

# **Umsetzung einer inversen Kinematik in Unity VR-Anwendungen am Beispiel der Diagnostik- und Reha-Applikationen im Forschungsprojekt VReha**

## **Bachelorarbeit**

Internationaler Studiengang Medieninformatik

Fachbereich 4

vorgelegt von  
Larissa Wagnerberger  
am 26.03.2021

**Erstgutachter:** Prof. Dr. Tobias Lenz

**Zweitgutachter:** Dr. Detlef Runde

Berlin 2021

# Kurzfassung

Oft sieht man in Anwendungen von Virtueller Realität (VR) in der Ich-Perspektive nur seine Hände. Es wäre allerdings realistischer, eine vollständige Darstellung des eigenen Körpers zu sehen. Dies kann durch einen Ganzkörperavatar erfolgen, was viele Vorteile bietet: Die Benutzenden fühlen sich präsenter [1], können Distanzen besser einschätzen [2] und sogar die kognitiven Fähigkeiten der Benutzenden können verbessert werden [3].

Aus diesen Gründen wurde in dieser Arbeit eine Lösung für eine Selbstrepräsentation durch einen Avatar mithilfe von inverser Kinematik umgesetzt. Die Lösung wurde am Beispiel des Forschungsprojektes VReha implementiert.

Zuerst wurden existierende Lösungen evaluiert. Die beste Lösung wurde anschließend in das Forschungsprojekt VReha integriert und an die Projektanforderungen angepasst: Der Avatar kann passend zur benutzenden Person skaliert werden, sitzen, stehen und laufen. Außerdem verarbeitet die Anwendung Daten von verschiedenen Trackingsensoren, passt sich an die Datenverfügbarkeit an und unterstützt unterschiedliche Avatare.

Eine abschließende Nutzerstudie kam zu dem Ergebnis, dass Nutzer\*innen eine Repräsentation durch einen Avatar einer Repräsentation durch Hand-Modelle vorziehen und sich mit einem Avatar in der virtuellen Welt verkörperter fühlen.

# Abstract

In virtual reality applications you often only see your hands from the first-person perspective. However, it would be more realistic to see a complete representation of your own body. This can be done through a full-body avatar, which offers many advantages: Users feel more present [1], can better estimate distances [2] and even cognitive skills can be improved [3].

For these reasons, a solution for self-representation by an avatar using inverse kinematics is realized on the example of the VReha research project in this thesis.

First, existing solutions were evaluated. The best solution was integrated into the research project VReha and adapted to the project requirements: The avatar can be scaled to fit the user, sitting, standing and walking. In addition, the solution processes data from different sensors, adapts to the data availability and supports different avatars.

A final user study concluded that users prefer representation through an avatar to representation by hand models, and feel more embodied in the virtual world with an avatar.

# Inhaltsverzeichnis

<b>1 Einleitung</b>	<b>1</b>
<b>2 Grundlagen</b>	<b>2</b>
2.1 Virtuelle Realität (VR) . . . . .	2
2.2 Forschungsprojekt VReha . . . . .	4
2.3 Inverse Kinematik (IK) . . . . .	5
2.4 Unity . . . . .	6
2.5 Stand der Forschung und Technik . . . . .	7
2.5.1 Auswirkungen der Selbstrepräsentation durch einen Avatar . . . . .	7
2.5.2 Existierende IK-Lösungen . . . . .	9
<b>3 Zielsetzung und Anforderungen</b>	<b>12</b>
3.1 Zielsetzung . . . . .	12
3.2 Anforderungen . . . . .	12
<b>4 Evaluation existierender IK-Lösungen</b>	<b>14</b>
4.1 Implementierung der Datensammlung . . . . .	15
4.2 Implementierung der IK-Lösungen . . . . .	16
4.3 Auswertung . . . . .	18
<b>5 Integration in VReha</b>	<b>22</b>
5.1 Grundlegende Softwarestruktur . . . . .	22
5.2 Integration von VRIK . . . . .	23
5.3 Skalierung des Avatars . . . . .	24
5.4 Konfiguration des Avatars . . . . .	27
5.5 Positionierung und Ausrichtung der Effektoren für die Sitzpose . . . . .	28
5.6 Positionierung und Ausrichtung der Kopf- und Arm-Effektoren . . . . .	29
5.6.1 Ermittlung der Rotationsoffsets . . . . .	29
5.6.2 Positionierung und Ausrichtung des Kopf-Effektors . . . . .	31
5.6.3 Positionierung und Ausrichtung der Arm-Effektoren . . . . .	31
5.6.4 Platzierung der Hände am Lenker . . . . .	33
5.7 Positionierung und Ausrichtung der Hüft- und Fuß-Effektoren . . . . .	34
5.8 Kalibrierung der Kinect . . . . .	38
5.9 Optimierung der Avatare . . . . .	40
<b>6 Nutzerstudie</b>	<b>41</b>
<b>7 Zusammenfassung und Ausblick</b>	<b>45</b>

