besitzt eine Achse, der durch den kleinen Potentiometer und dazugehörigen Socket durch geht, der die Rotation der *Brücke* auf den Potentiometer anwendet, wodurch die Drehung gemessen werden kann. Zudem ist die *Brücke* zunächst lose mit dem Fingerteil verbunden, in dem ebenso ein kleiner Potentiometer für das Messen der Drehungen des Mittellieds einsetzbar ist. Die *Brücke* wurde so gebaut, dass genug Platz für den Servo Motor vorhanden ist, dessen drehender Flügel die Brücke beim



Anschlag aufhalten sollte. Allerdings war die Kraft und die Fläche des Flügels des Servos in allen Iterationen zu klein, wodurch der Servomotor nicht korrekt Bewegungen blockieren konnte.

Da sich die Brücke bei Bewegungen des Fingers stets am Grundglied entlang hoch und runter bewegt hat, weil sich die Brücke aufgrund der Starrheit nicht verlängert oder verkürzt, musste eine dynamische Möglichkeit gebaut werden, damit die Brücke an der selben Position entlang des Grundglieds, über alle Bewegungen hinweg, bleibt. Hierfür wurde in der zweiten Version ein Bauteil entworfen, welches die Brücke um einen kleinen Weg erweitert, der sich um den Bourns Sensor um 180 Grad drehen kann, wie in Abb. 37 links an dem orangenen Elementen zu sehen ist. Da dieses System sich als wirkungsvoll präsentierte, wurde anhand dessen eine Klammer gebaut, die den Bourns Sensor mit zwei Teilen für eine bessere Stabilität umklammert. Ein neue prototypische Verbindung der Klemme zum Grundglied zeigte sein Potential, worauf hin die neuen Klammern und die Brücke weiter verbessert wurden. Löcher in den Klammern werden zudem benötigt, um mit einem Servomotor die Bewegung einzuschränken.

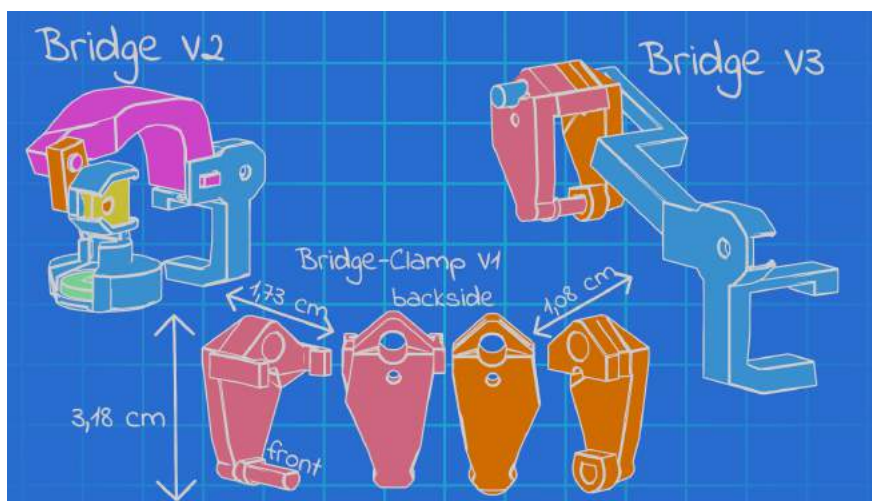


Abbildung 37: Zweite und dritte Version der Brücke (pink und orange) zum Finger.

Mithilfe der Löcher kann, wie in Abb. 38 zu erkennen ist, ein SG 90 Servomotor die Drehung der Brücke stoppen. So wird ein Faden durch den Flügel des Motors gezogen, der auch durch die Löcher der Klammern läuft. Die Drehwinkel des Motors geben so Spannung auf die Klammern. Bei einer Ausrichtung von 0 Grad, bei der das Ende des Flügels in die Richtung des Potentiometer-Halterung blickt, ist freies Spiel möglich. Je weiter der Motor sich dreht, desto mehr Kraft wird auf den Faden und somit auch auf die Klemmen ausgeübt, die so zurückgezogen werden. Wenn die Löcher allerdings mittig in den Klemmen eingesetzt werden, kann die Zugkraft des Motors die Klemme blockieren, wenn die Klemmen zu weit rotiert sind. Dies kann verhindert werden, indem die Länge des Fadens so bestimmt wird, dass der Winkel der Klemmen nicht



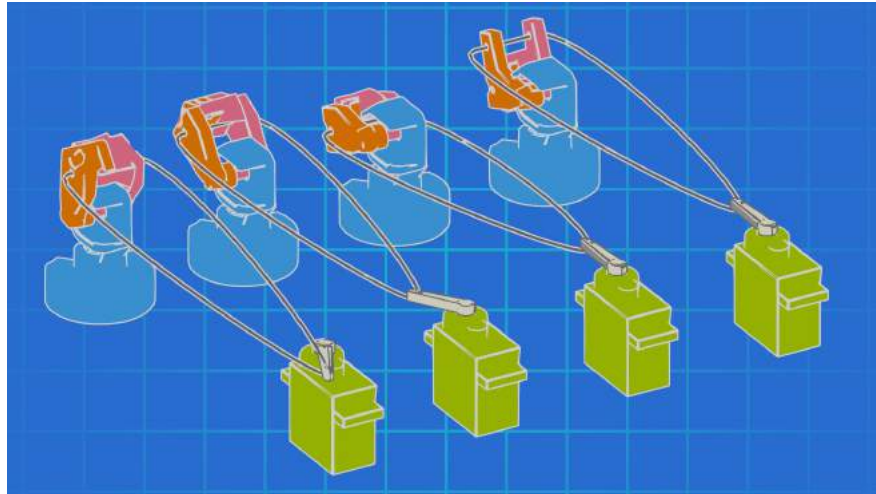


Abbildung 38: Haptische Blockade durch einen SG90 Servomotor (grün). Der dritte Motor kann die Klemme nicht mehr nach hinten bewegen, da die Klemme versucht sich nach unten zum Motor zu rotieren. Das Design der Klemme ganz rechts verhindert dies.

hoch genug sein kann, um diese zu blockieren. Um dem entgegenzuwirken, wurden die Klemmen nach hinten ausgearbeitet und dort Löcher platziert. Dies erhöht zudem den Radius der Drehachse, welches die Stärke des Motors durch die Drehachse nach dem zweiten newtonsche Gesetz für die Drehbewegung verstärkt, wenn der Finger stark gebogen wird. Die Kraft des Motors wirkt am stärksten, wenn der Flügel einen rechten Winkel zur Brücke besitzt. Entsprechend besitzt der Motor am meisten Kraft, wenn das Grundglied des Fingers um 45 Grad nach unten gedreht wird. So ist die Kraft des Motors ideal ausgeglichen.

Das neue Konzept der neuen Klammern wurde, wie in Abb. 39 links veranschaulicht wird, um das Design einer neuen Brücke zum Grundglied des Fingers zu unterstützen, in seine dritte Version umgearbeitet. Schließlich wurden, um eine stabile Halterung zu ermöglichen, Löcher in den Klammern und der Brücke gesetzt, um diese mit einer Schraube und zwei Muttern festmachen zu können. Die zweite Mutter dient als Kontermutter, wodurch sich Schrauben nicht selbst aus dem Gewinde drehen können. Die Brücke kann nun auch über einen Klettverschluss mit dem Finger verbunden werden. Anschließend wurden die bisherigen Konzepte auch auf eine Klammer inklusive Brücke für das Mittelglied angewendet. Eine Erweiterung auf das Endglied der Finger wäre so auch möglich. Dadurch war ein erstes komplettes Konzept für das Tracken von Fingern erarbeitet.

Um das Fingertracking-Konzept zu testen, wurden die Teile für jeden Finger, teilweise in seiner Größe abhängig zum Finger, mit PLA Filament mit einem Ender 3 V2 gedruckt und zusammen gebaut. Das Shield wurde, um als Halterung der Sensoren und Servomotoren zu dienen, umgearbeitet. Zudem wurden Öffnungen und Leitungen

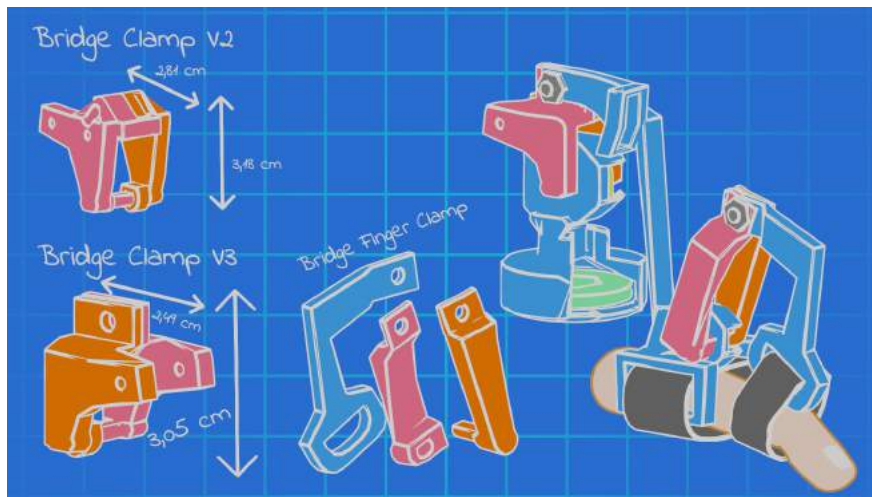
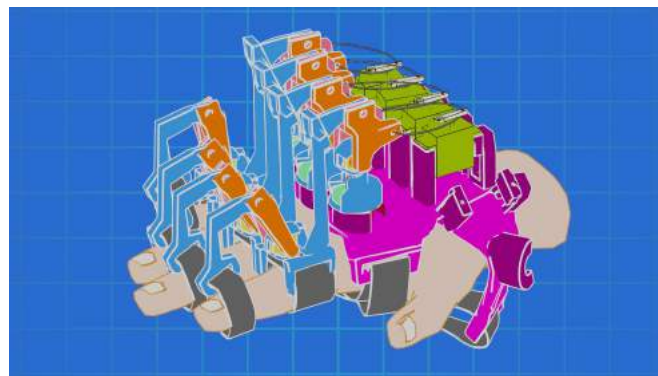


Abbildung 39: Erstes vollständiges Fingertracking Konzept. Brücke (blau) sowie Klemmen (pink, orange) wurden für das Mittelglied adaptiert.



(a) Test des vollständigen Konzepts auf einem Finger.



(b) Visualisierung für Zeige- bis kleinen Finger.

Abbildung 40: Erster Handtracking Prototyp.

für Kabel bedacht, um Kabel geordnet zum ESP32 leiten zu können. Für Tests können die unteren Alpha einfach in eine Halterung eingeschoben werden, die für Tests für ein Proof of Concept ausreichen. Die Servomotoren werden an das Shield geschraubt.

#### 5.4 Bewertung des ersten Designs

Der Handschuh wurde anschließend, ohne Nutzung der Daten der Sensoren, ausschließlich auf seine Bequemlichkeit und Haltbarkeit qualitativ zwei Stunden lang bei alltäglichen Bewegungen und Griffen in der Wohnung getestet. Die Servomotoren rotierten dauerhaft in verschiedene Winkel, um das Feedback in verschiedenen Fingerpositionen und -kräften zu testen. Dabei sind folgende Dinge aufgefallen:

1. Die Brücken für das Grund- und Mittelglied der Finger drücken sich beim Bilden einer Faust auf die Haut. Dadurch wird bei manchen Fingern nur unter geringen Schmerzen bei Grundgliedern eine Faust möglich.
2. Durch die Blockade der Brücken biegt sich beim Bilden einer Faust insbesondere die Brücke zum Grundglied, wodurch sich zwar die Hand mehr schließen lässt, allerdings dann Kraft auf andere Teile wirkt und somit das Potential entsteht, dass dadurch Teile brechen. Zudem sind diese Bewegungen, die durch das Biegen der Brücke möglich sind, nicht messbar mit dem Potentiometer.
3. Der Klettverschluss, der für jeden Finger sowie für das Umschnallen des Shields genutzt wird, klettet sich fingerübergreifend an. Selbst ein Wechsel der Klett-richtung an den Fingern führte zu vereinzelt Verklebungen zwischen den Fingern beim Greifen von kleinen Objekten oder Bilden einer Faust. Das ist allerdings auch dem genutzten, relativ dicken Klettverschluss zu verschulden.
4. Die Klammer für die Brücke für das Mittelglied aus Abb. 39 ist mehrfach einfach zebrochen.
5. Der Faden, der vom Servomotor für den Daumen in die Brückenklemme führt, zieht die Brückenklemme in einem unvorteilhaften Winkel zurück. Dadurch kann die Kraft des Servomotors nicht ganz übertragen werden und kann zu einem Verschleiß einer Seite der Klemmen führen. Dadurch muss das System für das Feedback des Daumens umgearbeitet werden.

### 5.5 Weitere Iterationen des Designs

Um die oben genannten Probleme zu lösen, musste das Design in manchen Teilen umgearbeitet werden. Für die nächste Iteration wurde zunächst das erste Problem aus Kapitel 5.4 analysiert. Das Grundglied drückt auf den Finger, da die Brücke nur einen Rotationspunkt an den Brückenklemme besitzt. Somit musste ein weiterer Rotationspunkt am Grundglied her. Hierfür wurde die Brücke in zwei Teile geteilt. Einmal als Objekt, das den Sensor hält, und an das Grundglied befestigt wird und einmal als Brücke, welches nun nur als Verbindung dient. Diese zwei Teile werden dann mit einer Schraube verbunden. Da nun zwei Gelenke vorhanden sind, biegt sich die Brücke nicht mehr, wodurch eine Faust ohne Schmerzen möglich ist. Dadurch wurde auch das zweite Problem gelöst. Diese Lösung ist in Abb. 41 integriert.

Um den Klettverschluss entfernen zu können, muss eine neue Möglichkeit für das Befestigen an den Finger gefunden werden. Hierfür wird ein Ring erstellt, der das Grundglied umhüllt. Da die Brückenklemme für das Mittelglied als Problem festge-

stellt wurde und Iterationen davon keine Besserung zeigten, musste eine Alternative gefunden werden. Es wurde ein Ring für das Mittel- und Grundglied gestaltet, welcher durch eine direkte Brücke verbunden wird. Zunächst wurde versucht den Sensor in den Ring hineinzudrücken, wie in Abb. 41 in der Build-in Version der Ring Chain zu sehen ist.

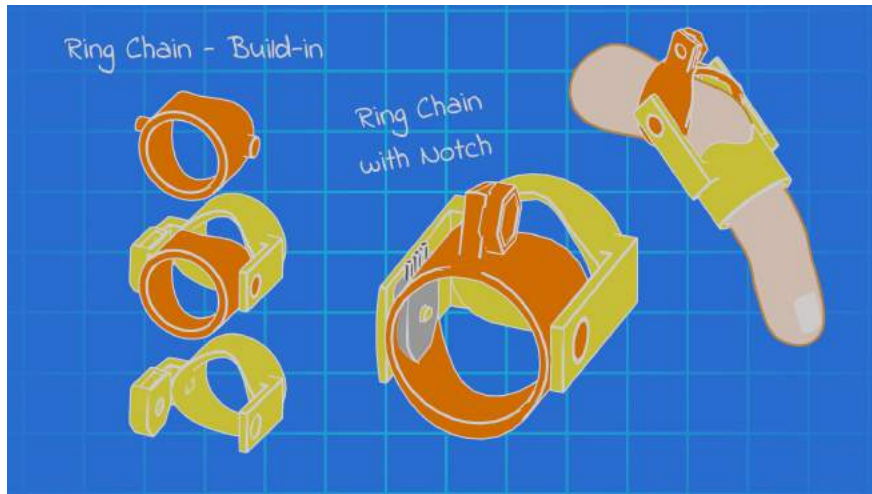
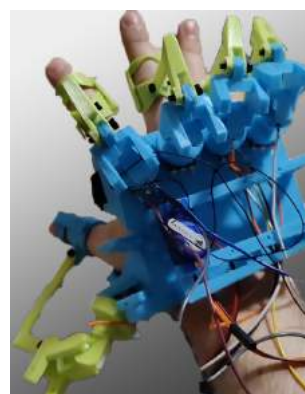


Abbildung 41: Entwicklung des Ring-Designs. Die ersten Versuche, Sensoren in das gelbe Objekt einzudrücken (links), bis diese nur noch festgehalten werden (rechts).

Beim Eindringen von Objekten ist gemeint, dass während des Druckvorgangs der Sensor eingelegt wird. Dies kann entweder durch das Pausieren des Drucks passieren, oder Objekte werden weit genug auf der Druckplatte verteilt, dass genug Zeit besteht, den Sensor während des Drucks eines anderen Objekts einzulegen. Somit drückt der 3D-Drucker über den Sensor, wodurch der Sensor in das Objekt fest eingebaut wird. Dies wird in Abb. 43 gezeigt. Dadurch können die Sensoren nach einem Bruch einfach entnommen werden. Ein Nachteil hierfür ist, dass der Druck beobachtet werden muss,



(a) Test des Ring Designs an einem Finger.



(b) Anwenden auf allen Fingern.

Abbildung 42: Produktion und Test des Ring Designs.



(a) Durchvorgang mit eingesetztem Sensor



(b) Prototyp des eingedruckten Sensors

Abbildung 43: Das Eindringen eines Sensors in ein Plastikstück.

da nur ein bestimmtes Zeitfenster besteht, in dem Sensoren eingefügt werden können. Alternativ kann der Code der Druckerdatei selbst bearbeitet werden, um eine Pause zu integrieren. Der Slicer *Cura* besitzt das Post Processing Plugin *Pause at height*, der den Code nach dem Slice automatisch bearbeitet. Zudem ist das nicht mit allen 3D-Druckern möglich. Im Falle des Ender 3 V2, der für das Prototyping benutzt wird und ein FDM Drucker ist, ist dies kein Problem, da die Druckplatte sich nur nach vorne und hinten und der Druckkopf sich nach oben und seitlich bewegt. Bei 3D-Druckern, bei denen z. B. die Druckplatte in der Höhe bewegt wird, könnte dies durch Platzmangel schwierig sein. Das Eindringen ist grundlegend aus diesem Grund bei 3D-Druckern, die Licht nutzen um Flüssigkeit auszuhärten, nicht möglich, da diese die Druckerplatte aus der Flüssigkeit nach oben bewegen, um 3D-Objekte zu erstellen.

Die äußeren Seiten der Ringe besitzen zudem Löcher und runde Aussparungen, die als Drehgelenk genutzt werden. Das Drehgelenk muss, damit dies funktioniert, auf der Länge des Drehgelenks der Knochen liegen. Um das Rotieren des Knochengelenks vollkommen zu ermöglichen, ohne dass Haut eingequetscht wird, wird oben und unten am Ring eine Kurve als Aussparung eingebaut. Dies bereitet Platz für Haut, die sich an den Fingern ansammelt, wenn der Finger bewegt wird. Da das Design allerdings zu dick ist um an mehreren Fingern angebracht zu werden, wurde eine Aussparung an einem Ring erstellt, worin der Sensor zwischengeklemmt wird. Das ist in Abb. 41 rechts zu sehen. Hier muss allerdings der Sensor festgeklebt werden. Da zudem die Teile schnell kaputt gehen, wurden Versuche der letzten Iterationen zu einem zusammengefügt.

Die Schwachstellen der bisherigen Versionen sind jeweils die Teile, die durch den Bourns Potentiometer durchlaufen, um diesen zum Rotieren zu bringen. Diese müssen dünn genug sein, um durch das Loch in der Bourns-Sensoren zu passen, allerdings auch dick genug sein, um Stabilität zu bieten und das Loch möglichst vollkommen



auszufüllen, um jede Bewegung messbar zu machen. Hierfür gibt es nur die Alternative andere Potentiometer zu nutzen als den Bourns. Je nach Fingergröße kann der Alpha genutzt werden, der eine stabile Metall-Achse besitzt. Demnach muss insbesondere der Ring- und Mittelfinger breit genug sein, damit ein Alpha-Potentiometer in gedrehter Stellung nicht herausragt, da sich ansonsten die Potentiometer gegenseitig stören. Für den Kleinen- und Zeigefinger könnten nämlich die Potentiometer versetzt angebracht werden. Da das Design allerdings grundsätzlich für alle Handgrößen funktionieren soll, müssen weiterhin Bourns genutzt werden. Demnach muss um das Problem herumdesignt werden.

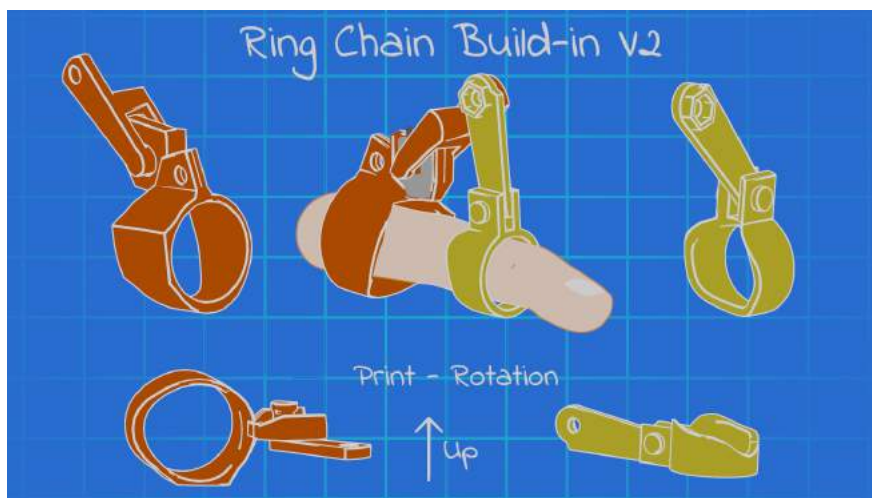


Abbildung 44: Zweite Version des Ring Designs, in der die Bourns-Potentiometer oben platziert werden (oben). Für das Eindrücken des Bourns-Potentiometer in das Stück für das Grundglied (orange) ist eine bestimmte Rotation beim 3D-Druck zu beachten (unten).

Zunächst wurde beschlossen, dass das Eindrücken der Sensoren weiterhin eine gute Möglichkeit ist die Sensoren zu integrieren. Da der Sensor wieder nach oben wandern muss, damit die Ringe nicht mehr zu breit sind, wurde eine Brücke beim Grundglied gebaut, welches den Potentiometer bereits einschließt. Diese Brücke, wie in Abb. 44 in orange zu erkennen ist, läuft in ein Loch, in dem der Bounce Potentiometer Platz findet, und verdickt sich anschließend, um den Potentiometer fest einzuklemmen. Der Ring für das Mittelglied besitzt den zweiten Teil der Brücke, welcher ebenso durch Löcher im Mittelring festhält und mit dem Ring zusammengedrückt wird. Dadurch fällt das einsetzen einer Schraube und Mutter weg. Entsprechend müssen die Objekte in einer eigenen Rotation gedruckt werden, wie in Abb. 44 unten dargestellt wird. Hierfür wurde der Ring für das Mittelglied an einer Seite flach designt, um den Druck zu ermöglichen. Da die Brücke des Grundglieds dadurch in der Luft steht, müssen Supports mitgedruckt werden, die schwebende Teile während des Drucks unterstützen. Ansonsten würde das Filament beim Druck einfach auf die Druckplatte fallen. Im liegenden Zustand des Rings für das Mittelglied liegt die Brücke flach mit dem

Ring auf der Druckfläche, um hier einen befestigten Druck zu ermöglichen. Damit wurde das Design der Brücke Finger Clamp aus Abb. 39 in zwei stabilere Druckteile zusammengefasst.

Da Ringe verwendet werden, die optimalerweise auf die Dicke der Finger angepasst werden sollten, müssen hierfür die 3D-Objekte jeweils vom Nutzer auf seine Größen angepasst werden. Um dies zu vereinfachen, werden die 3D-Objekte Modular freigegeben, um Ringe, unabhängig der Brücken, einfach personalisieren zu können. Ebenso ist hierdurch auch möglich nur das obere viertel eines Ringes auszudrucken und mit Klebstoff auf Handschuhe festzukleben. Hierzu werden Handschuhe mit festen Stoffen empfohlen, wie Leder- oder Motorradhandschuhe.

## 5.6 Letzte Reduzierungen des Designs

Da für die Sensoren bereits 45 Leitungen für die Stromversorgung und Datenübertragung nötig sind, die alle in ein Board laufen, ist eine Reduzierung der Kabeln nötig. Diese Entscheidung entstand auch dadurch, dass Wackelkontakte und versehentliches Lösen von Kabel schwer zu bemerken und finden war. Ebenso ist das Gesamtkonstrukt sehr groß und das Shield sehr dick. Auch das Anziehen der Ringe ohne Handschuh ist recht kompliziert. Aus diesem Grund wurde das Shield und die Kabelsetzung neu überdacht und umgesetzt. Die Ringe wurden für die neue Version eines tatsächlichen Motion Capture Handschuhs mit Force Feedback, wie vorher beschrieben, ebenso reduziert.

Zunächst wurde überlegt, inwiefern ein Shield überhaupt noch nötig ist, wenn die Komponenten an einen Handschuh direkt gesetzt werden können. Daher wurde nach einer Methode gesucht, um die Konstrukte fest, aber auch austauschbar, zu befestigen. In der bisherigen Version des Shields werden die Alpha nur eingeschoben und eingespannt. Dies führte allerdings oftmals dazu, dass diese mitsamt dem ganzen Konstrukt heraus fielen. Schließlich wurde zugunsten einer Schiebeleiste entschieden, in der die Komponenten ein- und ausgeschoben werden können. So eine Schiebeleiste ist in Abb. 45 in dunkelblau zu sehen. Auf einem Handschuh geklebt zeigte sich allerdings, dass der Alpha zu schwer ist, um direkt aufgeklebt zu werden. Aus diesem Grund wurde der Potentiometer-Halterung umgearbeitet, sodass Bourns, ähnlich wie bei Abb. 44, eingedrückt werden. Der obere Bourns Sensor kann in dieser Version des Potentiometer-Halterung ohne Socket, der in Abb. 35 erstmals dargestellt wurde, einfach eingeschoben werden. Das Gegenstück der Schiebeleiste wurde unten angebracht. Die neuen Designs sind in Abb. 45 abgebildet.

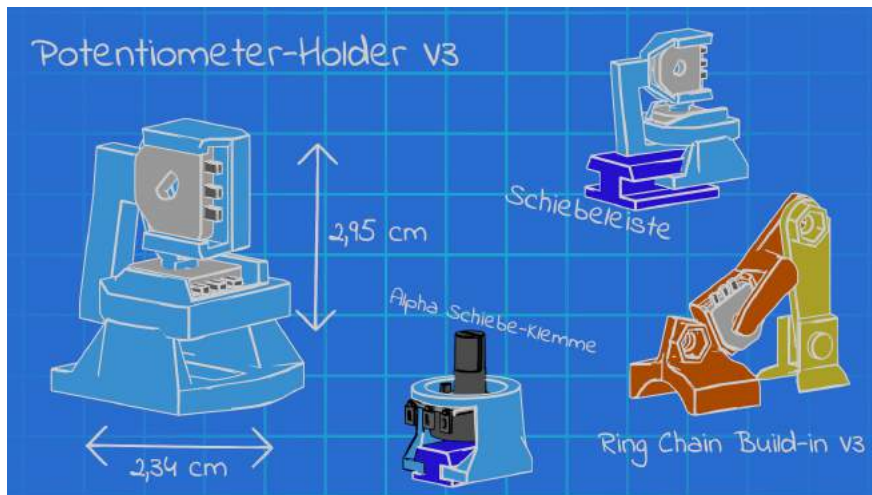


Abbildung 45: Die dunkelblaue Schiebeleiste ermöglicht das Einschieben von Objekten. Dadurch kann die Schiebeleiste überall verwendet und sogar, im Falle von zwei Bourns, direkt auf einen Handschuh geklebt werden.



Abbildung 46: Dünneres Shield mit Schiebeleisten. Es laufen Kupferkabel zur Stromversorgung entlang um Kabel am Board zu reduzieren.

Da allerdings die dreifache Nutzung von Bourns Sensoren den Preis des Handschuhs erhöht, wurde dennoch ein Shield gebaut, um auch weiterhin die Alpha Potentiometer zu unterstützen. Dieser ähnelt grundlegend der Form des bisherigen Shields, ist allerdings so dünn, dass er flach gedruckt werden kann, sich dennoch um den Handrücken biegen lässt ohne zu brechen und Schiebeleisten für die Anbringung von Motoren und Sensoren besitzt.

Um die Menge der Kabel zu reduzieren, wurden zusätzlich am Shield zwei offene Halbringe eingearbeitet. Hierdurch führt jeweils ein offenes Kupferkabel. Diese Kupferkabel werden an die Stromzufuhr des ESP32 am Board angeschlossen. Die Anschlüsse für + und - der Sensoren werden dann an die jeweiligen Kupferkabel in Parallelschaltung angelötet. Hierdurch kann die Anzahl von Kabel der Sensoren, die in das Board füh-



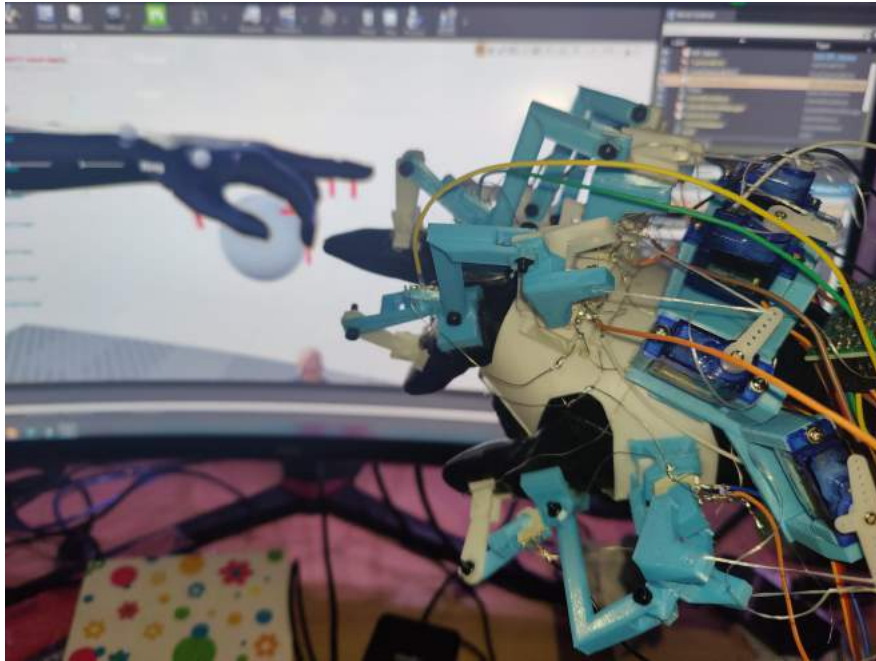


Abbildung 47: Funktionsfähiges Handschuhdesign mit an die Finger geklebten halben Ringen, der drei Bourns Sensoren nutzt. Motoren blockieren die Fingerbewegung.

ren, von 45 auf 17 reduziert werden. Ebenso ist die Suche nach Fehlerquellen dadurch einfacher, da nun entweder alle Sensoren ausfallen, weil kein Strom mehr läuft, oder nur ein Sensor ausfällt, weil das Datenkabel Probleme aufweist. Die Verbindung der Servomotoren sind dadurch nicht betroffen.



## 6 Implementierung

Um Sensordaten nutzen zu können, sowie Informationen für Servomotoren zu übertragen, mussten diese in einem Programm für einen ESP32 implementiert werden. Der ESP32 sollte zudem in Kontakt mit UE stehen und die Daten schnellstmöglich kabellos mit einer möglichst geringen Fehlerquote übertragen. In diesem Kapitel wird über Hardware, den Anschluss der Sensoren und Servomotoren, der Übertragung der Daten sowie Verarbeitung und Implementierung auf ein Rig eingegangen.

### 6.1 Hardware

Um alle Potentiometer sowie Servomotoren nutzen zu können, braucht es eine Proessoreinheit, welche genügend Anschlüsse und Prozessorleistung besitzt. Boards wie das *ESP32* oder ein *Arduino Uno* eignen sich hierfür, da diese Boards Microcontroller beinhalten. Microcontroller sind kleine Recheneinheiten, die ohne viel Strom Millionen von Kommandos in der Sekunde ausführen können. Der *Arduino Uno* verbindet diese Microcontroller auf einem Board mit vielen hilfreichen Komponenten, um die Arbeit mit ihnen zu vereinfachen. So besitzt ein *Arduino Uno* unter anderem ein USB-Anschluss, mit dem der gewünschte Programmcode von einem PC über USB auf den Microcontroller übertragen werden kann. Im Falle der *ESP32* sind sogar Module auf den Boards eingebaut, die das Nutzen von kabellosen Übertragungstechnologien ermöglichen. Um Bluetooth an einem *Arduino Uno* nutzen zu können, muss hierfür ein Bluetooth-Modul per Kabel an den *Arduino Uno* angeschlossen werden. Das *ESP32 Dev-Module* hat bereits Bluetooth und Wifi als Technologien auf dem Board verbaut. Die Module von *Arduino*, die in vielen verschiedenen Größen hergestellt werden, bieten in der Version *Arduino Uno* eine große Auswahl von Anschlüssen und mithilfe des ATmega328P Controllers mit bis zu 20 Megahertz (MHz) genug Leistung um kommende Aufgaben zu bewältigen.[88] Allerdings, da ein *ESP32 WROOM-32E DevModule* neben einem Xtensa 32-bit Doppelkern-Prozessor mit bis zu 240 MHz auch Module für Bluetooth und WLAN, sowie eine kompaktere Größe und viele Anschlüsse besitzt, wurde sich für solch einen entschieden.[98] (S. 2) Damit sind Möglichkeiten der Skalierung während des Prototypings größer, sowie die Herstellung einer kabellosen Verbindung ohne extra Module möglich. Zudem kann der ESP32 mithilfe der *Arduino Integrated Development Environment (IDE)* programmiert werden, wie als würde es

sich um ein Arduino Module handeln. Nur wenige zusätzliche Arduino-Bibliotheken oder Methoden werden nicht von ESP32 Modulen unterstützt.

Um die Winkelstellung eines Potentiometers auslesen zu können, muss dafür ein Stromkreis geschlossen sein. Dafür wird einerseits der Minus- und andererseits der Pluspol angeschlossen. Der mittlere Anschluss ist bei beiden genutzten Potentiometer jeweils der Ausgang, der die übrige Ausgangsspannung ausgibt. Die Ausgangsspannung wird durch das Drehen des Widerstands am Potentiometer verändert. Damit dient die Rotation beim Potentiometer als Spannungsteilwert, der in einem Microcontroller mit Werten von 0 bis 4095 angegeben wird.[168] Die SG90 Servomotoren werden ebenso über jeweils einen Anschluss für Plus- und einen Minuspol mit Strom versorgt. Der dritte Anschluss, ein Eingangsanschluss, dient der Übertragung von Daten, um die gewünschte Rotationsrichtung dem Servomotor zu senden. Der Anschluss der fünf Servomotoren ist nicht direkt über den ESP32 möglich. Die Stromaufnahme der Servomotoren ist höher als die Stromleistung, die der ESP32 ausgeben kann. Aus diesem Grund fallen die Servomotoren, wenn sie angesteuert werden, kurzzeitig aus oder können durch die zu geringe Stromleistung nicht die korrekte Position, in der sie sich befinden, erfassen. Dies führt zu ruckelnden und ständig bewegenden Servomotoren. Um die Stromleistung zur Verfügung zu stellen, muss dadurch eine externe Stromquelle angeschlossen werden. In diesem Aufbau wird ein fünf Volt starker mobiler Akku benutzt, der mithilfe eines aufgeschnittenen USB-Kabels die Servomotoren mit Strom versorgt.

## 6.2 Schaltplanung

Um die Potentiometer lesen zu können, werden sie zunächst über den 3,3 Volt und Erdungsanschluss mit dem ESP32 verbunden. Die Ausgänge werden mit analogen Pins verbunden. Da es sich um 15 Potentiometer handelt (drei für Zeige-, Mittel-, Ring-, kleinen Finger und Daumen), werden 15 Pins, die analoge Daten lesen können, benötigt. Da ein ESP32 Dev Module allerdings nicht genug Pins besitzt, die das Auslesen von Potentiometern ermöglicht, müssen die Daten anderweitig abgelesen werden. Hierfür sind sogenannte Multiplexer geeignet, mit denen man mithilfe von digitalen Schaltungen gezielt Pins abfragen kann. Die Verkabelung der Stromzufuhr ist in Abb. 48 abgebildet. Die obere Reihe der Stromzufuhr für die Bourns-Sensoren in Abb. 48 wird über die Kupferkabel am Shield V3 übernommen, welches in Abb. 47 dargestellt wird.

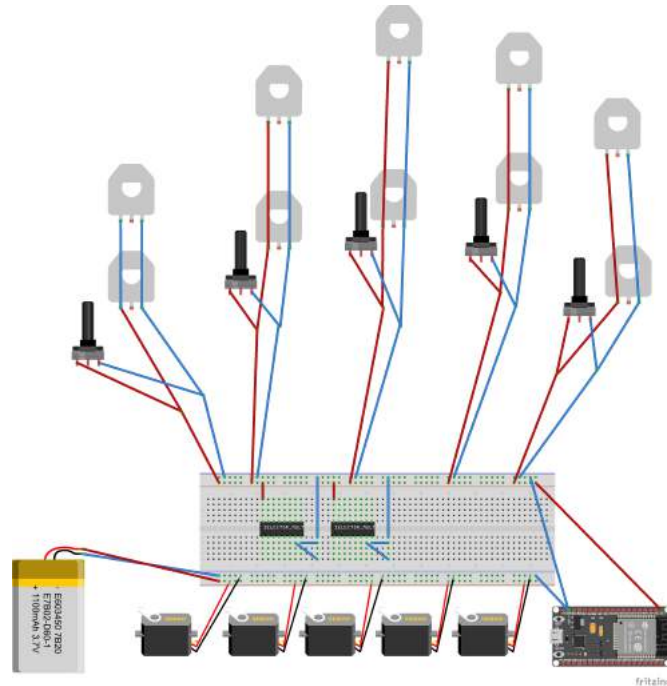


Abbildung 48: Verkabelung der Stromzufuhr. Die äußeren Bahnen, in denen die Motoren und Sensoren angeschlossen sind, sind über die ganze Länge verbunden. Die inneren sind nur über die Breite verbunden.