<b>Literaturverzeichnis</b>	<b>47</b>
<b>Bildquellenverzeichnis</b>	<b>53</b>
<b>Abkürzungsverzeichnis</b>	<b>54</b>
<b>Abbildungsverzeichnis</b>	<b>55</b>
<b>Quellcodeverzeichnis</b>	<b>57</b>
<b>Eigenständigkeitserklärung</b>	<b>58</b>
<b>Anhang</b>	<b>59</b>
A    Evaluation existierender IK-Lösungen . . . . .	59
B    Latenz der Azure Kinect . . . . .	63
C    Nutzerstudie . . . . .	64
C.1    Fragen . . . . .	64
C.2    Auswertung . . . . .	66
C.3    Antworten . . . . .	71

# 1 Einleitung

In den letzten Jahren verbreitete sich Virtuelle Realität (VR) zunehmend. Mittlerweile wird sie in vielen Bereichen eingesetzt – von Spielen bis hin zu medizinischen Zwecken.

Dabei sehen Benutzende in VR in der Ich-Perspektive oft nur eine Darstellung ihrer Hände. Eine Repräsentation durch einen Ganzkörperavatar ist noch selten, bietet allerdings viele Vorteile: Personen fühlen sich präsenter in der virtuellen Welt [1] und verhalten sich realistischer und natürlicher. Sie neigen seltener dazu, in dieser Umgebung durch Wände zu gehen [4], können Distanzen besser abschätzen [2] und sogar ihre kognitiven Fähigkeiten werden gefördert [3].

In dieser Arbeit wird daher eine Lösung für die Darstellung der Nutzer\*innen in der virtuellen Realität durch einen Ganzkörperavatar umgesetzt. Am Beispiel des Forschungsprojektes VReha des Fraunhofer Heinrich-Hertz-Instituts wird dies realisiert. Dieses Projekt beschäftigt sich mit dem Potenzial von VR für die Diagnose und Therapie von Erkrankungen mit Beeinträchtigung der kognitiven Funktionen.

Anhand der Kopf- und Händepositionen und gegebenenfalls der Fußpositionen der benutzenden Person werden die Gelenke des Avatars positioniert. Dies wird mittels inverser Kinematik (IK) umgesetzt. Sie ermöglicht eine realistische Darstellung des Skeletts des/der Benutzenden aus wenigen bekannten Informationen über die Position der Gelenke. Die Gliedmaßen werden dabei als kinematische Kette betrachtet. Ausgehend von der Position und Orientierung des letzten Glieds der Kette (Endeffektor), beispielsweise der Hand, werden alle Glieder der Kette in eine schlüssige Lage gebracht.

Der Avatar wird außerdem passend zur benutzenden Person skaliert und kann sitzen, stehen und laufen. Es werden zudem die Daten von verschiedenen Trackingsensoren je nach Datenverfügbarkeit verwendet.

Im ersten Teil dieser Arbeit werden grundlegende Begriffe, die verwendete Hardware, der aktuelle Forschungsstand zur Selbstrepräsentation durch einen Avatar in VR sowie existierende IK-Lösungen erklärt. Weiterhin werden existierende IK-Lösungen evaluiert, um die am besten geeignete IK-Lösung für die Anwendung in VReha als auch für eine stimmige Positionierung der Gelenke zu finden.

Darauf aufbauend wird die beste IK-Lösung in das Forschungsprojekt VReha integriert und erweitert. Es wird zunächst eine Skalierung des Avatars anhand der Körpergröße und den Körperproportionen der benutzenden Person durchgeführt. Kopf-, Hand- und Ellenbogen-Effektoren werden je nach Datenlage der verschiedenen Sensoren positioniert und ausgerichtet. Die Platzierung der Hüft- und Fuß-Effektoren wird mit den Daten der Kinect von Microsoft durchgeführt. Zusätzlich werden nötige Anpassungen für die Verwendung in dem Forschungsprojekt VReha umgesetzt, beispielsweise das Einnehmen einer Sitzpose.

Im Abschluss wird eine Nutzerstudie durchgeführt und die Ergebnisse dieser Arbeit zusammengefasst.

## 2 Grundlagen

In diesem Kapitel werden die für diese Arbeit relevanten Begriffe, die verwendete Hard- und Software sowie der aktuelle Forschungsstand erläutert.

### 2.1 Virtuelle Realität (VR)

Die Virtuelle Realität bezeichnet eine vom Computer durch spezielle Software erzeugte Realität, in die eine Person mit entsprechender Hardware eintauchen kann. Dazu wird üblicherweise ein VR-Headset benutzt, es können aber auch speziell designete Räume mit Bildschirmen verwendet werden. Die Nutzer\*innen können sich in einer virtuellen 360°-Umgebung durch ihre Kopfbewegung umschauen und gegebenenfalls auch bewegen. Eine 3D-Tiefenwirkung und räumliches Audio können das Erlebnis realistischer machen.

VR wird vielseitig eingesetzt. Zum einen wird es für verschiedenste Spiele und interaktive Filme verwendet, aber auch in anderen Bereichen, wie zum Beispiel in der Architektur, für die Produktentwicklung oder in der Medizin. So kann VR beispielsweise für die Therapie von Phobien, Süchten oder Schmerzen eingesetzt werden. Auch um Kosten zu sparen und Risiken zu verringern, wird VR eingesetzt, zum Beispiel in der Ausbildung von Pilot\*innen oder bei Medizinstudent\*innen, die Operationen an virtuellen Patient\*innen erlernen können. [5], [6]

#### VR Hardware

Die wichtigste Hardware für VR ist ein Head-Mounted-Display (HMD). Durch Stereodisplays wird ein Tiefeneindruck geschaffen. Pro Auge gibt es ein Display, welches die gleiche Szene aus der Perspektive des jeweiligen Auges zeigt. Die angezeigten Bilder werden im Gehirn zu einem räumlichen Bild zusammengesetzt. [7]

Ein Trackingsystem ermöglicht die Verfolgung der Kopf- beziehungsweise Headset-Lage, sodass die Bilder auf den Displays entsprechend der Perspektive des/der Benutzenden angezeigt werden. Zusätzlich zum HMD kann noch weitere Hardware verwendet werden, um zum Beispiel die Bewegungen der benutzenden Person zu verfolgen.

Im Folgenden wird die Hardware vorgestellt, die für diese Arbeit verwendet wurde. In Abbildung 2.1 ist diese dargestellt.