Mithilfe des *Multiplexer* von *Texas Instruments* ist es möglich, acht Kanäle, die an den Multiplexer angeschlossen sind, mit nur fünf Pins am ESP32, abzufragen. Hierfür werden drei der vom Multiplexer benötigten Pins für Switches genutzt. Diese Switches werden im Falle des Multiplexers mit den Buchstaben A, B und C benannt. Durch die gezielte (De-)Aktivierung der drei Switches können wie in Tabelle 2 ersichtlich die acht Kanäle ausgelesen werden. Null steht für deaktivieren, eins für das Aktivieren.

A	B	C	Kanal
0	0	0	0
1	0	0	1
0	1	0	2
1	1	0	3
0	0	1	4
1	0	1	5
0	1	1	6
1	0	1	7

Tabelle 2: Gezielte (De-)Aktivierung von Switches für Kanäle

$V_{DD}$  dient dem Anschluss des positiven Pols,  $V_{EE}$  und  $V_{SS}$  hingegen für den Anschluss des negativen Pols. Der Pin INH am Multiplexer dient der Löschung der bisherigen Messwerte und kann mit einem Reset verglichen werden. Nach der Auswahl des zu lesenden Kanals, werden die Daten über den Com Out/In Pin des Multiplexers an den ESP32 gesendet. Anschließend kann sofort der Kanal gewechselt und ausgelesen

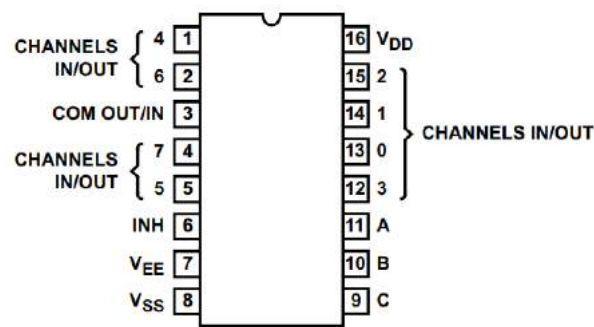


Abbildung 49: Aufbau des Multiplexers (von oben). [138] (S. 3)

werden. Um die benötigten 15 Potentiometer nun alle auslesen zu können, werden zwei Multiplexer verwendet. Um Pins am ESP32 zu sparen, können Pins des Multiplexers, die für die Auswahl sowie das Zurücksetzen der Kanäle benutzt werden, für beide Multiplexer gleichzeitig über eine Parallelschaltung genutzt werden. Um somit zwei Multiplexer mit 16 Kanälen nutzen und auslesen zu können, werden nur 6 Pins am ESP32 benötigt. Die Verkabelung der Datenkabel ist in Abb. 50 abgebildet. Die Pins und deren Zweck sind in Tabelle 3 gelistet.

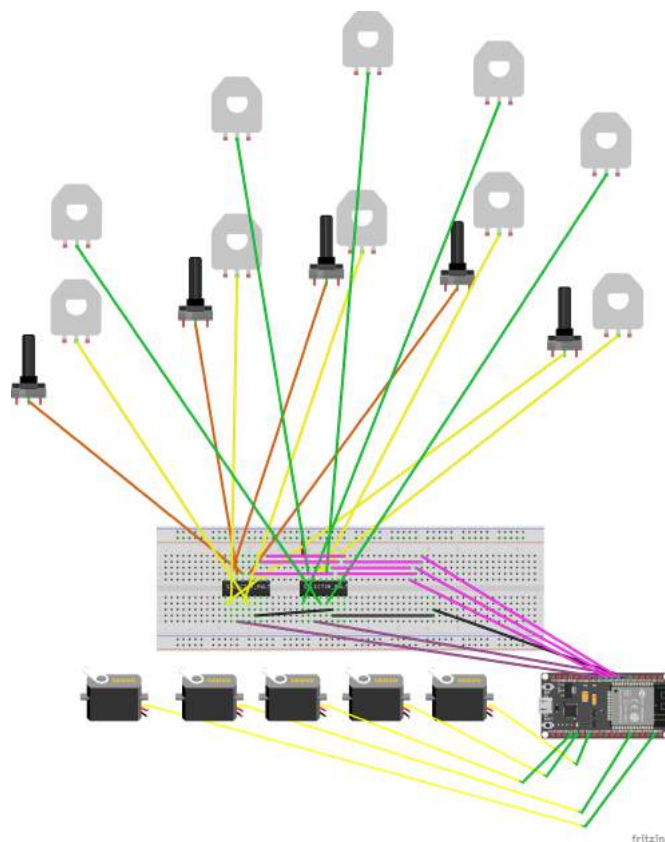


Abbildung 50: Verteilung der Datenkabel.

Modul	ESP32 Pin
Multiplexer Switch A	26
Multiplexer Switch B	25
Multiplexer Switch C	33
Multiplexer INH (Reset)	32
Multiplexer 1 Read	35
Multiplexer 2 Read	27
ServoMotor Daumen	21
ServoMotor Zeigefinger	23
ServoMotor Mittelfinger	4
ServoMotor Ringfinger	2
ServoMotor Kleiner Finger	15
Button für Kalibrierung	12

Tabelle 3: Pinout Tabelle.

### 6.3 Software

Im folgenden Kapitel wird über die Software genauer eingegangen. Neben gewählten Sprachen werden ebenso Aufbau und Abläufe genauer erklärt.

#### 6.3.1 Programmiersprachen

Die erste, grundlegende und sehr wichtige Entscheidung ist die Wahl der Programmiersprachen für dieses Projekt. Um einen sogenannten *Sketch*, eine Scriptdatei eines Arduinos und somit auch für die ESP32, zu programmieren, gibt es potentiall mehrere Möglichkeiten. Für das Programmieren von Arduinos wird in der Arduino Software standardmäßig mit einer eigenen Arduino Programmiersprache geschrieben, die C++ sehr nahesteht. Dies wird ersichtlich, da beim Kompilieren der Arduino-Code in C++ Code vorverarbeitet wird, welcher sich nur geringfügig vom ursprünglichen Quellcode unterscheidet. Zudem werden benötigte Bibliotheken als binäre Objekt-Dateien (Dateiendung .o) gesammelt und in eine .o Datei zusammengefügt. Der vorbereitete Sketch wird dann mit der .o Datei aller Bibliotheken verlinkt. Daraufhin wird der für den Menschen noch lesbare Code in eine Maschinensprache durch einen avr-gcc und avr-g++ Kompiler kompiliert. Während dem Kompilieren werden zudem nur Teile der großen zusammengefügten Bibliothek beigefügt, die im Sketch genutzt werden. Die damit entstandene Intel Hex Datei beinhaltet somit nur den für die Ausführung des Sketches benötigten Code. Dadurch wird Speicherplatz gespart. Die Hex-Datei wird dann über eine Serial-Verbindung auf den Speicher des Microcontrollers übertragen.[66]

Alternativen wären Programmiersprachen, die ebenso mit einem avr Kompiler in einen für Microcontroller lesbaren Code kompiliert werden können, wie Pascal [55] oder FORTH [72][81][25]. Mit Javascript Node.js Systemen können mithilfe des Esprui-

no Interpreters ebenso Sketches für ESP32 geschrieben werden.[22] Auch die Programmierung mit einem Assembler ist möglich, da bereits die Arduino-IDE auf dem GNU-Assembler aufbaut.[132] (S. 1)

Aufgrund der in der *Arduino-IDE* standardisierten Arduino-Programmiersprache und dessen Nähe zu C/C++, wurde sich für das Erstellen der Sketche für die *Arduino-IDE* entschieden. Zumal bereits Vorkenntnisse in beiden Sprachen vorhanden waren. Insbesondere wegen der vielen Ähnlichkeiten zu C/C++ und somit teils auch C#, kann die Arduino-Sprache auch von einer Vielzahl von Programmierern gelesen und verstanden werden. Dies liegt daran, das anhand der Suchanfragen für Tutorials im November 2021 C/C++ auf Platz 5 und C# an Platz 4 der weltweit meist genutzten Programmiersprachen zu finden ist.[122] In Deutschland ist C/C++ auf Platz 7 und C# auf Platz 3.[123]

Worldwide, Nov 2021 compared to a year ago:

Rank	Change	Language	Share	Trend
1		Python	29.47 %	-1.5 %
2		Java	17.27 %	+0.8 %
3		JavaScript	8.97 %	+0.5 %
4		C#	7.17 %	+0.8 %
5	↑	C/C++	6.6 %	+0.6 %
6	↓	PHP	5.52 %	-0.5 %
7		R	3.88 %	-0.0 %
8		Objective-C	2.21 %	-1.4 %
9		Swift	2.13 %	-0.2 %
10		TypeScript	2.09 %	+0.2 %

Abbildung 51: Top 10 Programmiersprachen weltweit.

Unreal Engine 4 hingegen bietet mit *Unreal C++* eine Version von C++, die beim Programmieren für UE als Unterstützung dienen soll. Das klassische C++ wird ebenso unterstützt.[37] Mithilfe von Erweiterungen werden auch andere Programmiersprachen in UE nutzbar: Mithilfe der Erweiterung *Unreal.js* kann JavaScript für die Programmierung innerhalb Unreal Engine 4 genutzt werden.[76] Auch die skriptbasierte Sprache *Python* ist mit der UE Version 4.26 als ein experimentelles Feature hinzugefügt worden.[75] Das *UnrealEnginePython* Plugin verspricht sogar mehr Features zu besitzen als die noch experimentelle offizielle Python-Anbindung.[62] Die Scriptsprache *SkookumScript* ermöglicht Features von C++ mit den Vorzügen von Scriptsprachen zu bewältigen. Epic Games erwarb sogar 2019 *SkookumScript* und das



Entwicklerstudio dahinter.[67, 108] In Zukunft plant Epic Games eine noch offiziell unbenannte eigene Scriptsprache für UE zu entwickeln, deren Namen der Dateien mit `.verse` enden.[170] UE bietet auch eine Methode, um visuell das Programmieren zu ermöglichen. Ähnlich wie visuelle Hilfsmittel für das Erstellen von Shadern in Programmen wie Blender, Unreal Engine und Unity, werden Blöcke zusammen verbunden, um so Funktionen, Variablen und Abläufe zu gestalten und zu verändern. Diese Blöcke, im weiteren Verlauf *Nodes* genannt, können damit das Programmieren mit Text ersetzen.

### 6.3.2 Blueprints

*„I think getting started with Blueprint is easy. [...] It's kind of like a flowchart and it kind of just works in a way“ [141] (3:40 - 3:50)*

Die Arbeit mit Blueprints in UE hat seine Vor- und Nachteile. Die Vorteile sind zu einen, dass ohne tieferes Wissen in einer Programmiersprache Blueprints erlernt und genutzt werden können. Die visuelle Darstellung der Nodes und deren Verbindung ermöglicht eine einfache Darstellung von Reihenfolgen. Eine Aufgabe oder ein Problem kann oftmals in einzelne Bedingungen oder Punkte aufgeteilt werden, die nacheinander ausgeführt oder abgefragt werden müssen.[13] Ein Flowchart kann fast identisch als Nodes in Blueprints angelegt werden. In Abb. 52 wird ein Flowchart für eine einfache Aufgabe in Blueprints dargestellt.

Eine Zone soll, beim Eintritt eines Objekts überprüfen, ob es der Spieler ist oder ein anderes Objekt. Für den Fall, dass es ein Spieler ist, soll etwas ausgeführt werden. Wenn es sich nicht um einen Spieler handelt, soll nichts ausgeführt werden. Im unteren Teil sind die verbundenen Nodes abgebildet, um diese Aufgabe zu lösen. Die rote Node ist ein Event, das ausgeführt wird, wenn ein Objekt mit diesem überlappt. Die weiße Verbindung zu einer Branch-Node, die überprüft, ob etwas stimmt oder falsch ist, zeigt die Ausführreihenfolge an. Entsprechend wird der Branch ausgeführt, nachdem eine Überlappung passiert. Der blaue Punkt des Event-Node beinhaltet das Objekt, welches die Überlappung verursacht. Die Verbindung zu einem `==` Node vergleicht, ob dieses Objekt identisch zum Player-Objekt ist. Das Ergebnis führt zur Branch-Node, welches diese Information als Bedingung nutzt. Die weiße Linie aus True führt nun etwas aus, wenn die Bedingung wahr ist, in diesem Fall also, wenn das Eintrittsobjekt der Player ist. Da, wenn es sich nicht um den Spieler handelt, nichts ausgeführt werden soll, fährt auch keine Linie aus dem False-Ausgang aus.

Das System der Blueprints ist damit sehr gut für das schnelle Konzipieren und Überprüfen von Aufgaben geeignet. Das Ändern von Abläufen kann so durch einfaches

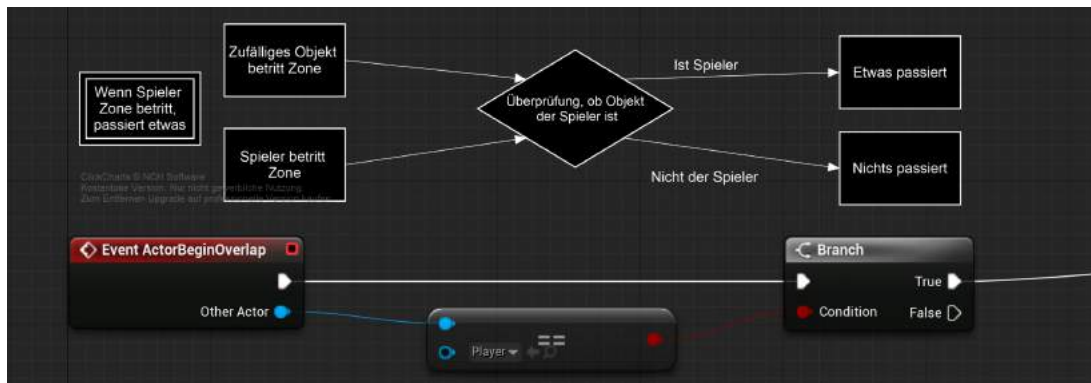


Abbildung 52: Aufbau eines Blueprint.

Ziehen und neu verknüpfen von Nodes umgesetzt und erweitert werden. Der Nachteil ist allerdings, dass Blueprints bei identischen Aufgaben nicht identisch performant sind wie C++ Code. Dies liegt daran, dass Blueprints auf virtuellen Maschinen (VM) laufen, die dann native C++ Funktionen aufrufen. Blueprints sind damit, abhängig von der Komplexität der Aufgabe, ungefähr zehn Mal langsamer als nativer C++ Code. Ein weiterer Unterschied in der Performance von Blueprints ist der Status der Anwendung, in dem dieser Ausgeführt wird. So ist, wie in Abb. 53 zu erkennen ist, die Performance besonders schlecht, wenn das Projekt im Unreal Engine Editor läuft. Nach dem Bauen einer Anwendung ist im Falle einer Entwickler-Version die Performance deutlich besser. Bei einer normalen Spielversion ist die Performance noch einmal verbessert. Diese Ergebnisse entstanden in einem Testprojekt, das in einem Loop komplexe Zahlen berechnet und kann daher allerdings nicht generell angewendet werden.[141] (17:55 - 30:52)

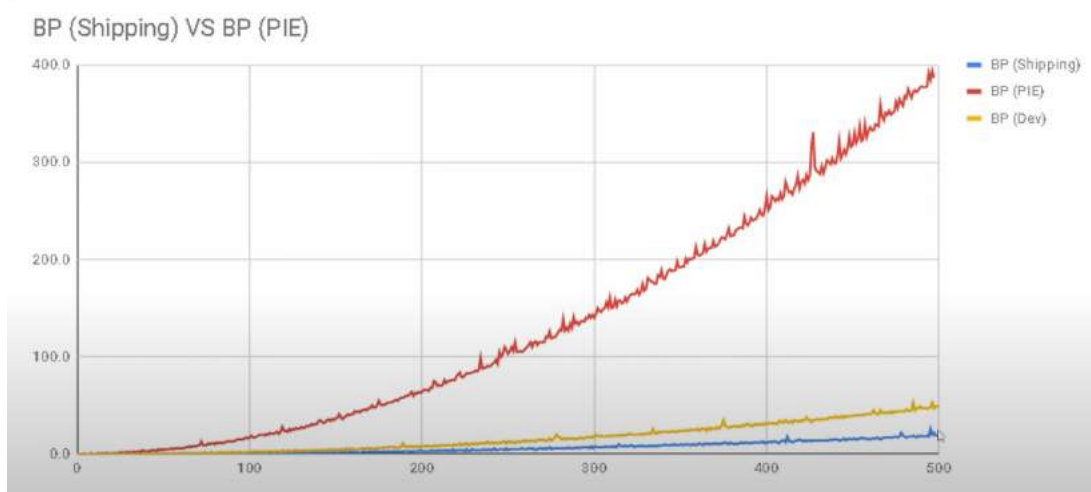


Abbildung 53: Performance von Blueprints im Editor (Rot), Developer Build (Gelb) und Shipping Version (Blau). [141] (22:58)

Allerdings kann die Ausführung von Blueprints in VMs umgangen werden, indem diese vorher in C++ Code umgewandelt werden. Dies nennt Epic Games *Nativizing* und ermöglicht eine performante Nutzung von Blueprints. Dies führt allerdings dazu, dass die Speichermenge für das Projekt erhöht wird. Der C++ Code, der durch Nativizing entsteht, ist Code, der nicht für das Lesen durch Menschen formatiert ist. Dennoch kommen auch so Nutzer ohne Programmiererfahrung auf performanten Code.[49]

Für das Projekt wurde, aufgrund des schnellen Prototypings, Blueprints genutzt. Da Blueprints nahezu keine Kompilierzeit benötigen, ist damit möglich, schnell Dinge zu ändern und das Programm neu auszuführen. Die Performance der Blueprints wird nach Vollendung geprüft und evaluiert.

#### 6.4 ESP32 Programmierung als Sender

Die Datenübertragung soll über Bluetooth direkt an den PC, sowie über Wifi mit einem ESP32-Empfänger als USB-Dongle, funktionieren. Daher wird zunächst BluetoothSerial für die Nutzung von Bluetooth, ESPnow für Wifi-Übertragungen zwischen ESPs, sowie ESP32Servo für die Ansteuerung von Servomotoren mit ESP32-Geräten, als Bibliothek eingebunden. Zudem werden Abkürzungen für die Pins, die am ESP32 genutzt werden, definiert. Die Pins können Tabelle 3 entnommen werden. Anschließend wird der Name des Geräts genannt. Dieser bildet sich aus einem Wort (Standard ist *Hand*) und einem Integer, der als String in den Namen eingefügt wird. Bei der Nutzung von Bluetooth, wird der Name für die Suche nach Bluetooth-Geräten genutzt (wie z. B. *Hand1*). Der Integer wird für die Wifi-Übertragung genutzt, um die Herkunft der Daten zu speichern. Die Wahl, welche Übertragungstechnologie genutzt wird, wird über einen Bool im Sketch entschieden.

Um die Daten der Sensoren speichern zu können, müssen in unserem Sketch Variablen angelegt werden. Da es sich um fünf Finger pro Hand mit jeweils drei Sensoren handelt, können Structs verwendet werden, um Werte logisch zu speichern. Nun wird definiert, welche Daten pro Sensor gespeichert werden sollen. Um eine Kalibrierung zu ermöglichen, wird ein Min- und einen Max-Wert, den wir speichern, benötigt. Zusätzlich wird der aktuelle Wert gespeichert, der vom Sensor entnommen wurde. Um eine Filterung zu ermöglichen, wird ein Array erstellt, der die letzten gemessenen Werte beinhaltet. Diese Variablen werden demnach in einem Struct mit dem Namen *FingerJoint* zusammengefasst.

Nun können die Finger als Struct deklariert werden. Dies beinhaltet ein Label als String, welches den Namen des Fingers für das Debugging in der Konsole speichert. Ebenso wird das eben beschriebene Struct eingefügt. Da pro Finger drei Sensoren genutzt werden, wird ein Joint-Array aus den drei *FingerJoint* erstellt.



Abbildung 54: Aufbau der Struct für Sensoren.

Das Struct *Finger* wird anschließend als Array mit dem Namen der Finger gestartet. Dieser Aufbau ermöglicht einen einfachen Durchlauf der Daten mithilfe von zwei Loops für die Verarbeitung und Ausgabe der Daten.

```

1 struct Finger Fingers[5] = { { "Pinky" }, { "Ring" }, { "Middle" }, { "Index" }, { "Thumb" } };

```

Codeauschnitt 6.1: Finger definieren und Objekte erstellen

Beim Starten des Sketches wird entweder Bluetooth oder Wifi gestartet. Anschließend werden alle Pins als Input oder Output deklariert. Nun werden die Aufgaben auf beide verfügbaren Kerne des ESP32 aufgeteilt:

## 6.5 Daten auslesen

Das ESP32-Framework ermöglicht das Starten von Tasks, die auf eigenen Kernen laufen und dabei, von den anderen Kernen unabhängig, bearbeitet werden können. Damit kann die Datenabfrage auf einem Kern, unabhängig von der anderen Kalkulationen, dauerhaft ausgeführt werden. Hierfür wird auf dem ersten Kern eine Schleife der Funktion *readMultiplexer()* erstellt, der die Daten aus dem Multiplexer löscht. Anschließend werden Daten von dem Multiplexer angefragt, nachdem die zugeteilten Kanäle und Multiplexer auf die Sensoren je Finger kategorisiert wurden. In folgendem Beispiel wird die Abfrage anhand des ersten Sensors des kleinen Fingers dargestellt:

```

1 Fingers[0].joints[0].Data[FilterNum] = readMultiplexer(5, 1);

```

Codeauschnitt 6.2: Erster Sensor des kleinen Fingers wird aus Kanal 5 des ersten Multiplexers ausgelesen

Hierfür werden, wie in Tabelle 2 zu erkennen ist, die Channels mit den A, B und C Channels in der Funktion *readMultiplexer* angesteuert und ausgelesen. Die Parameter, die übergeben werden, sind der Kanal (im Beispiel Kanal 5) und die Nummer des

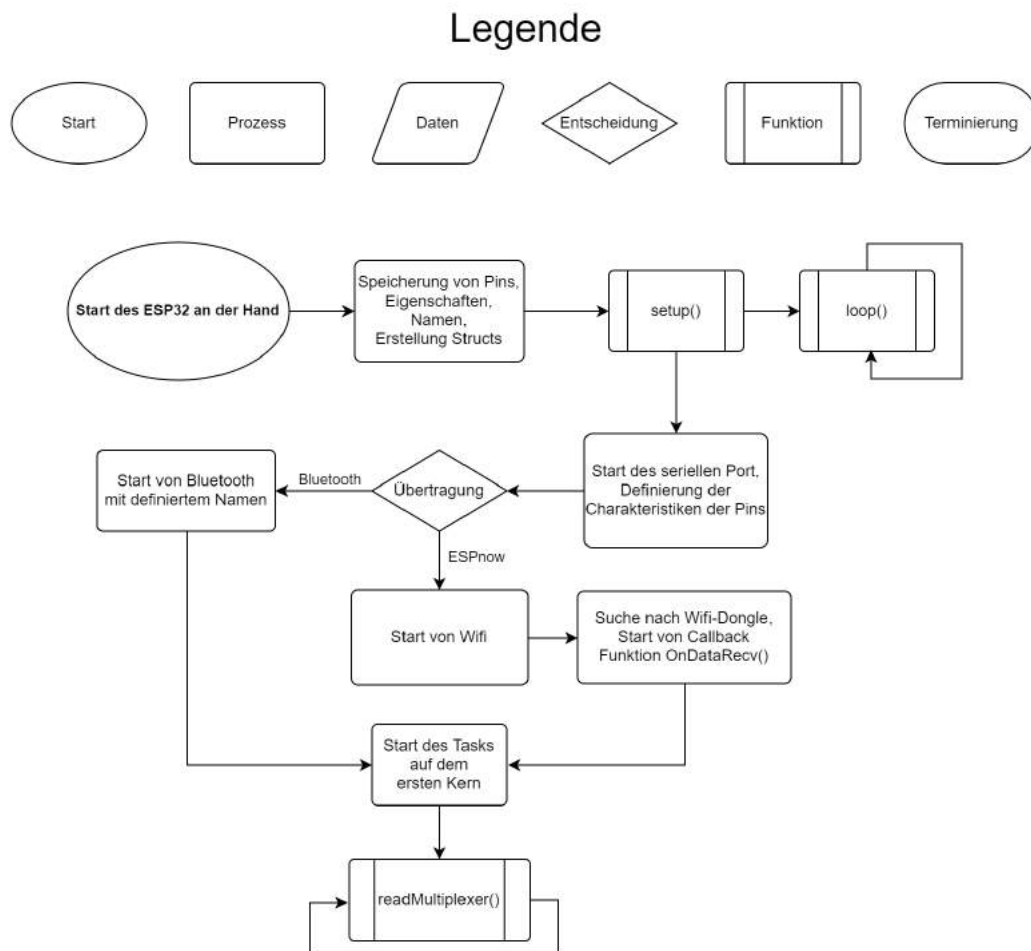


Abbildung 55: Ablauf des Starts des Programms inklusive Auslesen der Sensordaten in `readMultiplexer()`. Oben ist die Legende für dieses und kommende Ablaufdiagramme dargestellt.

Multiplexers. Die ausgelesenen Daten werden in das Data-Array an eine nummerierte Stelle, oben *FilterNum* genannt, eingesetzt. Die nummerierte Stelle ist abhängig vom Durchlauf des Loops. Zu Anfang ist die erstellte Platzierungsnummer Null und wird nach jedem Durchlauf um eins erhöht. Wenn nach einem Durchlauf die Platzierungsnummer die Größe des Data-Arrays erreicht hat, wird die Platzierungsnummer wieder auf null gesetzt. Damit wird das Data-Array zunächst gefüllt und dann immer wieder mit neuen Daten überschrieben. Diese Daten werden später beim Filtern verwendet.

## 6.6 Daten verarbeiten und senden

Auf dem zweiten Kern werden die gemessenen Daten als Schleife im `Loop()` verarbeitet. Die Daten im Data-Array werden mit einem Hoch- und Tiefpassfilter gefiltert. Der Durchschnittswert wird anschließend als *Val* in jedem *FingerJoint* gespeichert.

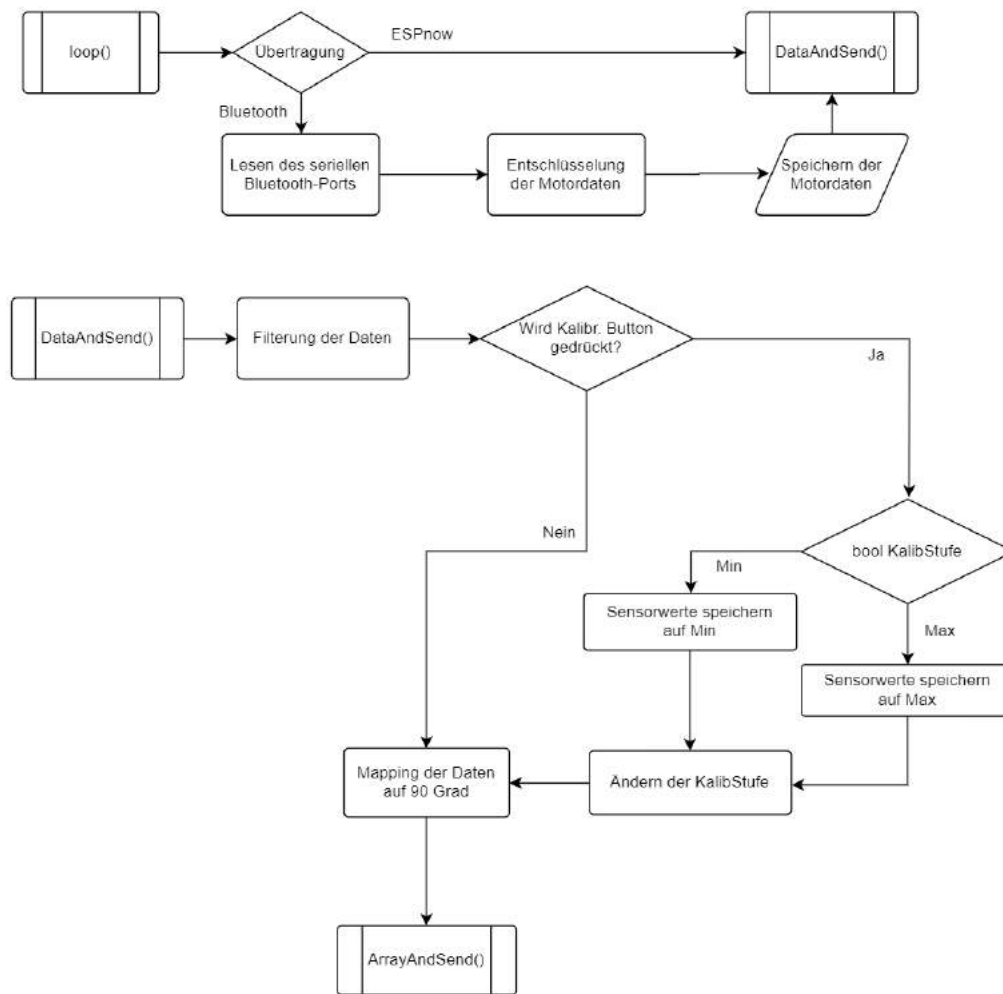


Abbildung 56: Vor dem versenden der Daten werden diese zunächst gefiltert und anhand der Kalibrierung in ein Rotationsfenster verrechnet.

Für das Kalibrieren der Werte für die Daten des Fingerspreizens, wird dauerhaft der Durchschnitt mit dem Min- und Max-Wert verglichen, ob diese größer oder niedriger ist. Für das Kalibrieren der anderen Sensoren kann entweder über einen Knopfdruck der Min und Max Wert eingestellt, oder wie beim Spreizen dauerhaft die höchsten und niedrigsten Werte in Max und Min gespeichert werden. In Abb. 56 wird das Kalibrieren per Knopf dargestellt. Um die Werte einfacher als Drehung interpretieren zu können, wird der Zahlenbereich der Messdaten umgerechnet. So erhalten die neuen Zahlen einen Wert zwischen Null bis 90, abhängig der kalibrierten Werte.

Um die Daten über Bluetooth an UE versenden zu können, müssen die Messdaten in Bytes konvertiert werden. Diese Bytes werden anschließend in ein Byte-Array zusammengefügt. An den Anfang des Byte-Arrays wird ein Byte hinzugefügt, welcher nicht als Messung in dem Byte-Array sonst enthalten sein darf. Dieser Byte dient zur Identifizierung des Anfangs. Die Daten werden anschließend an den über Bluetooth

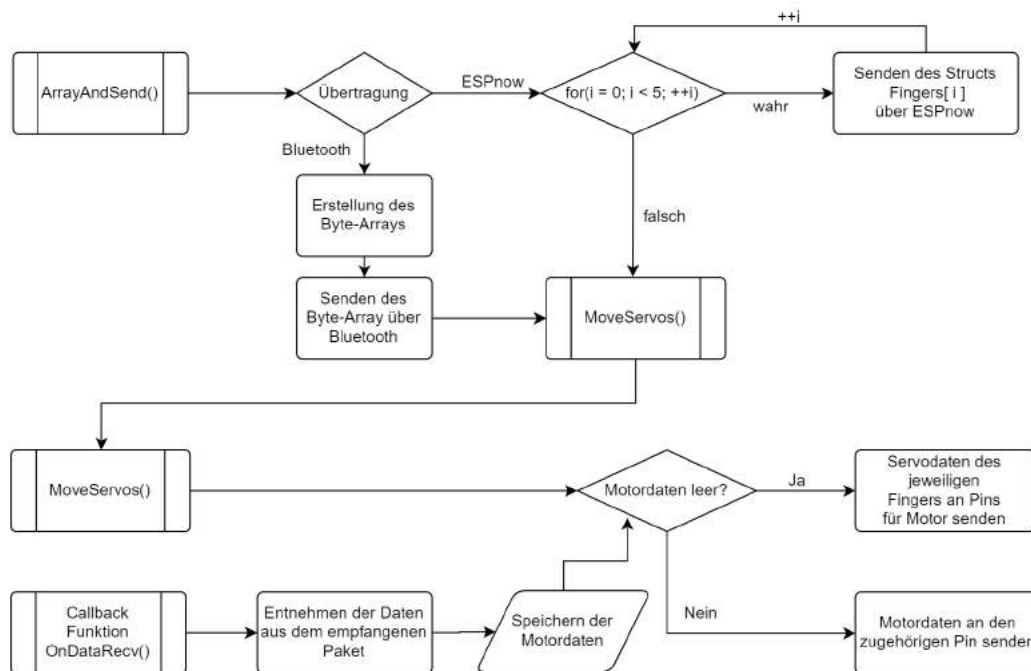


Abbildung 57: Ablauf des Sendens der Daten über Bluetooth an den PC oder über Wifi an den USB-Dongle. Anschließend werden die Motoren bewegt.

verbundenen Computer versendet. Im Falle von Bluetooth wird nach dem Senden des Arrays für eine kurze Zeit das Bluetooth-Signal pausiert, um einen Overflow zu unterbinden und die Möglichkeit für das Empfangen von Daten zu ermöglichen.

Wenn Wifi genutzt wird, werden, wie in Abb. 57 dargestellt, die Structs *Finger* vom ESP32 des Handschuhs an den ESP32 Dongle mit Wifi über *ESPnow* versendet. Der Dongle liest, wie in Abb. 59 in der Funktion *OnDataRecv()* zu sehen ist, zuerst die Handschuhnummer aus einem *Finger* aus, um daraufhin die empfangenen Daten in einem Struct-Array mit dem Namen *Hand* zu speichern. Die Handschuhnummer, die entnommen wurde, entspricht dem Platz, der im *Hand*-Array gespeichert wird. Der USB-Dongle verpackt, ähnlich wie beim Übertragen mit Bluetooth, die Daten in ein Byte-Array und versendet die Daten über den Serial Port. Dies wird in Abb. 59 dargestellt. Allerdings wird an zweiter Stelle des Byte-Arrays die Handschuhnummer hinzugefügt.

Heraus kommt ein Byte-Array mit 16 Bytes, wenn Bluetooth genutzt wird. Die Größe des Arrays, wenn Wifi mit einem Dongle verwendet wird, beträgt 17 Bytes. Ein Byte-Array, der auf dem USB-Dongle erstellt wird, wird in Abb. 58 dargestellt

Start    Pinky1    Pinky3    Ring2    Middle1    Middle3    Index2    Thumb1    Thumb3  
 99|1|19|17|09|46|47|31|58|31|84|61|76|21|76|54|31  
 HandNum    Pinky2    Ring1    Ring3    Middle2    Index1    Index3    Thumb2

Abbildung 58: Ein Byte-Array, der bei der Verwendung von Wifi auf dem USB-Dongle erstellt wird.

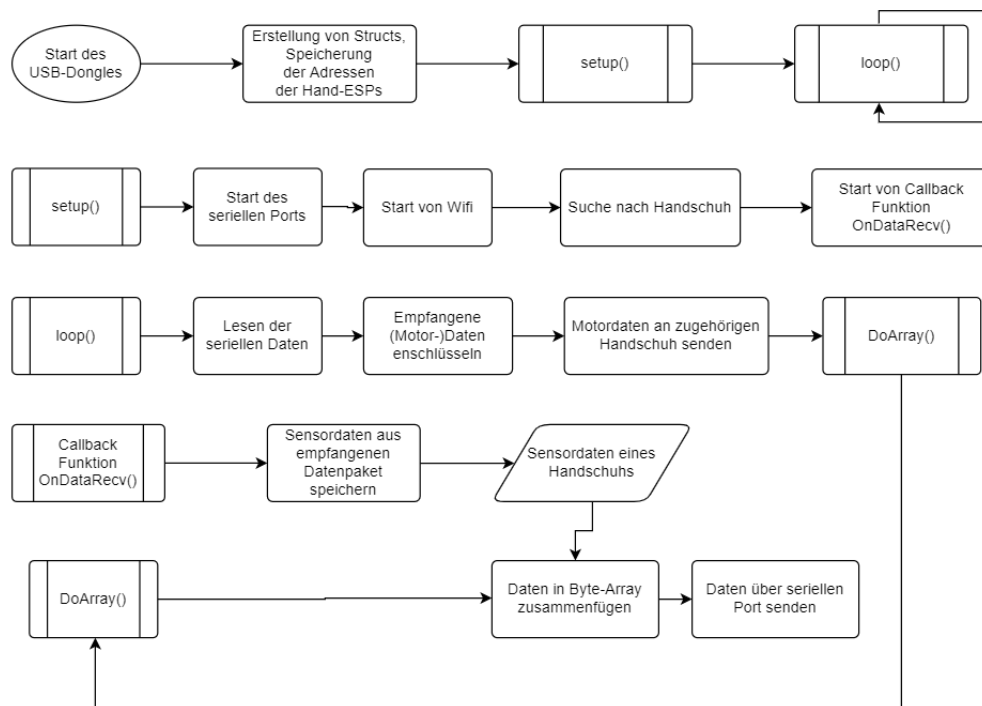


Abbildung 59: Ablauf des Programms für den USB-Dongle. Er liest den seriellen Port, sendet die Daten an die Hand, empfängt die Daten der Hand und sendet diese über den seriellen Port weiter.