- **Oculus Quest**

Die Oculus Quest ist ein autarkes VR-Headset, das heißt, die Applikation läuft direkt auf dem Headset. Im Gegensatz zur Oculus Rift ist kein PC erforderlich, um die VR-Anwendung auszuführen. Auch werden keine externen Sensoren für das Tracking der Headsetposition benötigt. Dies bietet den Vorteil, dass sich der/die Nutzer\*in frei und

ungestört von Kabeln bewegen kann. Das Headset kann aber auch per Kabel an einen PC angeschlossen werden und wie ein PC-Headset verwendet werden. Vier eingebaute Weitwinkelkameras ermöglichen das räumliche Orten und das Handtracking. Die VR-Anwendungen können mit zwei Oculus Touch Controllern oder mit den Händen bedient werden. [8]

#### ▪ Oculus Rift

Die Oculus Rift ist ein VR-Headset für den PC. Zur Lagebestimmung werden standardmäßig zwei separate Sensoren verwendet, welche die Position des Headsets mithilfe der vom Headset ausgehenden Infrarot-Strahlung bestimmen. Die Sensoren werden so im Raum platziert, dass sie das ausgestrahlte Infrarotlicht vom Headset gut empfangen können. [9] In diesem Projekt wurde die Oculus Rift in Verbindung mit der Leap Motion für das Handtracking verwendet.

#### ▪ Leap Motion

Die Leap Motion ist eine Trackingeinheit, die auf VR-Headsets montiert werden kann. In diesem Projekt wurde sie auf der Oculus Rift montiert. Sie kann Hände und Finger sowie Ellenbogen erfassen. Dazu werden drei Infrarot-LEDs und zwei Infrarotkameras verwendet. Die gewonnenen Daten werden per Kabel an den angeschlossenen PC übertragen und dort verarbeitet. [10]

#### ▪ Kinect 2.0

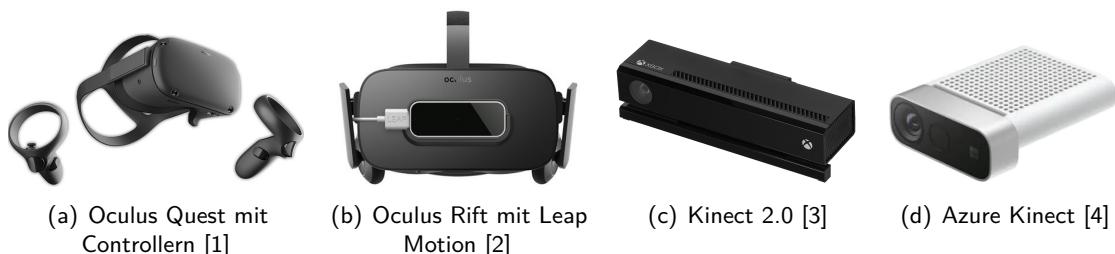
Die Kinect 2.0 ist eine von Microsoft entwickelte Hardware, gedacht für die Steuerung der Xbox (Spielekonsole) durch Körperbewegungen. Durch eine Weitwinkelkamera und einen Infrarotsensor werden Daten erfasst. Es können bis zu sechs Skelette erfasst werden und deren Herzfrequenz, Mimik und Gelenkpositionen bestimmt werden. [11]

Sie wird aktuell bei den Projektpartnern von VReha eingesetzt.

#### ▪ Azure Kinect DK

Die Azure Kinect DK ist ein Entwicklerkit mit Funktionen für maschinelles Sehen und Sprachverarbeitung. Mithilfe einer Kamera und einem Tiefensensor werden Daten gesammelt. Wie bei der Kinect 2.0 können Skelette erkannt und Gelenkpositionen bestimmt werden. [12]

Sie ist eine mögliche Alternative für die Kinect 2.0, da diese im Handel nicht mehr erhältlich ist.



**Abbildung 2.1:** Verwendete Hardware

## 2.2 Forschungsprojekt VReha

VReha steht für „Virtuelle Welten für digitale Diagnostik und kognitive Rehabilitation“. Das Forschungsprojekt erforscht das Potenzial der Diagnose und Therapie in VR. Der Fokus liegt auf der Diagnose und Therapie von Erkrankungen, welche die kognitiven Funktionen beeinträchtigen, wie zum Beispiel Demenz oder Schädel-Hirn-Traumata. Realitätsnahe Tests in VR sollen die Diagnose präziser und sensitiver machen. Anhand der Diagnose werden VR-basierte Therapieprogramme für die Patient\*innen ausgewählt. An dem Projekt arbeiten zahlreiche Institutionen wie die Charité Berlin und das Max-Planck-Institut für Kognitions- und Neurowissenschaften in Leipzig mit. [13]

Weiterführend kann ein von VReha veröffentlichtes Video [14] angesehen werden.