## 6.7 Implementierung in Unreal Engine

Zunächst wird überprüft, ob der eingegebene Port ein ESP32 mit der gewünschten Software ist und ob der Port frei zur Verfügung steht. Wenn Bluetooth für die Übertragung genutzt wird, wird für jede Hand ein eigener Port benötigt. Es muss zudem in UE definiert werden, ob Bluetooth oder Wifi genutzt wird. Nun wird die Aktualisierungsrate bestimmt, die je nach USB-Verbindung, Bluetooth-Version und Bluetooth-Dongle eingestellt werden muss. Die Aktualisierungsrate bestimmt die zeitliche Verzögerung für das Event, welches die Daten des ESP32 empfängt und entschlüsselt. Das empfangene Byte-Array wird zunächst gespeichert und anschließend in einer Schleife Byte für Byte verarbeitet.



Die Bytes werden wegen UE-Eigenheiten zunächst in einen String und anschließend direkt in einen Integer umgewandelt. Hierdurch werden die Bytes als Hexadezimale interpretiert, wodurch mit einem Byte auch Zahlen von 0-255 genutzt werden können. Alternativ können Bytes auch ohne Umwandlungen interpretiert werden, brauchen hierfür allerdings Bytes, die als Trennungen der Werte dienen. Da Werte von Null bis 90 genutzt werden, müssen mindestens zwei Bytes hintereinander als Trennung von Werten genutzt werden, wie in Abb. 60 dargestellt ist. Dies erhöht die Menge an Bytes, die vom ESP32 gesendet werden müssen. Die mehrfache Umwandlung der Bytes ist zwar nicht performant, spart allerdings an Datenrate am Serial Port sowie am ESP32. Aus diesem Grund wird die mehrfache Umwandlung genutzt.

9900	100	1900	1700	0900	4600	4700	3100	5800	3100	8400	6100	7600	2100	7600	5400	3100
Start	HandNum	Pinky1	Pinky2	Pinky3	Ring1	Ring2	Ring3	Middle1	Middle2	Middle3	Index1	Index2	Index3	Thumb1	Thumb2	Thumb3

Abbildung 60: Längeres Byte-Array mit Nullen als Trennung.

Da eine serielle Kommunikation nicht immer synchron abläuft, ist der erste Byte, der von einem ESP32 gesendet wird, nicht immer der erste Byte, der in Unreal Engine ankommt. Daher wurde ein Start definiert. In diesem Fall ist der Wert '99' als Beginn der Übertragung markiert, wodurch zunächst alle Werte durchlaufen werden, bis dieser Wert gefunden wird. Für die Speicherung der Daten werden die Structs aus Abb. 54 sehr vereinfacht genutzt. Als Struct-Array gibt es die *Hand*, welche fünf *Finger* beinhaltet, die jeweils drei *Joints* und für Debugging ein Label besitzt.

Der Wert '99' definiert den Start des Datenflusses mit einem Bool. Wenn Wifi genutzt wird, ist der nachfolgende Wert die Nummer der Hand, in der die nachfolgenden Werte gespeichert werden sollen. Nachdem die zu speichernde Hand definiert ist, wird dies auch mit einem Bool vermerkt. Der nächste Wert des Sensors wird nun in eine Verstrickung der Struct-Arrays gespeichert:

Der Byte, der nun einen Sensorwert beinhaltet, wird in die nullte Position der *FingerJoints* des nullten *Fingers* der Position des *Hand* Arrays gespeichert. Nach der Speicherung wird der *JointNr* um eins erhöht. Dadurch ist er nun auf der eins. Nun wird der nächste Byte gelesen und interpretiert. Dieser wird schließlich wieder in die gespeicherte Hand, dessen Finger und nun erster Indexplatz des *FingerJoint*-Arrays, gespeichert. Wenn die Menge an *Joints*, die pro Finger an den Handschuh gesetzt wurden, durchgelaufen sind, wird schließlich das Int *FingerNr* um eins erhöht und *JointNr* wieder auf Null gestellt. Dann wird der nächste Wert gelesen. Dieser Wert wird in den ersten Platz der *Finger* im nullten *FingerJoint* gespeichert werden. Dies geschieht, bis alle *Finger* und deren *FingerJoints* durchlaufen sind oder der Wert '99' als Beginn des Byte-Arrays gelesen wird. Anschließend geht die Speicherung von neu-

em los mit einer neuen Hand. Die Speicherung folgt entsprechend der Reihenfolge, dass zuerst alle FingerJoints eines Fingers gefüllt werden, worraufhin der nächste Finger gefüllt wird, bis alle Finger einer Hand gespeichert sind. Der Durchlauf der Daten wird in Abb. 62 dargestellt.

Die Daten können nun anhand von Händen im *Hand Struct Array* zugeordnet werden. Die benötigten Daten werden dann in einem Animation Blueprint in Unreal Engine übertragen. Diese Animation Node ist für jeden 3D-Charakter individuell. Hier können die Daten für zwei Hände nun aus dem ersten Blueprint ausgelesen und noch einmal gespeichert werden. Dies ist nötig, da in Animation Blueprints keine seriellen Objekte erstellt werden können. Die Daten werden dann an die Rotationsachsen der jeweiligen *Bones* des Charakters übertragen. Um die Finger nicht anhand der immer neuen Daten herumspringen zu lassen, werden die Rotationsachsen zwischen den letzten und den aktuellen Daten linear interpoliert. Dies kann über eine Easing Node umgesetzt werden.

## 6.8 Force Feedback und Hand Posing

Die Finger des 3D-Charakters in UE strahlen einen *Ray* aus den Fingerkuppen aus, um die Distanz zu Objekten messen. Ein *Ray* ist ein Pfad oder Lichtstrahl, der bei einem Kontakt Informationen über das Objekt oder den Kontaktpunkt ausgeben kann. Falls Objekte in eine nahe Distanz kommen, wird die Entfernung in eine Zahlenskala konvertiert und in einen String verbunden. Diese Werte werden, wie beim Sammeln der Sensordaten, in einem Array gesammelt. Der Anfang des Arrays besitzt auch hier wieder eine Information, die den Start des Arrays definiert.

Auch hier muss wieder unterschieden werden, ob die Übertragung über Bluetooth oder per USB geschieht. Falls Bluetooth genutzt wird, muss die Aktualisierungsrate in Unreal Engine reduziert werden, da der Datenfluss sonst gestört werden kann. Wohingegen mit Bluetooth die Motordaten direkt an den Port des ESPs für den Handschuh gesendet werden, muss für die Wifi-Variante, ähnlich wie beim Senden zu UE, eine Kennziffer mitgesendet werden. Diese Kennziffer dient zur Zuordnung des Handschuhs, an den die Daten gesendet werden sollen. Anschließend läuft auf dem ESP, ähnlich wie beim Sortieren der Daten in Unreal Engine, ein Algorithmus, der die empfangenen Datenpakete wieder aufteilt. Wenn Wifi genutzt wird, wird die Kennziffer für die Auswahl des Handschuhs ausgelesen und die nachkommenden, verarbeiteten Motordaten an den dazugehörigen Handschuh versendet. Der empfangende Handschuh überträgt die Daten an die Pins für die Servomotoren. Wenn Bluetooth genutzt wird, werden die Daten direkt vom ESP am Handschuh entschlüsselt und auf die Pins für die Servomotoren geschrieben. Damit rotieren sich die Motorflügel in die

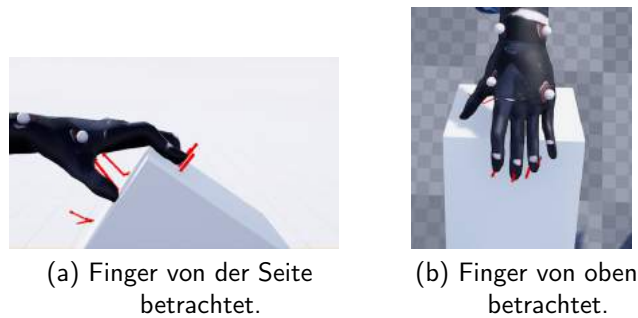


Abbildung 61: Automatisches Hand Posing dargestellt an einem Würfel.

Rotationsachse, die von Unreal Engine gesendet wurden.

Da die genutzten Servomotoren sich gegebenenfalls einen bestimmten Weg drehen müssen, bis die Finger an der gewünschten Stelle gestoppt werden können, muss diese dadurch entstehende Verzögerung weitestgehend minimiert werden. Die Servomotoren bewegen sich anhand der Sensordaten der jeweiligen Finger, jedoch mit einer Verzögerung. Diese Verzögerung ermöglicht eine Verkürzung der Dauer, die der Motor benötigt, um die Finger zu stoppen, da sich die Motoren nicht mehr bis zu  $180^\circ$  rotieren müssen. Die empfangenen Daten aus UE überschreiben allerdings jederzeit die automatischen Motorrotationen. Die Zahlenskala aus UE, wie nah ein Finger an einem Objekt ist, kann zudem auch genutzt werden, um Vibrationsmotoren, die an den Handschuh angeklebt werden können, zum Vibrieren zu bringen. Hierfür kann die Entfernung als Vibrationsstärke dienen. Ebenso kann dies auch binär genutzt werden, wodurch die Vibrationsmotoren nur zu vibrieren anfangen, wenn die Entfernung unter einem bestimmten Wert ist.

Beim Hand Posing werden die Finger des 3D-Charakters auf Oberflächen rotiert, so dass der natürliche Eindruck entsteht, dass man das Objekt greift. Umgesetzt wird das, indem jedes Gelenk einen Ray ausstrahlt, der die Rotation des Gelenks beim Greifen darstellt. Wenn diese Rays ein Objekt treffen, wird die Position des Fingerknochens mit der Position des Treffers verglichen. Wenn die Positionen nah genug beieinander sind, bewegt sich der Finger an die Position des Treffers. Dies wird am Beispiel eines Würfels in Abb. 61 dargestellt. Das Bewegen des Fingerskeletts wird durch eine inverse Kinematik (IK) umgesetzt. Eine inverse Kinematik ermöglicht, dass sich ein Knochen an einen Punkt bewegt, sodass die Knochen, die in der Hierarchie darüber liegen, diesem folgen. Das Annähern der Finger zur Oberfläche kann durch den Alphawert in der *Two Bone IK* Node animiert werden, indem bei einem Treffer der Alphawert langsam erhöht wird. Diese Umsetzung ist eine veränderte Umsetzung und einfachere Version des Systems, welche im Spiel *Lone Echo* genutzt wird.[125] (S. 17-27)

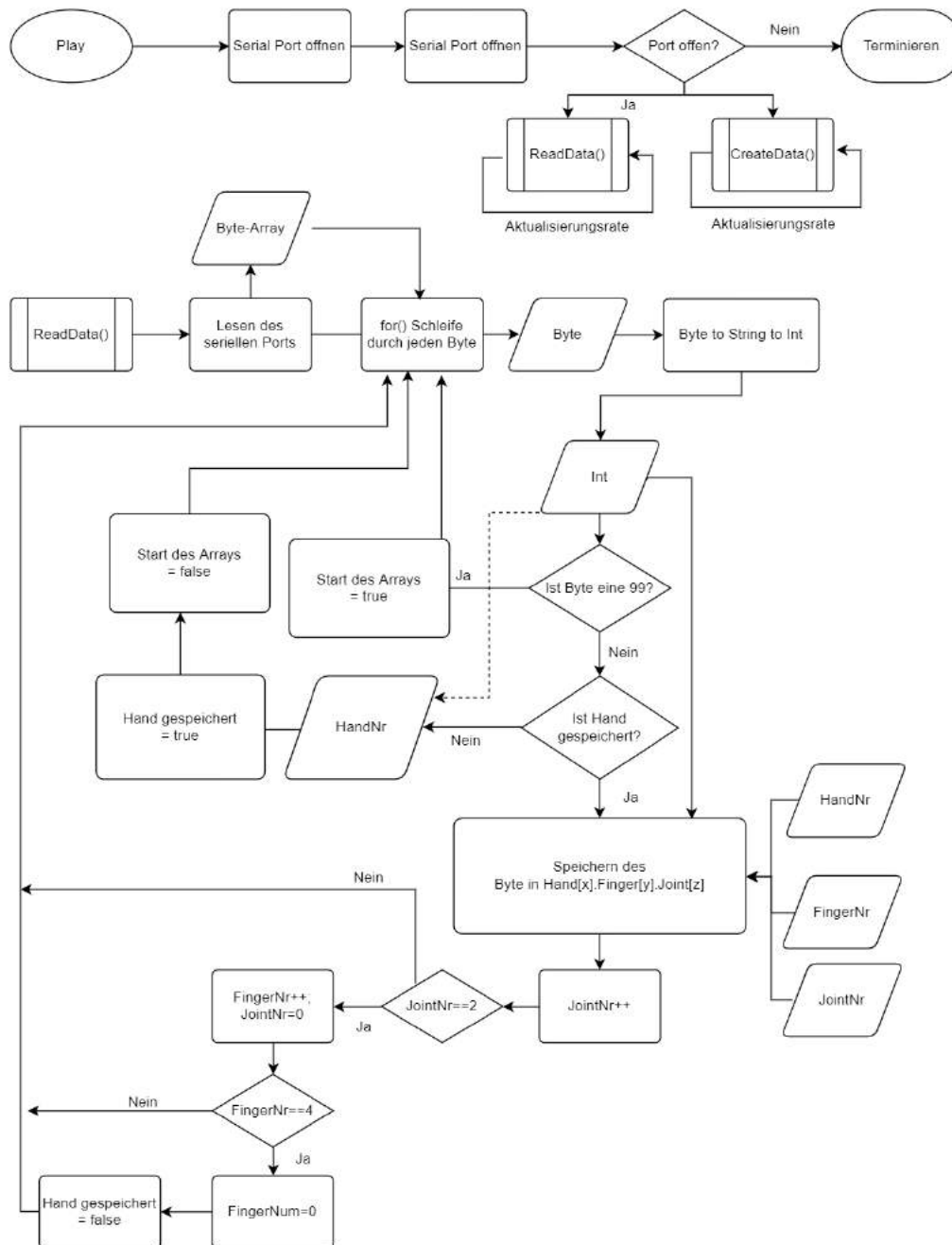


Abbildung 62: Ablauf für das Lesen und Speichern der Daten, vom USB-Dongle empfangen werden.

## 7 Evaluation

In diesem Kapitel wird die Arbeit in verschiedenen Kategorien evaluiert.

### 7.1 Einarbeitung

Die Evaluation der Einarbeitungszeit soll den Aufwand für die Arbeit darstellen und für eine Selbstreflektion dienen. Die Einarbeitung in Techniken des 3D-Drucks, das Entdecken der Limitationen und Möglichkeiten, war einer der Schwerpunkte dieser Arbeit. Es wurde viel Zeit aufgewendet, um stabile und haltbare Modelle zu entwerfen und diese auszudrucken. Ebenso wurde für die Konstruktion einer kabellosen Kommunikation Optimierungsmöglichkeiten studiert, um Probleme zu lösen und die Geschwindigkeit und Übertragungsrate durch Optimierungen zu steigern. Das Verständnis von Sensoren, deren Filterung, Nutzen und Vorteile wurden tiefgreifend verstanden und handwerkliche Fähigkeiten wurden durch die Umsetzung mehrerer Prototypen verstärkt. Die Erstellung und Integration eines Motion Capture Handschuhs mit haptischem Feedback konnte trotz Einarbeitungszeit in einer präsentierfähigen Prototypversion umgesetzt werden.

### 7.2 Innovation der Arbeit

Die Innovation dieser Arbeit liegt vor allem darin, dass ein möglichst kostengünstiger Motion Capture Handschuh mit haptischem Feedback und 15 *degrees of freedom* (DoF) für die Finger entstanden ist, der nach der Veröffentlichung der Arbeit im Internet frei zum Herunterladen verfügbar sein wird. Freie Handschuhe gibt es zwar bereits zum Herunterladen, allerdings besitzen diese über bisher über keine äquivalente Menge an Sensoren oder Eigenschaften.

### 7.3 Evaluierung des Handschuh Designs

Hier wird das Design des Handschuhs anhand der in Kapitel 5 beschriebenen Werte evaluiert.

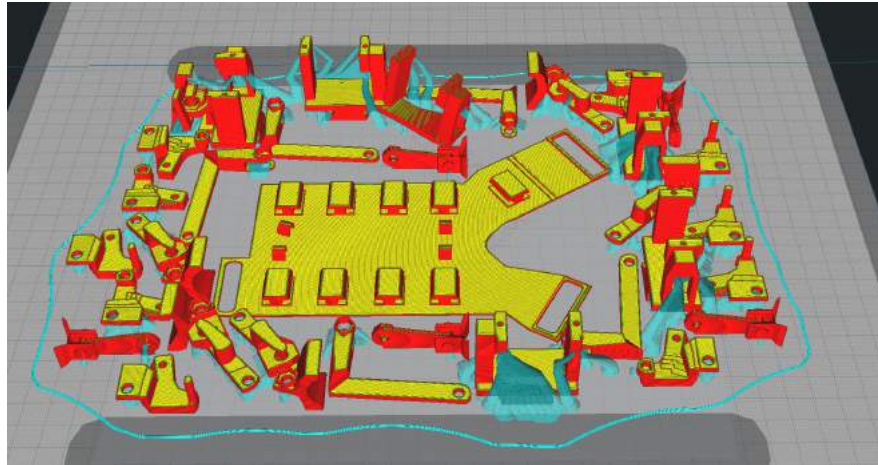
#### 7.3.1 Kosten

Für die Kosten des Handschuhs werden in der Berechnung keine nötigen Arbeitswerkzeuge wie Schraubenzieher und 3D-Drucker miteinberechnet. Die Preise der Hardware für die Evaluation, die in Tabelle 4 dargestellt werden, sind vom Stand des 15.03.2022.