Die Aufgabe des Fraunhofer HHI ist es, die benötigte VR-Technologie für die Interaktion und Visualisierung in VR zu entwickeln. Dabei liegt der Fokus auf der Entwicklung von neuen berührungslosen Technologien, wie zum Beispiel Greifen und der Visualisierung des/der Nutzers\*in in VR.

### Aufgaben in VReha

Es gibt zwei verschiedene Aufgaben in VReha: die immersive Virtual Memory Task (imVMT) und die immersive Virtual Supermarket Task (imVST), wie in Abbildung 2.2 dargestellt. Bei der imVMT wird das räumliche Gedächtnis getestet und trainiert. Der/die Patient\*in muss sich dabei die Anordnung von Gegenständen merken. Bei der imVST können exekutive Funktionen<sup>1</sup> getestet und trainiert werden, indem der/die Patient\*in in einem virtuellen Supermarkt einkaufen geht und dabei alle Produkte finden soll, die auf einer Einkaufsliste aufgeführt sind. Dabei gibt es eine stehende Version, in der der/die Nutzer\*in einen Einkaufswagen schiebt und eine sitzende Version, in der der/die Nutzer\*in ein Elektromobil steuert. Der/die Versuchsleiter\*in kann auf seinem/ihrem Bildschirm die Aktion des/der Patients\*in verfolgen.



(a) Immersive Virtual Memory Task (imVMT)      (b) Immersive Virtual Supermarket Task (imVST)

**Abbildung 2.2:** Beispieldarstellung der Aufgaben in VReha

<sup>1</sup>Der Begriff **exekutive Funktionen** fasst verschiedene kognitive Fähigkeiten zusammen, welche für die Kontrolle und Selbstregulierung des Verhaltens nötig sind. Dies sind beispielsweise Planen, Entscheidungen treffen und kurzfristig Informationen aufnehmen. [15]

### Technischer Aufbau von VReha

In Abbildung 2.3 wird der technische Aufbau von VReha veranschaulicht. Es gibt eine mobile und eine feste Installation, da VReha sowohl im klinischen als auch im häuslichen Bereich eingesetzt werden soll. Die mobile Version soll mit der Oculus Quest und gegebenenfalls der Kinect arbeiten. Die feste Version verwendet die Oculus Rift, die Leap Motion und die Kinect. Für die Steuerung in der imVST-Anwendung wird zusätzlich ein Controller verwendet, der an einer Lenkerstange befestigt ist.

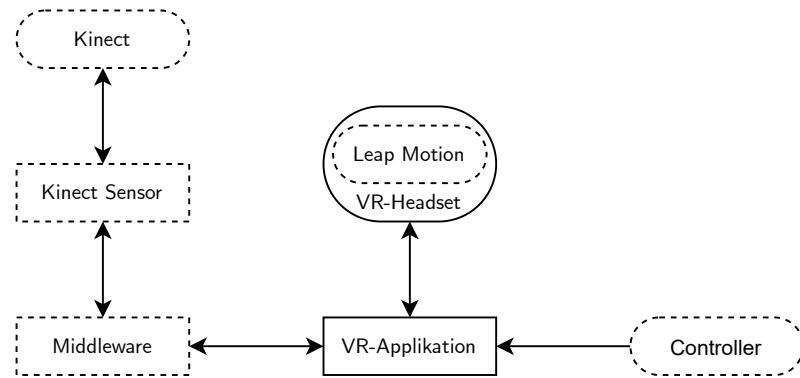


Abbildung 2.3: Technische Aufbau von VReha

### 2.3 Inverse Kinematik (IK)

Bei der inversen Kinematik werden Gliedmaßen als kinematische Kette betrachtet, beispielsweise kann der Arm als kinematische Kette mit den Gelenken Schulter, Ellenbogen und Handgelenk dargestellt werden. Anhand der Position und Orientierung des letzten Glieds der Kette (Endeffektor), beispielsweise der Hand, werden alle Glieder der Kette in eine passende Lage gebracht. Inverse Kinematik wird vor allem in der Robotik und Computeranimationen eingesetzt.

## 2.4 Unity

Das Projekt VReha wird mit der Spiele-Engine Unity (Version 2019.4.2) entwickelt. Unity ist eine verbreitete Engine für die Entwicklung von Spielen und Anwendungen in 2D und 3D. Als Programmiersprache wird C# verwendet.

Eine Unity Szene ist mit sogenannten *GameObjects* aufgebaut. Diese können im Editor platziert und bearbeitet werden. Jedes *GameObject* besteht aus Komponenten, die im Inspektor bearbeitet werden können. Die Transform-Komponente existiert bei jedem *GameObject*. Sie definiert die Positions-, Rotations- und Skalierungswerte.

Skripte können direkt einem *GameObject* zugewiesen werden. Die Skripte werden üblicherweise von der Klasse *MonoBehaviour* abgeleitet. Diese beinhaltet wichtige Methoden, wie zum Beispiel die Start- und Update-Methode. Die Start-Methode wird beim Start der Applikation beziehungsweise beim Aktivieren des Skriptes aufgerufen. Die Update-Methode wird bei der Aktualisierung jedes Frames ausgeführt.

### Avatare in Unity

Unitys Animationssystem bietet spezielle Funktionen für die Arbeit mit humanoiden Avataren. Es können Modelle importiert werden und Knochenstrukturen automatisch erkannt werden. Avatare sind durch eine Hierarchie von *GameObjects* aufgebaut. Jeder Knochen wird durch eine Transformationskomponente definiert. Die Anordnung in der Hierarchie ermöglicht das Mitbewegen der untergeordneten Knochen. Wird zum Beispiel der linke Oberarm bewegt, bewegen sich der linke Ellenbogen und die Hand entsprechend mit. In Abbildung 2.4 sieht man die Knochenstruktur eines in dieser Arbeit verwendeten Avatars sowie rechts die Avatar-Struktur in Unity.



**Abbildung 2.4:** Aufbau eines humanoiden Avatars in Unity

## 2.5 Stand der Forschung und Technik

Im Folgenden wird der Stand der Forschung in Bezug auf das Verkörperungsgefühl und die Beeinflussung des/der Nutzers\*in durch seinen/ihren Avatar in VR vorgestellt. Anschließend wird der aktuelle Stand der Technik von IK-Lösungen für Avatare vorgestellt.

### 2.5.1 Auswirkungen der Selbstrepräsentation durch einen Avatar

Personen, die in immersive virtuelle Welten eintauchen, können die Illusion erleben, sich an diesem Ort wirklich zu befinden. Dies wird als „Präsenz“ bezeichnet. Diese Illusion besteht aus zwei Aspekten: der Illusion an diesem Ort zu sein („Place Illusion“) und die Illusion, dass das Szenario wirklich passiert („Plausibility Illusion“). [1]

Die Selbst-Repräsentation durch einen Avatar beeinflusst beide Faktoren (Place und Plausibility Illusion). Den eigenen Körper betrachten zu können, ruft das Gefühl hervor, wirklich an dem virtuellen Ort zu sein (Place Illusion). Bewegt sich der Avatar synchron zum/zur Nutzenden, wird die Plausibility Illusion verstärkt, es kann ein Gefühl der Verkörperung entstehen. [1]

Dieses Gefühl ist eine Kombination von Empfindungen. Es entsteht in Verbindung mit der Wahrnehmung, ein (virtueller) Körper zu sein, ihn zu besitzen und zu kontrollieren. Wichtig ist in diesem Zusammenhang vor allem die Empfindung des Körperbesitzes, der Handlungsfähigkeit und der eigenen Position im Raum. [16]