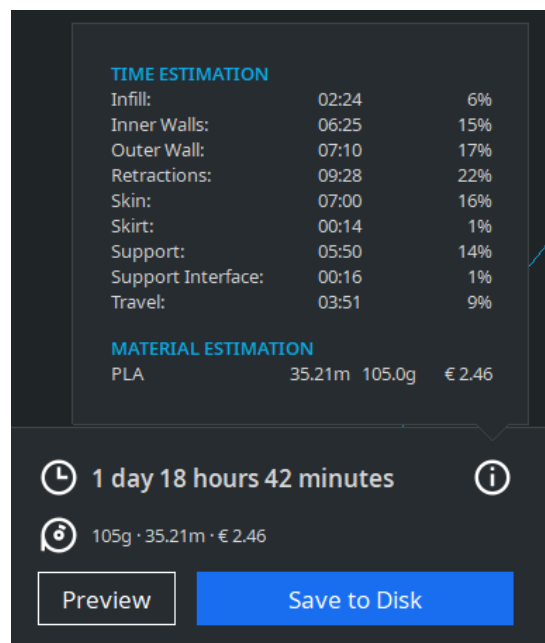
Die Kosten, um einen Handschuh nachzubauen, kann Anhand der Hardware und Materialien erschlossen werden. Zunächst einmal zum Filament, welches gebraucht wird, um einen Handschuh drucken zu können. Die Preise für die Hardware sind in Tabelle 4 aufgelistet. Die Kosten an Filament, wenn alle Teile für einen Handschuh, inklusive Supports, gedruckt werden, erschließt sich, durch das Gewicht von knapp unter 105 Gramm, auf 2.46€, wie in Abb. 63 (b) dargestellt wird. Für die teuerste Variation des Handschuhs, in dem ausschließlich Bourns und Wifi als Kommunikation genutzt werden, kommt ein Preis von 134.64€ für zwei Handschuhe zustande. Für die günstigste Variante, in der 20 Bourns-Sensoren und zehn große Potentiometer benötigt werden, ist der Preis bei 127.50€. Der teuerste Anteil am Preis sind die Bourns-Sensoren, die mehr als 1/3 des Preises ausmachen.

Somit ist der anvisierte Preis von maximal 90€ für ein Paar nicht erreicht, ohne die Menge an Sensoren und damit zusammenhängende Trackingqualität zu verringern. Flex Sensoren sind noch teurer als Bourns, da diese für zwei Handschuhe bereits 75.10€ kosten. Zudem müssen beim Gebrauch von Flex Sensoren ebenso Alpha genutzt werden, die das Spreizen der Finger tracken. Auch IMUs mit Beschleunigungs- und Gyrosensoren sind mit 48€ für ein Handschuhpaar von den Kosten ziemlich nahe an den 20 Bourns mit zehn großen Potentiometern. Nur Hall Sensoren können mit 18.71€ inkl. Magnete eine wahre günstige Alternative sein, sofern hierfür ein funktionierendes Design gefunden wird. Personen mit sehr großen Händen und dicken Fingern müssten mit 30 großen Potentiometern dennoch 94.85€ zahlen.

Mit einem Blick auf den Markt ist ersichtlich, dass kommerzielle Handschuhe dennoch ab \$249 beginnen und, wenn im präsentierten Handschuhdesign Vibrationsmotoren eingebaut sind, das haptischen Feedback des Handschuhdesigns sogar den günstigsten Handschuh übertrumpfen. Fraglich ist nur, ob ein Eigenbau überhaupt eine Alternative ist. So wurden, nachdem Probanden ein Design anzogen und Objekte zusammenbauten, gefragt, ob für sie in Betracht kommt, so ein Design nachzubauen. Die Ergebnisse sind in Tabelle 5 abgebildet. Von den sechs befragten gaben drei



(a) Slice in *Cura* für alle Objekte, die für eine Hand gedruckt werden müssen.



(b) Dauer und Kosten für alle Objekte einer Hand.

Abbildung 63: Informationen und Slice aus dem Slicerprogramm *Cura*.

an, dass sie sich eher einen selbst bauen würden, wenn der Eigenbau etwa 1/3 der kommerziellen Handschuhe kosten würde.

### 7.3.2 Tragekomfort

Um den Tragekomfort zu bewerten, wurden sechs Personen im Alter zwischen 19 und 30 qualitativ befragt. Diese Umfrage basiert auf einer Version des Handschuhs, der in Abb. 42 (b) dargestellt wird. In dieser Handschuhversion wurde noch ein dickes Shield genutzt sowie die Ringkonstruktion aus Abb. 44. Aufgrund dessen, dass bis zu diesem Zeitpunkt die Fingerkomponenten nur in einer Größe vorhanden waren, mussten Probanden eine bestimmte Größe der Finger besitzen. Aufgrund der Corona-

Produkt	Menge	Preis pro Stück	Kosten für benötigte Menge	Quelle
Bourns 301-56-540	30	1.20€	51.77€ (inkl. Versand)	[4]
Bourns 301-56-540	20	1.20€	44.63€ (inkl. Versand)	[4]
Große Potentiometer	10	0.60€	5.99€	[38]
ESP32	3	8€	23.99€	[10]
SG90 Servomotor	10	2.30€	22.99€	[9]
Jumper Kabel	120	0.05€	10.99€	[85]
1kg PLA Filament	180g	23.39€	4.92€	[26]
Vibrationsmotoren	10	1.77	17.78€	[5]
Handschuhe	2	-	13.99€	[82]
Flex Sensor	10	7.51€	75.10€	[7]
Hall Sensoren	20	0.46€	9.22€	[1]
IMUs mit Besch.- und Gyrosensor	20	2.40€	48€	[11]
Magnete	20	0.47€	9.49€	[43]

Tabelle 4: Kostentabelle verwendbarer Hardware.

Pandemie sowie der benötigten kleinen Finger, da die Ringe bis dahin nur für kleine Finger geschaffen waren, ist damit die Menge an Probanden sehr gering ausgefallen.

Um den Tragekomfort zu bewerten, haben die Probanden das Gerüst angezogen und sollten sich anschließend für zwei Minuten mit ihren Händen bewegen und echte Objekte greifen. Nach dieser Zeit wurden sie befragt, wie sie das Tragekomfort bewerten. Die Bewertungen basieren auf deutschen Schulnoten, wodurch die Note 1 *Sehr gut* und die Note 6 *Ungenügend* entsprechen.

Die Probanden aus Tabelle 5 gaben an, dass der Tragekomfort durchschnittlich befriedigend ist. Dies liegt daran, dass die gedruckten Objekte unter anderem kratzig waren. Zwei Probanden hatten dadurch auch sehr geringe Schmerzen. Die Probanden gaben zudem an, dass sie das Design ein wenig einschränkte in ihrer Beweglichkeit. Die Einschränkungen der Finger, die Proband 2 berichtete, wurde allerdings durch die für ihn zu kleinen Ringe an zwei Fingern hervorgerufen. Vier der Probanden würden einen Handschuh bevorzugen, die zwei anderen würden ein dünneres Shield, wie die Version 3, bevorzugen.

Die geäußerte Kritik wurde in späteren Designs implementiert, wodurch ein Handschuh genutzt werden kann, sowie eine dünneres Shield. Zudem wurde die Menge an Kabeln sowie das Gewicht reduziert.

### 7.3.3 Trackingqualität

Die Trackingqualität des Handschuhdesigns kann leider nicht innerhalb eines Grades korrekt sein. Dies liegt daran, dass während dem 3D-Druck immer eine geringe Ungenauigkeit entsteht. Diese Ungenauigkeit ist der Grund, weswegen die Formen der Klammern, die durch die Bourns gehen, nicht exakt an den Masen des Lochs gleichen, sondern ein klein wenig kleiner skaliert sind. Dies bedeutet, dass sehr geringe Drehun-



Probandennummer	1	2	3	4	5	6
Tragekomfort des Handschuhs	4	4	2	2	3	3
Schmerzen	1	1	1	2	2	1
Freiheit der Hand	2	4	1	1	3	3
Shield	Handschuh	Handschuh	Handschuh	Shield V3	Handschuh	Shield V3
Bewertung des handw. Geschicks	2	3	4	3	3	1
Schwierigkeit Einschrauben in Pot.-Halterung	3	3	1	1	3	3
Schwierigkeit des Zusammenschraubens	2	2	1	1	2	1
Kritik	Schwer, kratzig	Kratzig, Kabel	Klett	Groß	Anpassbarkeit	Kabel
Nutzer von XR	VR	VR	Nein	VR	nein	VR
Kaufen oder bauen	Kaufen	Kaufen	Bauen	Kaufen	Bauen	Bauen
Prioritäten bei DIY Projekten	Preis Funktion Konkurrenz	Preis Funktion	Preis Verfügbare Teile Aufwand	Source Code Dokumentation	Simpler Aufb. Verfügbare Teile	Konkurrenz Reparierbarkeit einf. Aufbau

Tabelle 5: Umfrage übers Design, den Selbstbau und Nutzung.

gen von unter einem Grad nicht einmal den Widerstand im Sensor bewegen. Hinzu kommt, dass das allgemeine Rauschen der Sensordaten zwar gefiltert werden kann, aber man sich immer nur den wahren Messwerten annähern kann, diesen jedoch nie erreicht. Das Rauschen im Zusammenhang der nicht messbaren Rotation kann demnach eine Ungenauigkeit von über einem Rotationsgrad zur Folge haben. Jedoch kann davon ausgegangen werden, dass die Ungenauigkeit nicht über drei Rotationsgrade hinaus geht.

### 7.3.4 Haptisches Feedback

Das haptische Feedback konnte leider nicht durch Nutzertests überprüft werden, da zur Zeit der Nutzertests der Handschuh technische Probleme aufwies. Dies war zurückzuführen auf verschiedene Wackelkontakte, weshalb die Daten der Sensoren unbrauchbar waren. Die Anzahl der Kabel wurden daraufhin im Design des *Shields V3* in Abb. 47 drastisch verringert.

### 7.3.5 Reparierbarkeit

Da die Einzelteile gegebenenfalls vom Nutzer selbst gedruckt und gelötet wurden, können diese einfach ausgetauscht werden. Das Shield bietet hierbei Schiebeleisten, mit der die kaputten Objekte einfach ersetzt werden können. Einzig die Fingerkomponenten, wenn sie an den Handschuh geklebt wurden, sind problematisch. Hierzu wird allerdings das Ringdesign mit und ohne Handschuh angeboten, wodurch der Nutzer selbst entscheiden kann, was für eine Stufe der Reparierbarkeit er nutzen möchte. Jedoch ist die Funktion der Schiebeleisten ebenso auf den Fingern für die Fingerkomponenten in Betracht zu ziehen, um diese ohne direktes Ankleben leichter reparierbar zu machen.

### 7.3.6 Komplexität der Konstruktion

Um die Komplexität für das Bauen des Handschuhs zu erforschen, wurden die sechs Personen, die bereits das Tragekomfort beurteilt haben, ebenso gefragt, Teile des Handschuhs nachzubauen. Diese Umfrage basiert auf den Handschuh, der in Abb. 42 (b) dargestellt wird. Es wurden ausschließlich die für den Bau benötigten Werkzeuge bereitgestellt und keine Anleitung verteilt. Die Probanden durften sich den Handschuh während des Tests anschauen. Fragen wurden nicht beantwortet. Zudem wurde keine Zeit gemessen.

Zuerst wurde die Probanden gefragt, wie sie ihr handwerklichen Geschick beurteilen. Die Probanden sollten zunächst einen Alpha in eine Potentiometer-Halterung, wie er links in Abb. 37 zu sehen ist, mit einer Mutter, einschrauben. Anschließend sollten sie die Fingerstücke über die Brücken verbinden und mit Schrauben befestigen. Die Ergebnisse aus dem Probandentest sind in Tabelle 5 dargestellt. Die Fragen wurden nach der Anprobe des Handschuhs gestellt.

Ersichtlich ist, dass keiner der Probanden das Zusammenbauen der Handschuhteile für überaus schwierig empfand. Selbst die Kontermuttern wurden jeweils eigenständig angezogen und genutzt. Angemerkt wurde jedoch, dass das Design durchdacht erscheint. Für die Drehplatte im Potentiometer-Holder, in dem die Mutter seinen Platz findet zum Festschrauben des Potentiometers, wurde zwei Mal angemerkt, dass ein Pfeil, der die Drehrichtung für das Einschrauben bestimmt, hilfreich wäre.

## 7.4 Evaluation der Software

Die Software kann anhand der Verzögerung des Datentransfers sowie der Aktualisierungsrate gemessen werden. Die Transferdauer über die direkte Wifi-Kommunikation mithilfe *ESPnow* wurde gemessen, indem zwei ESP32 die Uhrzeit vom Zeitserver *pool.ntp.org* wiederholend lesen. Der ESP32 des Handschuhs verarbeitet anschließend die Daten der Sensoren und sendet anschließend die Daten an den Dongle, der die Handschuhdaten verarbeitet, diese an UE sendet und dann die Uhrzeit des Handschuhs und des Dongles im Serial Port ausgibt. Die hinteren drei Zahlen, die in Abb. 64 dargestellt werden, zeigen die Millisekunden der Uhrzeit an. Es ist ersichtlich, dass die Übertragung und Verarbeitung zwischen 60 und 80 Millisekunden benötigt. Die Übertragung zwischen zwei ESP32 über *ESPnow* ist allerdings vermutlich um ein vielfaches geringer als in dieser Messung ersichtlich ist. In einer Arbeit, in der über *ESPnow* 307 Bytes unkomprimiert übertragen wurden, konnte eine durchschnittliche Übertragungslatenz von 4 Millisekunden gemessen werden.[137] (S. 111)

In UE kann die Leistung des Blueprints, welches die Daten empfängt, an der Aktua-

```
Zeit der Hand: 1647451877803
Zeit des Dongles: 1647451878068
Zeit der Hand: 1647451877806
Zeit des Dongles: 1647451878074
Zeit der Hand: 1647451877810
Zeit des Dongles: 1647451878079
Zeit der Hand: 1647451877813
Zeit des Dongles: 1647451878085
Zeit der Hand: 1647451877824
Zeit des Dongles: 1647451878090
Zeit der Hand: 1647451877836
Zeit des Dongles: 1647451878096
Zeit der Hand: 1647451877848
Zeit des Dongles: 1647451878101
Zeit der Hand: 1647451877851
```

Abbildung 64: Timings des Handschuhs und Dongles, nachdem Daten der Finger gesammelt, zusammen gesetzt, über Wifi übertragen, entschlüsselt und dann an UE versendet wurden. Die letzten drei Stellen zeigen die Millisekunden. Es handelt sich um den UNIX Zeitstempel in Millisekunden, der die Zeit seit dem 01.01.1970 misst. [80]

lisierungsrate gemessen werden. Die Leistung der Blueprints kann, je nach Leistung der Hardware im Computer, unterschiedlich sein. Die Aktualisierungsrate ist jedoch maximal von der *Baud Rate* begrenzt. Für dieses Projekt wurde stets eine *Baud Rate* von 115200 genutzt. Die *Baud Rate* gibt an, wie oft ein Bit innerhalb eines Kanals gewechselt werden kann.

Auf einem *Ryzen 7 2700X* mit einer *GTX 1080* von *KFA2* und 32 Gigabyte Arbeitsspeicher mit 3200 MHz Geschwindigkeit in einer leeren Szene kann eine Aktualisierungsrate von 200 Herz im Editor erreicht werden, ohne eine sinkende Framerate zufolge zu haben. Da Blueprints im Editor ohne *Nativizing*, wie in Kapitel 6.3.2 beschrieben wird, sehr langsam sind, ist die Nutzung der Blueprints bei dieser Aktualisierungsrate nicht problematisch. Für eine Bluetooth-Kommunikation wird jedoch empfohlen, diese auf 60 Herz zu stellen. Mit der Wifi-Kommunikation sind 200 Herz möglich. Allerdings teilen sich diese Herz auf die verwendete Anzahl von Handschuhen auf. Wenn entsprechend zwei Handschuhe mit einem Dongle genutzt werden, ist die Aktualisierungsrate pro Handschuh durchschnittlich 100 Herz. Mit zwei Dongles, und entsprechend zwei seriellen Ports, kann für beide Hände eine Aktualisierungsrate von 200 Herz erzielt werden. Der Flaschenhals ist jedoch die Anzahl der ausführbaren Kommunikationen über denselben seriellen Port.

## 7.5 Problematiken

Aufgrund dessen, dass die Implementierung der Software nur innerhalb UE umgesetzt wurde, ist die Nutzung zunächst stark begrenzt. Zudem kann der Handschuh dadurch noch keine VR Spiele unterstützen, da keine übergreifende Implementierung vorhanden ist. Eine Implementierung in der Software *OpenGlove* [51] steht bisher noch aus. *OpenGlove* ist die Software, die für den *LucidGlove* entwickelt wurde. Der Support für mehrere Sensoren pro Finger sowie Fingerspreizen wurde dort erst am 24.01.2022 hinzugefügt.[52] Dadurch kann jeder Sensor des in dieser Arbeit beschriebenen Designs von *OpenGloves* genutzt werden. Die Implementierung in *OpenGloves* ist bis Mai 2022 geplant.

## 8 Zusammenfassung und Ausblick

In diesem letzten Kapitel wird die Arbeit kurz zusammengefasst und mit einem Fazit beendet. Anschließend werden weiterführende Arbeiten vorgeschlagen sowie die Zukunft des Projekts beschrieben.

### 8.1 Zusammenfassung

In der vorliegenden Arbeit wird beschrieben, wie ein Motion Capture Handschuh mit haptischem Feedback designt und implementiert wurde, der für jeden Finger drei Sensoren besitzt, um deren Position zu tracken. Das Design des Handschuhs wurde durch eine Vielzahl an Iterationen laufend verbessert, wodurch ein modularer Aufbau entstanden ist. Damit können Nutzer die Teile verwenden, die sie für ihre eigenen Zwecke als notwendig empfinden. Das Tracking der Hand kann über jedes System umgesetzt werden, da ein Adapter für das Anschließen von diversen VR Controllern und Trackern beigelegt und angeschlossen werden kann. Dadurch ist es egal, ob die Handschuhe für Motion Capture der Finger oder VR-Spiele genutzt wird. Der Handschuh bietet eine hohe Reparaturfreundlichkeit, die durch den Eigenbau und die Notwendigkeit, nichts festkleben zu müssen, gegeben ist. Nutzer können demnach, wenn Objekte brechen, diese selbst ersetzen. Auch die Auswahl der kabellosen Kommunikation, zwischen Handschuh und Computer, kann zwischen Bluetooth und dem stärkeren Wifi ausgesucht werden. Die Implementierung wurde in der Spiele-Engine *Unreal Engine 4* umgesetzt. Dort werden die Daten des Handschuhs auf das Rig übertragen sowie Daten für Force Feedback erzeugt. Über eine Wifi-Verbindung in Zusammenhang eines USB-Dongles pro Handschuh beträgt die Aktualisierungsrate in *Unreal Engine 4* 200 Herz. Mithilfe des Force Feedbacks kann der Nutzer aufgrund des Handschuhs die Präsenz von Objekten beim Greifen spüren, da Servomotoren die Finger davon abhalten, sich weiter zu schließen. Die Kosten für ein paar Handschuhe beträgt in seiner teuersten Konstellation 134.64€. Dadurch, dass der Handschuh und seine Daten nicht proprietär verschlüsselt sind, kann dieser zudem auch als Interface für Robotik genutzt werden. Durch den Anschluss von Joysticks und weiteren Knöpfen kann das Button-Layout von VR Controllern simuliert werden, um Aktionen in Spielen, die über dem Greifen hinaus gehen, mit dem vorgestellten Design zu ermöglichen. Die Kosten des Handschuhs sind weit unter den Kosten von kommerziellen Handschuhen, die mit

weniger Features und einem Preis von 250€ beginnen. Daher ist der Handschuh eine wahre Alternative zu kommerziellen Produkten.