Das Gefühl der Verkörperung wurde in zahlreichen Studien untersucht. Ein bekanntes Beispiel ist *die Puppenhand-Illusion*. Bei diesem Experiment wird eine künstliche Hand neben die Hand der Probanden gelegt und die echte Hand verdeckt. Die künstliche und die echte Hand werden synchron haptisch stimuliert. Nach kurzer Zeit kommt es zu einer Sinnestäuschung bei den Probanden. Sie nehmen die künstliche Hand als Teil ihres Körpers wahr. [17]

Diese Puppenhand-Illusion wurde in VR erfolgreich von Slater u. a. reproduziert. [18]

In einem darauf aufbauenden Projekt wurde das Experiment mit synchronen Bewegungen beider Hände statt der haptischen Stimulierung durchgeführt. Es zeigte, dass die Illusion des Körperbesitzes auch durch synchrone Bewegung von virtueller und echter Hand geschaffen werden kann. [19]

Ein ähnliches Experiment wurde mit einem Ganzkörperavatar durchgeführt. Bewegte sich der Avatar synchron zum/zur Benutzer\*in, war das Gefühl des Körperbesitzes höher, als wenn er sich asynchron bewegte. [20]

Die genannten Studien zeigen, dass ein Gefühl des Körperbesitzes über virtuelle Gliedmaßen entwickelt werden kann. Darüber hinaus wurde gezeigt, dass sogar ein Empfinden des Körperbesitzes für Avatare aufgebaut werden kann, die sich im Aussehen von ihrem/ihrer Benutzer\*in unterscheiden [21]–[24].

Für ein Gefühl der Präsenz ist eine Repräsentation durch einen Avatar förderlich [25]. Nutzer\*innen ziehen die Repräsentation durch einen Avatar, der sich synchron zu ihren Bewegungen bewegt, keinem Avatar und einer Repräsentation durch einen statischen Avatar<sup>2</sup> vor. Außerdem kann die Selbstrepräsentation durch einen Avatar das Gefühl deutlich verstärken, in der virtuellen Welt präsent zu sein und die virtuelle Welt als echt zu empfinden. [26]

Des Weiteren zeigten Slater u. a., dass das Gefühl der Präsenz abhängig vom Gefühl der Assoziation mit dem Avatar ist [27], [28].

Weitere Studien belegen, dass das Verhalten der Nutzer\*innen durch die Selbstdarstellung mittels Avatar verändert werden kann. Baisen u. a. präsentieren die Theorie *Veränderung sozialer Interaktionen*. Diese definiert drei Dimensionen, in denen sich die reale Darstellung von der virtuellen unterscheiden kann: Selbstrepräsentation, Wahrnehmungsfähigkeiten und situativer Kontext. [29]

Für diese Arbeit ist die Verhaltensänderung, die durch Selbstrepräsentation mithilfe eines Avatars entstehen kann, hervorzuheben. Dies wurde in einigen Studien untersucht und nachgewiesen. Es wurde festgestellt, dass Personen, die durch einen größeren Avatar als ihr Gegenüber repräsentiert wurden, dazu neigten, selbstbewusster und bestimmter zu verhandeln als Personen mit einem kleineren Avatar. In einem zweiten Experiment wurde beobachtet, dass Personen, die durch einen Avatar mit einem attraktiven Gesicht dargestellt wurden, dazu tendierten, Fremden gegenüber offener zu sein. Aus diesen beiden Experimenten schlossen Yee u. a., dass sich Personen den Verhaltensweisen und Einstellungen, welche sie mit dem Aussehen ihres Avatars assoziieren, anpassen. [30]