## 8.2 Ausblick

Eine Weiterführung dieser Thesis könnte sich beispielsweise mit einer Implementierung von Vibrationsmotoren beschäftigen, die einem höherem Zweck dienen als der in Kapitel 6.8 beschriebenen binären, oder abhängig von der Entfernung zu Oberflächen steigenden Vibrationsstärke. So kann die Simulation von Effekten, wie eine elektrische Ladung, wie sie in *Stratos* Systemem möglich sind, erforscht werden. Desweiteren ist mit dem aktuellen Handschuh nur möglich, feste Objekte zu simulieren. Eine Simulation von weichen Objekten ist damit nicht möglich. Dies würde einher gehen mit einem System, welches auch in den Controller der *Playstation 5* enthalten sind. Dies wäre ebenso ein Anhaltspunkt, um das haptische Feedback des Handschuhs zu verbessern.

Mit dem Abschluss dieser Arbeit endet allerdings nicht der Support für das Projekt. Wie in Kapitel 7.5 beschrieben, wird bis Mai 2022 der Handschuh in der Software *OpenGloves* implementiert. Weitere Verbesserungen, wie das Hinzufügen von Sensoren, die die Stärke des Griffs messen, um z. B. das Zerdrücken von Dosen in VR-Spielen zu ermöglichen, sowie eine Verbesserung der Kraftübertragung der Servomotoren für das Force Feedback, sind in Planung. Der weitere Verlauf des Projekts kann auf der Plattform *GitHub* im Repository *VRonGlove-Project* verfolgt werden.[77]







## Literaturverzeichnis

- [1] 20 x 49e oh49e ss49e linear-sensoren s49e to-92 3 pins sensorelement schalter für elektrofahrzeuge motorreparatur. Accessed on: Mar. 20. 2022, <https://www.amazon.de/gp/product/B08LD4BBFR>.
- [2] 2701 patent term [r-10.2019]. Accessed on: Mar. 3. 2022, <https://www.uspto.gov/web/offices/pac/mpep/s2701.html>.
- [3] 3382 - 12 mm rotary position sensor. Accessed on: Mar. 3. 2022, <https://www.mouser.de/datasheet/2/54/3382-776135.pdf>.
- [4] 3382g-1-252g - drehpotentiometer 2.5kohm smd, bourns. Accessed on: Mar. 20. 2022, <https://www.distrelec.de/de/drehpotentiometer-5kohm-smd-bourns-3382g-252g/p/30156540?track=true&no-cache=true&marketingPopup=false>.
- [5] 5 stücke dc 3 v 80ma mini vibrationsmotoren 11000 rpm 1020 flache münze taste-typ micro dc vibrationsmotor für handy und pager(27mm). Accessed on: Mar. 20. 2022, <https://www.amazon.de/st%C3%BCcke-Vibrationsmotoren-Flache-Taste-Typ-Vibrationsmotor/dp/B0839H19HY>.
- [6] Ab-004 - understanding erm vibration motor characteristics. Accessed on: Feb. 13. 2022, <https://www.precisionmicrodrives.com/ab-004>.
- [7] Adafruit short flex/bend sensor [ada1070]. Accessed on: Mar. 20. 2022, <https://www.amazon.de/gp/product/B00SK8PZVS>.
- [8] Aiangelive. Accessed on: Feb. 07. 2022, <https://www.twitch.tv/aiangelive>.
- [9] Allomn 10 stücke micro servo flugzeug servo hubschrauber servo lot hohe ehrgeizige sg90 9g mini micro servo für rc hubschrauber flugzeug auto fahrzeugmodelle. Accessed on: Mar. 20. 2022, <https://www.amazon.de/ALLOMN-Flugzeug-Hubschrauber-Ehrgeizige-Fahrzeugmodelle/dp/B07Q1GJJZS>.
- [10] Azdelivery 3 x esp32 nodemcu module wlan wifi dev kit c development board mit cp2102 (nachfolgermodell zum esp8266) kompatibel mit arduino und inklusive e-book! Accessed on: Mar. 20. 2022, [https://www.amazon.de/AZDelivery-NodeMCU-Development-Nachfolgermodell-ESP8266/dp/B074RGW2VQ/ref=sr\\_1\\_3?keywords=esp32&qid=1647877288&sr=8-3](https://www.amazon.de/AZDelivery-NodeMCU-Development-Nachfolgermodell-ESP8266/dp/B074RGW2VQ/ref=sr_1_3?keywords=esp32&qid=1647877288&sr=8-3).
- [11] Azdelivery 5 x gy-521 mpu-6050 3-achsen-gyroskop und beschleunigungs-sensor kompatibel mit arduino und raspberry pi inklusive e-book! Acces-

- sed on: Mar. 20. 2022, <https://www.amazon.de/AZDelivery-MPU-6050-3-Achsen-Gyroskop-Beschleunigungssensor-Arduino/dp/B07NP2WF7J>.
- [12] Azure kinect dk. Accessed on: Feb. 8. 2022, <https://azure.microsoft.com/en-us/services/kinect-dk/#faq>.
- [13] Blueprint visual scripting. Accessed on: Dez. 02. 2021, <https://docs.unrealengine.com/4.27/en-US/ProgrammingAndScripting/Blueprints/>.
- [14] Cobra gloves high-end gloves for robotics, animation, virtual reality, medical and bio-mechanics research. Accessed on: Feb. 12. 2022, <https://www.synertial.com/cobra-gloves>.
- [15] Codemiko. Accessed on: Feb. 14. 2022, <https://www.twitch.tv/codemiko>.
- [16] A complete motion capture solution. Accessed on: Feb. 10. 2022, <https://neuronmocap.com/perception-neuron-studio-system>.
- [17] Cyberforce. Accessed on: Feb. 13. 2022, <http://www.cyberglovesystems.com/cyberforce>.
- [18] Cyberglove iii. Accessed on: Feb. 13. 2022, <http://www.cyberglovesystems.com/cyberglove-iii>.
- [19] Cybergrasp. Accessed on: Feb. 13. 2022, <http://www.cyberglovesystems.com/cybergrasp>.
- [20] Cybergrasp. Accessed on: Feb. 13. 2022, [https://static1.squarespace.com/static/559c381ee4b0ff7423b6b6a4/t/5602fc01e4b07ebf58d480fb/1443036161782/CyberGrasp\\_Brochure.pdf](https://static1.squarespace.com/static/559c381ee4b0ff7423b6b6a4/t/5602fc01e4b07ebf58d480fb/1443036161782/CyberGrasp_Brochure.pdf).
- [21] Die knochen der menschlichen hand. Accessed on: Feb. 27. 2022, <https://www.ksw.ch/gesundheitsthemen/handknochen/>.
- [22] Espruino Repository. Accessed on: Nov. 27. 2021, <https://github.com/espruino/espruino>.
- [23] Find the right tactsuit x for you. Accessed on: Feb. 12. 2022, <https://www.bhaptics.com/tactsuit/compare>.
- [24] Flex 3 datasheet. Accessed on: Jan. 03. 2022, <https://d111srqycjesc9.cloudfront.net/Flex%203%20Data%20Sheet.pdf>.
- [25] FORTH Arduino / ESP32. Accessed on: Nov. 27. 2021, <https://arduino-forth.com/>.
- [26] Giantarm petg filament 1,75 mm schwarz für 3d-drucker 1 kg spool. Accessed on: Mar. 20. 2022, <https://www.amazon.de/GIANTARM-PETG-Filament-3D-Drucker-schwarz/dp/B08C51YZH7>.

- [27] Haptic workstation. Accessed on: Feb. 12. 2022, <http://www.cyberglovesystems.com/haptic-workstation>.
- [28] Haptx gloves dk2. Accessed on: Feb. 10. 2022, <https://haptx.com/technology/>.
- [29] Haptx statement on meta microfluidic glove prototype. Accessed on: Feb. 13. 2022, <https://haptx.com/meta-glove-prototype-response/>.
- [30] Hatsune miku v4x. Accessed on: Feb. 14. 2022, <https://sonicwire.com/product/virtualsinger/special/mikuv4x?lang=en>.
- [31] Hi5 vr glove. Accessed on: Feb. 01. 2022, <https://hi5vrglove.com/#slideshow-0>.
- [32] How do you make a virtual reality glove? - smarter every day 191. Accessed on: Feb. 10. 2022, <https://youtu.be/s-HAsxt9pV4?t=286>.
- [33] How does an mr damper work? Accessed on: Feb. 13. 2022, <https://www.lord.com/products-and-solutions/active-vibration-control/industrial-suspension-systems/how-does-mr-damper-work>.
- [34] Index controller. Accessed on: Feb. 14. 2022, <https://www.valvesoftware.com/de/index/controllers>.
- [35] Industrial-grade haptic technology. Accessed on: Feb. 10. 2022, <https://haptx.com/>.
- [36] Interfacing flex sensor with arduino. Accessed on: Jan. 03. 2022, <https://lastminuteengineers.com/flex-sensor-arduino-tutorial/>.
- [37] Introduction to C++ Programming in UE4. Accessed on: Dez. 01.2021, <https://docs.unrealengine.com/4.27/en-US/ProgrammingAndScripting/ProgrammingWithCPP/IntroductionToCPP/>.
- [38] Laomao 10pcs b10k 3pin 16mm 10k ohm potentiometer mono dreh poti. Accessed on: Mar. 20. 2022, [https://www.amazon.de/gp/product/B00H969CM0/ref=ppx\\_yo\\_dt\\_b\\_asin\\_title\\_o00\\_s00?ie=UTF8&psc=1](https://www.amazon.de/gp/product/B00H969CM0/ref=ppx_yo_dt_b_asin_title_o00_s00?ie=UTF8&psc=1).
- [39] Lucas de bonet. Accessed on: Feb. 13. 2022, <https://www.linkedin.com/in/ldebonet/>.
- [40] lucas vrtech. Accessed on: Feb. 12. 2022, [https://www.tiktok.com/@lucas\\_vrtech](https://www.tiktok.com/@lucas_vrtech).
- [41] Lucas vrtech. Accessed on: Feb. 13. 2022, <https://www.youtube.com/channel/UCjWBEvwq3iNLXEdrGTR5S9A>.
- [42] Lucidgloves. Accessed on: Feb. 13. 2022, <https://github.com/LucidVR/lucidgloves>.

- [43] Magenesis neodym magnete 10x2 mm 52 stück mini magnete extrem stark ca. 2 kilo haftstärke. Accessed on: Mar. 20. 2022, <https://www.amazon.de/gp/product/B06X977K8L>.
- [44] Magnetorheologische flüssigkeiten. Accessed on: Feb. 14. 2022, <https://www.cesma.de/de/materialien/smart-fluessigkeiten/magnetorheologische-fluessigkeiten.html>.
- [45] Manus prime x haptic vr. Accessed on: Feb. 12. 2022, <https://www.manus-meta.com/haptic-gloves>.
- [46] Motion capture. Accessed on: Feb. 10. 2022, <https://www.xsens.com/motion-capture>.
- [47] Motion capture - the data arena has an optitrack motion capture system. Accessed on: Mar. 17. 2022, <https://dataarena.net/projects/motion-capture>.
- [48] Mvn animate. Accessed on: Feb. 10. 2022, <https://www.xsens.com/products/mvn-animate?hsCtaTracking=0031f976-823a-4074-8cc4-d6f2347422ae%7C584bb7ed-596e-4dd6-992d-245825acf04f>.
- [49] Nativizing blueprints. Accessed on: Dez. 01. 2021, <https://docs.unrealengine.com/4.27/en-US/ProgrammingAndScripting/Blueprints/TechnicalGuide/NativizingBlueprints/>.
- [50] Notch pioneer. Accessed on: Feb. 09. 2022, <https://wearnotch.com/order?kitnum=3>.
- [51] Opengloves. Accessed on: Feb. 14. 2022, <https://store.steampowered.com/app/1574050/OpenGloves/>.
- [52] Opengloves v0.5 update. Accessed on: Feb. 14. 2022, <https://steamcommunity.com/games/1574050/announcements/detail/3123810455817692445>.
- [53] Openpose. Accessed on: Feb. 8. 2022, <https://github.com/CMU-Perceptual-Computing-Lab/openpose>.
- [54] Optitrack - build your own system. Accessed on: Jan. 03. 2022, <https://optitrack.com/systems/>.
- [55] Pascal-scm for Atmel AVR. Accessed on: Dez. 02. 2021, [https://www.e-lab.de/AVRco/index\\_en.html](https://www.e-lab.de/AVRco/index_en.html).
- [56] Perception neuron. Accessed on: Feb. 10. 2022, <https://neuronmocap.com/>.
- [57] Plexusvr. Accessed on: Feb. 01. 2022, <http://plexus.im/>.
- [58] Prime x. Accessed on: Jan. 13. 2022, <https://www.manus-meta.com/haptic-gloves>.

- [59] Prime x marker mocap. Accessed on: Feb. 12. 2022, <https://www.manus-meta.com/mocap-products>.
- [60] Rokoko - the future is in your hands. Accessed on: Feb. 01. 2022, <https://www.rokoko.com/products/smartgloves>.
- [61] Rv16af20xx. Accessed on: Mar. 3. 2022, <https://asset.conrad.com/media10/add/160267/c1/-/da/001694306DS01/datenblatt-1694306-alpha-rv16af20kb100km-dreh-potentiometer-mono-200-mw-100-k-1-st.pdf>.
- [62] Scripting the Editor using Python. Accessed on: Dez. 03. 2021, <https://docs.unrealengine.com/4.26/en-US/ProductionPipelines/ScriptingAndAutomation/Python/>.
- [63] Sg90 9 g micro servo. Accessed on: Mar. 3. 2022, <https://datasheetspdf.com/pdf-file/791970/TowerPro/SG90/1>.
- [64] Shockwave. Accessed on: Feb. 12. 2022, <https://www.shockwavesuit.com/>.
- [65] Shockwave. Accessed on: Feb. 12. 2022, <https://www.kickstarter.com/projects/1110033869/shockwave/description>.
- [66] Sketch build process. Accessed on: Nov. 27. 2021. URL: <https://arduino.github.io/arduino-cli/0.20/sketch-build-process/>.
- [67] SkookumScript | Superpowered game development. Accessed on: Nov. 28. 2021, <https://skookumscript.com/>.
- [68] Stratos explore. Accessed on: Feb. 9. 2022, <https://www.ultraleap.com/product/stratos-explore/>.
- [69] Stratos explore datasheet. Accessed on: Feb. 8. 2022, [https://www.ultraleap.com/datasheets/STRATOS\\_Explore\\_Development\\_Kit\\_datasheet.pdf](https://www.ultraleap.com/datasheets/STRATOS_Explore_Development_Kit_datasheet.pdf).
- [70] Stratos inspire datasheet. Accessed on: Feb. 9. 2022, [https://www.ultraleap.com/datasheets/STRATOS\\_Inspire\\_Datasheet.pdf](https://www.ultraleap.com/datasheets/STRATOS_Inspire_Datasheet.pdf).
- [71] The suit. Accessed on: Feb. 13. 2022, <https://teslasuit.io/the-suit/>.
- [72] SwiftX Cross Compilers for Embedded Systems Applications. Accessed on: Nov. 27. 2021, <https://www.forth.com/embedded/>.
- [73] The best controller for ar/vr. Accessed on: Jan. 12. 2022, <https://senso.me/>.
- [74] Touch the untouchable. Accessed on: Jan. 13. 2022, <https://www.dextarobotics.com/>.

- [75] UnrealEngine Python Repository. Accessed on: Dez. 01. 2021, <https://github.com/20tab/UnrealEnginePython>.
- [76] Unreal.js Repository. Accessed on: Dez. 03. 2021, <https://github.com/ncsoft/Unreal.js>.
- [77] Vron-project. Accessed on: Mar. 17. 2022, <https://github.com/MrMarans/VRon-Project>.
- [78] We remade the jurassic park t-rex with modern vfx. Accessed on: Mar. 3. 2022, <https://youtu.be/1UUQDBsV8Gw?t=283>.
- [79] Wearables. Accessed on: Feb. 10. 2022, <https://optitrack.com/accessories/wear/>.
- [80] What is unix time? Accessed on: Mar. 20. 2022, <https://kb.narrative.io/what-is-unix-time>.
- [81] Why program in FORTH language on ARDUINO? Accessed on: Nov. 27. 2021, [https://arduino-forth.com/article/FORTH\\_whyForth#forthCfaire/](https://arduino-forth.com/article/FORTH_whyForth#forthCfaire/).
- [82] Winter handschuhe herren faux leder warm dick vlies winddicht kälteschutz thermal fäustlinge anti-rutsch berührungssensitiver bildschirm für outdoor fahren camping damen teenager. Accessed on: Mar. 20. 2022, <https://www.amazon.de/gp/product/B07YCLXSGW>.
- [83] World's most advanced haptic solution for training in virtual reality. Accessed on: Jan. 14. 2022, <https://sensorialxr.com/>.
- [84] Xr solutions for your business. Accessed on: Jan. 12. 2022, <https://sensorialxr.com/xrsolutions/>.
- [85] Yxpcars 240 stück jumper kabel kit 10cm and 20cm male zu female, male zu male, female zu female jumper wires steckbrett für arduino. Accessed on: Mar. 20. 2022, <https://www.amazon.de/gp/product/B08HQ7K6M7>.
- [86] Variable stiffness electroactive polymer systems. Accessed on: Feb. 16. 2022, <https://www.freepatentsonline.com/6882086.pdf>, 4 2005.
- [87] New bionics let us run, climb and dance | hugh herr. Accessed on: Feb. 14. 2022, <https://youtu.be/CDsNZJTww?t=428>, 3 2014.
- [88] megaavr data sheet. Accessed on: Dez. 02. 2021, <https://ww1.microchip.com/downloads/en/DeviceDoc/ATmega48A-PA-88A-PA-168A-PA-328-P-DS-DS40002061A.pdf>, 2018.
- [89] A real life haptic glove (ready player one technology today) - smarter every day 190. Accessed on: Feb. 10. 2022, <https://youtu.be/OK2y4Z5IkZ0?t=122>, 3 2018.
- [90] Ultraleap product overview | arrow.com. Accessed on: Feb. 12. 2022, <https://www.youtube.com/watch?v=due9Mco0EBo>, 12 2018.

- [91] The artificial muscles that will power robots of the future | christoph keplinger. Accessed on: Feb. 14. 2022, <https://youtu.be/ER15KmrB8h8?t=428>, 5 2019.
- [92] Dexmo development kit 1. Accessed on: Feb. 12. 2022, [https://oss-main.dextarobotics.com/specifications\\_en-us.pdf](https://oss-main.dextarobotics.com/specifications_en-us.pdf), 5 2019.
- [93] Haptics | how our virtual touch technology works | ultraleap. Accessed on: Feb. 11. 2022, <https://www.youtube.com/watch?v=XnDoej4mnHU>, 10 2020.
- [94] Oculus quest 2 hand tracking | oculus. Accessed on: Feb. 09. 2022, <https://youtu.be/uztFcEA6Rf0?t=38>, 12 2020.
- [95] Ps5 adaptive trigger live mechanics in game - full teardown. Accessed on: Feb. 12. 2022, <https://www.youtube.com/watch?v=8VBZ1upH93w&t=226s>, 11 2020.
- [96] Touching masterpieces. Accessed on: Feb. 12. 2022, [https://www.youtube.com/watch?v=J\\_u1FISc-Zc](https://www.youtube.com/watch?v=J_u1FISc-Zc), 7 2020.
- [97] Design science research methodologie (einfach erklärt). Accessed on: Jan. 12. 2022, <https://shrike.de/design-science-research-methodologie/>, 3 2021.
- [98] Esp32 wroom 32e, esp32 wroom 32ue, datasheet. Accessed on: Dez. 02. 2021, [https://www.espressif.com/sites/default/files/documentation/esp32-wroom-32e\\_esp32-wroom-32ue\\_datasheet\\_en.pdf](https://www.espressif.com/sites/default/files/documentation/esp32-wroom-32e_esp32-wroom-32ue_datasheet_en.pdf), 2021.
- [99] Inside reality labs research: Bringing touch to the virtual world. Accessed on: Feb. 12. 2022, <https://about.fb.com/news/2021/11/reality-labs-haptic-gloves-research/>, 11 2021.
- [100] Ps5 games - adaptive triggers. Accessed on: Feb. 8. 2022, [https://www.youtube.com/watch?v=9\\_AE21WqhU4](https://www.youtube.com/watch?v=9_AE21WqhU4), 7 2021.
- [101] Temperature simulation in vr. Accessed on: Feb. 14. 2022, [https://www.tiktok.com/@lucas\\_vrtech/video/7034982220980882693](https://www.tiktok.com/@lucas_vrtech/video/7034982220980882693), 11 2021.
- [102] Training and optimizing a 2d pose estimation model with nvidia tao toolkit, part 1. Accessed on: Feb. 04. 2022, <https://developer.nvidia.com/blog/training-optimizing-2d-pose-estimation-model-with- tao-toolkit-part-1/>, 6 2021.
- [103] Vr haptic glove prototype 4. Accessed on: Feb. 13. 2022, [https://www.tiktok.com/@lucas\\_vrtech/video/6946994173539126533](https://www.tiktok.com/@lucas_vrtech/video/6946994173539126533), 4 2021.
- [104] Vr haptic glove prototype 5 sensor. Accessed on: Feb. 14. 2022, [https://www.tiktok.com/@lucas\\_vrtech/video/7016739390307241222](https://www.tiktok.com/@lucas_vrtech/video/7016739390307241222), 10 2021.

- [105] Vr haptic glove squishyness!! Accessed on: Feb. 13. 2022, [https://www.tiktok.com/@lucas\\_vrtech/video/7029381747427036422](https://www.tiktok.com/@lucas_vrtech/video/7029381747427036422), 11 2021.
- [106] Hardwarespezifikationen für azure kinect dk. Accessed on: Feb. 8. 2022, <https://docs.microsoft.com/de-de/azure/kinect-dk/hardware-specification>, 1 2022.
- [107] I built \$60 vr haptic gloves to feel virtual reality. Accessed on: Feb. 13. 2022, <https://www.youtube.com/watch?v=ZTzn37Usa-U>, 1 2022.
- [108] Agog Labs (and SkookumScript) Joins Epic Games. Accessed on: Dez. 01. 2021, <https://skookumscript.com/>, January 23, 2019.
- [109] Fadel Adib, Chen-Yu Hsu, Hongzi Mao, Dina Katabi, and Frédo Durand. Capturing the human figure through a wall. *ACM Transactions on Graphics (TOG)*, 34(6):1–13, 2015.
- [110] Fadel Adib and Dina Katabi. See through walls with wifi! In *Proceedings of the ACM SIGCOMM 2013 conference on SIGCOMM*, pages 75–86, 2013.
- [111] Farzad Ahmadkhanlou, Gregory N Washington, Stephen E Bechtel, and Yingru Wang. Magnetorheological fluid based automotive steer-by-wire systems. In *Smart Structures and Materials 2006: Industrial and Commercial Applications of Smart Structures Technologies*, volume 6171, pages 157–168. SPIE, 2006.
- [112] Kizuna AI. Twitter. Accessed on: Feb. 14. 2022, <https://www.youtube.com/watch?v=EoPFGj3uuYo>, 11 2016.
- [113] D. Allen. Resident evil village - lady dimitrescu motion capture behind the scenes. Accessed on: Feb. 11. 2022, <https://www.youtube.com/watch?v=EBI59ZdP6mY>, 5 2021.
- [114] T. Babb. How a kalman filter works, in pictures. Accessed on: Feb. 10. 2022, <https://www.bzarg.com/p/how-a-kalman-filter-works-in-pictures/>.
- [115] B. Baker. Anwendung der sensorfusion auf beschleunigungsmesser und gyroskope. Accessed on: Feb. 10. 2022, <https://www.digikey.de/de/articles/apply-sensor-fusion-to-accelerometers-and-gyroscopes>, 1 30, 2018.
- [116] Manfred Berger and Johan Westermarck. Selspot 3-D motion analysis system. In James S. Walton, editor, *Image-Based Motion Measurement*, volume 1356, pages 51 – 52. International Society for Optics and Photonics, SPIE, 1990.
- [117] T. Bezmalinovic. Oculus quest 2: Kaufen, preis, release, technik alle infos. Accessed on: Feb. 14. 2022, <https://mixed.de/oculus-quest-2-infos/>.
- [118] Jonathan Blake and Hakan B Gurocak. Haptic glove with mr brakes for virtual reality. *IEEE/ASME Transactions On Mechatronics*, 14(5):606–615, 2009.



- [119] A. Boyle. 5 tech twists from 'ready player one,' steven spielberg's film about gaming's past and vr's future. Accessed on: Mar. 17. 2022, <https://www.geekwire.com/2018/5-tech-twists-ready-player-one/>, 3 2018.
- [120] M. W. Brown. New fluid thickens in an electric field. Accessed on: Mar. 3. 2022, <https://www.nytimes.com/1988/05/17/science/new-fluid-thickens-in-an-electric-field.html>.
- [121] Z. Cao. tf-pose-estimation. Accessed on: Feb. 8. 2022, <https://github.com/ZheC/tf-pose-estimation>.
- [122] P. Carbonnelle. PYPL PopularitY of Programming Language. Accessed on: Dez. 02. 2021, <https://pypl.github.io/PYPL.html>.
- [123] P. Carbonnelle. PYPL PopularitY of Programming Language - Germany. Accessed on: Dez. 02. 2021, <https://pypl.github.io/PYPL.html?country=DE>.
- [124] V. Christ. Real-time sensor-based motion capture system for virtual reality. Accessed on: Feb. 09. 2022, [https://kola.opus.hbz-nrw.de/opus45-kola/frontdoor/deliver/index/docId/1464/file/MasterThesis\\_VeraChrist.pdf](https://kola.opus.hbz-nrw.de/opus45-kola/frontdoor/deliver/index/docId/1464/file/MasterThesis_VeraChrist.pdf), 6 2017.
- [125] J. Copenhaver. Vr animation and locomotion systems in lone echo. Accessed on: Jan. 03. 2022, <https://lastminuteengineers.com/flex-sensor-arduino-tutorial/>, 3 2018.
- [126] B. Douglas. Drone control and the complementary filter. Accessed on: Feb. 10. 2022, <https://youtu.be/whSw42XddsU?t=228>, 11 2018.
- [127] B. Douglas. Understanding sensor fusion and tracking, part 2: Fusing a mag, accel, gyro estimate. Accessed on: Feb. 10. 2022, <https://youtu.be/0rlvvYgmTvI?t=660>, 10 22, 2019.
- [128] I. Failes. When motion capture puppets were all the rage. Accessed on: Feb. 23. 2022, <https://beforesandafters.com/2020/07/13/when-motion-capture-puppets-were-all-the-rage/>.
- [129] I. Failes. Recalling total recall. Accessed on: Dec. 28. 2021, <https://www.fxguide.com/afx/featured/recalling-total-recall/#mocap>, 6 2015.
- [130] Max Fleischer. Method of producing moving-picture cartoons., October 9 1917. US Patent 1,242,674.
- [131] R. França. What is rotoscoping? learn all about this animation technique (+ course guide). Accessed on: Mar. 3. 2022, <https://classpert.com/blog/all-about-roscoping>, 6 2020.
- [132] A. Gernemann-Paulsen. Assembler in der arduino-ide: ein einblick. Accessed on: Mar. 3. 2022, [http://www.uni-koeln.de/phil-fak/muwi/ag/praktikum/assembler\\_arduino.pdf](http://www.uni-koeln.de/phil-fak/muwi/ag/praktikum/assembler_arduino.pdf).

- [133] M. Glovart. Animate reality with rotoscoping. Accessed on: Mar. 3. 2022, <https://www.adobe.com/creativecloud/video/discover/roscoping-animation.html>.
- [134] Alan R Hevner, Salvatore T March, Jinsoo Park, and Sudha Ram. Design science in information systems research. *MIS quarterly*, pages 75–105, 2004.
- [135] Gines Hidalgo, Yaadhav Raaj, Haroon Idrees, Donglai Xiang, Hanbyul Joo, Tomas Simon, and Yaser Sheikh. Single-network whole-body pose estimation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, October 2019.
- [136] Walter T Higgins. A comparison of complementary and kalman filtering. *IEEE Transactions on Aerospace and Electronic Systems*, (3):321–325, 1975.
- [137] Tu Ngoc Hoang, Su-Tran Van, and B. D. Nguyen. Esp-now based decentralized low cost voice communication systems for buildings. In *2019 International Symposium on Electrical and Electronics Engineering (ISEE)*, pages 108–112, 2019.
- [138] Texas Instruments. Cd405xb cmos single 8-channel analog multiplexer/demultiplexer with logic-level conversion. Accessed on: Mar. 3. 2022, <https://www.ti.com/lit/ds/symlink/cd4051b.pdf?HQS=dis-mous-null-mouser-mode-dsf-pf-null-wwe>.
- [139] Wenjun Jiang, Hongfei Xue, Chenglin Miao, Shiyang Wang, Sen Lin, Chong Tian, Srinivasan Murali, Haochen Hu, Zhi Sun, and Lu Su. Towards 3d human pose construction using wifi. In *Proceedings of the 26th Annual International Conference on Mobile Computing and Networking*, pages 1–14, 2020.
- [140] Z. Johnson. 20 fun facts about snow white and the seven dwarfs on its 80th anniversary. Accessed on: Feb. 25. 2022, <https://www.eonline.com/news/901665/20-fun-facts-about-snow-white-and-the-seven-dwarfs-on-its-80th-anniversary>.
- [141] S. D. Jong. Blueprints in-depth - part 1 | unreal fest europe 2019 | unreal engine. Accessed on: Nov. 28. 2021, <https://www.youtube.com/watch?v=j6mskTgL7kU>.
- [142] Rudolph Emil Kalman. A new approach to linear filtering and prediction problems. DOI: 10.1115/1.3662552, 1960.
- [143] Christoph Keplinger, Martin Kaltenbrunner, Nikita Arnold, and Siegfried Bauer. Röntgen's electrode-free elastomer actuators without electromechanical pull-in instability. *Proceedings of the National Academy of Sciences*, 107(10):4505–4510, 2010.
- [144] Brian Knep, Craig Hayes, Rick Sayre, and Tom Williams. Dinosaur input device. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 304–309, 1995.
- [145] Julian Kreimeier, Sebastian Hammer, Daniel Friedmann, Pascal Karg, Clemens Bühner, Lukas Bankel, and Timo Götzelmann. Evaluation of different types of

- haptic feedback influencing the task-based presence and performance in virtual reality. In *Proceedings of the 12th acm international conference on pervasive technologies related to assistive environments*, pages 289–298, 2019.
- [146] A. Laussedat. *Recherches sur les instruments : les méthodes et le dessin topographiques / par le colonel A. Laussedat*. Gauthier-Villars, 1898.
- [147] S. Leibson. Imus: mehr ruhezeiten für den host durch integriertes maschinelles lernen. Accessed on: Feb. 9. 2022, <https://www.digikey.de/de/articles/imus-let-your-host-sleep-with-on-board-machine-learning>, 4 2019.
- [148] WH Li, B Liu, Prabuono B Kosasih, and XZ Zhang. A 2-dof mr actuator joystick for virtual reality applications. *Sensors and Actuators A: Physical*, 137(2):308–320, 2007.
- [149] C. Lu. Alphapose. Accessed on: Feb. 8. 2022, <https://github.com/MVIG-SJTU/AlphaPose>.
- [150] Nadia Magnenat-Thalmann and Daniel Thalmann. *Handbook of virtual humans*. John Wiley & Sons, 2005, ISBN: 9780470023174.
- [151] Delle Rae Maxwell. *Graphical marionette: a modern-day Pinocchio*. PhD thesis, Massachusetts Institute of Technology, 1983.
- [152] Dushyant Mehta, Srinath Sridhar, Oleksandr Sotnychenko, Helge Rhodin, Mohammad Shafiei, Hans-Peter Seidel, Weipeng Xu, Dan Casas, and Christian Theobalt. Vnect: Real-time 3d human pose estimation with a single rgb camera. *ACM Trans. Graph.*, 36(4), jul 2017.
- [153] Manuel Meier, Paul Streli, Andreas Fender, and Christian Holz. Tapid: Rapid touch interaction in virtual reality using wearable sensing. In *2021 IEEE Virtual Reality and 3D User Interfaces (VR)*, pages 519–528, 2021.
- [154] A. Mikhailovsky. Teslasuit motion capture system. Accessed on: Feb. 13. 2022, <https://teslasuit.io/blog/teslasuit-motion-capture-system/>, 3 2017.
- [155] Code Miko. These youtubers are too unhinged...). Accessed on: Feb. 07. 2022, <https://youtu.be/E00iQFhck54?t=229>, 3 2022.
- [156] H. Miku. Hatsune miku, magical mirai 2016 (hatsune miku concert). Accessed on: Feb. 14. 2022, <https://youtu.be/X11CTE7CmnI?t=729>, 4 2018.
- [157] J. MUYBRIDGE. The horse in motion. *Nature*, 25(652):605–605, 1882.
- [158] Netflix. Say hello to n-ko. Accessed on: Feb. 14. 2022, <https://youtu.be/rE0-583WBcw?t=70>, 4 2021.
- [159] P. Nickeas. Teslasuit - improving human performance. Accessed on: Feb. 12. 2022, <https://teslasuit.io/special/TESLASUIT-presentation-Q12021.pdf>, 2021.

- [160] Ultimate History of CGI. Brilliance - cgi making of (1984). Accessed on: Feb. 9. 2022, [https://www.youtube.com/watch?v=HZY5\\_ZzRdbk](https://www.youtube.com/watch?v=HZY5_ZzRdbk), 6 2009.
- [161] C. Pao. Sensor fusion. Accessed on: Feb. 9. 2022, <https://www.ceva-dsp.com/ourblog/what-is-an-imu-sensor/>, 11 15, 2018.
- [162] Vittorio Passaro, Antonello Cuccovillo, Lorenzo Vaiani, Martino De Carlo, and Carlo Edoardo Campanella. Gyroscope technology and applications: A review in the industrial perspective. *Sensors*, 17(10):2284, 2017.
- [163] J. Purtill. 'haptic feedback' virtual reality teslasuit can simulate everything from a bullet to a hug. Accessed on: Feb. 12. 2022, <https://www.abc.net.au/news/science/2021-04-01/vr-teslasuit-simulates-virtual-reality-touch-haptic-feedback/100030320>, 4 2021.
- [164] Facebook Research. Densepose in detectron2. Accessed on: Feb. 8. 2022, <https://github.com/facebookresearch/detectron2>.
- [165] Iasonas Kokkinos Riza Alp Güler, Natalia Neverova. Densepose: Dense human pose estimation in the wild. 2018.
- [166] R. Rojas. The kalman filter. Accessed on: Feb. 10. 2022, <http://www.inf.fu-berlin.de/lehre/SS04/robotik/Vorlesung/Vorlesung5/kalman.pdf>.
- [167] Wilhelm Conrad Röntgen. Ueber die durch electricität bewirkten form-und volumenänderungen von dielectrischen körpern. *Annalen der Physik*, 247(13):771–786, 1880.
- [168] S. Roy. Potentiometer | Variable Resistor | It's Important applications. Accessed on: Nov. 25. 2021. URL: <https://lambdageeks.com/potentiometer/>.
- [169] Carsten Stoll, Nils Hasler, Juergen Gall, Hans-Peter Seidel, and Christian Theobalt. Fast articulated motion tracking using a sums of gaussians body model. In *2011 International Conference on Computer Vision*, pages 951–958, 2011.
- [170] T. Tillotson. Inside unreal: 2020 year in review. Accessed on: Nov. 28. 2021, <https://www.twitch.tv/videos/840713360?t=1h6m20s>.
- [171] SA Veress, FG Lippert, MCY Hou, and T Takamoto. Patellar tracking patterns measurement by analytical x-ray photogrammetry. *Journal of biomechanics*, 12(9):639–650, 1979.
- [172] Fei Wang, Sanping Zhou, Stanislav Panev, Jinsong Han, and Dong Huang. Person-in-wifi: Fine-grained person perception using wifi. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 5452–5461, 2019.
- [173] P. Wiebel, W. Vasulka, D. Dunn, and S. Vasulka. *Eigenwelt Der Apparatewelt: Pioneers of Electronic Art*. Ars electronica. The Vasulkas, 1992.

- [174] A. Yamato. First vlog. trying this out... Accessed on: Feb. 14. 2022, <https://www.youtube.com/watch?v=5pciLZ8ffPY>, 6 2011.
- [175] A. Yamato. Makeup tips tutorial - natural digital look. Accessed on: Feb. 09. 2022, <https://youtu.be/yxD2xGz-QPk?t=49>, 11 2011.
- [176] M. Young. How the polar express created a new type of animation. Accessed on: Feb. 9. 2022, <https://www.thethings.com/polar-express-animation-tom-hanks-motion-capture/>, 2 9, 2022.



## Eidesstattliche Erklärung

Ich erkläre hiermit an Eides statt, dass ich die vorliegende Master-Thesis selbständig und ohne unzulässige fremde Hilfe angefertigt habe. Alle verwendeten Quellen und Hilfsmittel, sind angegeben.

Die Arbeit wurde bisher in gleicher oder ähnlicher Form in keinem anderen Studiengang als Prüfungsleistung vorgelegt oder an anderer Stelle veröffentlicht.

Ich bin mir bewusst, dass eine falsche Erklärung rechtliche Folgen haben kann.

---

Mössingen, 31.03.2021. Metzger, Ron Marvin





## Anhang

Auf dem angehängten USB-Stick sind folgende Dateien zu finden:

- Die ESP32 Sketche
- Das UE Projekt
- Eine Anleitung für das Nachbauen des VRonGlove
- 3D-Modelle des Handschuhs
- 3D-Bilder der Designiterationen
- Bilder und Videos von Prototypen
- Das Schriftstück als PDF
- Quellennachweise als BibTeX Datei