Die Selbstrepräsentation durch einen Avatar kann sogar die Leistung bei einer Aufgabe beeinflussen: Teilnehmer\*innen, verkörpert durch einen lässig gekleideten, dunkelhäutigen Avatar, trommelten vielfältiger als Teilnehmer\*innen mit einem formell gekleideten, hellhäutigen Avatar [31]. Außerdem wurde demonstriert, dass die Darstellung des Avatars auch die kognitiven Fähigkeiten beeinflussen kann. Personen, welche durch Albert Einstein im höheren Alter verkörpert wurden, erbrachten bessere kognitive Leistungen unter Berücksichtigung der vorherigen kognitiven Fähigkeiten als Personen mit einem neutralen Avatar. Dabei war die kognitive Verbesserung bei Personen mit geringerem Selbstwertgefühl am größten. [32]

Des Weiteren wurde bei dieser Studie festgestellt, dass Vorurteile gegenüber älteren Menschen bei Personen abgebaut wurden, welche durch Einstein verkörpert wurden. [32]. Dieses Phänomen wurde auch in einer weiteren Studie beobachtet, in der Personen entweder durch einen alten oder jungen Avatar repräsentiert wurden [33].

Zwei weitere Studien zeigen, dass auch Voreingenommenheiten Hellhäutiger gegenüber Dunkelhäutigen durch die Verkörperung in einem dunkelhäutigen Avatar beeinflusst werden können. In einem Experiment wurden hellhäutige Teilnehmer\*innen in einem dunkelhäutigen oder hellhäutigen Avatar verkörpert. Die Teilnehmenden, die mit einem dunkelhäutigen Avatar verkörpert wurden, hatten kurzzeitig weniger rassistische Voreingenommenheiten gegenüber Dunkelhäutigen.

---

<sup>2</sup>In dieser Studie bedeutet ein **statischer Avatar**, dass der Avatar in einer Sitzpose dargestellt ist und sich entsprechend der Blickrichtung des/der Nutzers\*in ausrichtet. Ansonsten bewegt er sich nicht. [26]

gen. [34]

Eine frühere Studie kam zu einem gegensätzlichen Ergebnis [35]. Peck u. a. vermuten allerdings, dass dies nicht an der Verkörperung durch einen Avatar lag, sondern an der Situation eines Bewerbungsgespräches, in die sich die Teilnehmenden hineinversetzen sollten. Sie mutmaßen, dass diese Situation bekannt für Diskriminierung ist und deshalb das Ergebnis beeinflusste. [34]

Neben dem Phänomen, dass das Aussehen des Avatars die Verhaltensweise der benutzenden Person beeinflussen kann, haben weitere Studien gezeigt, dass die Selbstdarstellung in VR durch einen Avatar auch Einfluss auf Distanzschätzungen und kognitive Fähigkeiten haben kann.

In VR ohne die Selbstdarstellung durch einen Avatar neigen Personen dazu, Distanzen kürzer wahrzunehmen [36]. Die Distanzschätzung kann verbessert werden, wenn die Nutzer\*innen durch einen Avatar verkörpert werden [2], [37]–[39]. Zum Beispiel mussten Teilnehmende in einem Experiment in VR entscheiden, ob sie über eine Stange steigen können. Dabei war das Urteilsvermögen, ob dies machbar ist, bei Personen besser, welche durch einen Avatar verkörpert wurden [39].

Wie oben erwähnt, können die kognitiven Fähigkeiten durch die Selbstdarstellung mit einem Avatar verbessert werden. Es wurde beobachtet, dass Personen sich Buchstaben besser merken können, wenn diese einen Avatar haben, der ihre echten Handgesten abbildet. Personen ohne Avatar, die ihre Hände nicht sehen konnten, hörten nach kurzer Zeit auf, ihre Hände zu bewegen und konnten sich die Buchstaben schlechter merken. [3]

Darüber hinaus wurde belegt, dass sich Nutzer\*innen mit einem Avatar in VR natürlicher verhalten. Sie gehen seltener durch Wände als Personen ohne Avatar [4].

Insgesamt bietet die Darstellung durch einen Avatar viele Vorteile. Für diese Arbeit sind vor allem das Gefühl der Verkörperung, die bessere Distanzschätzung und das natürlichere Verhalten hervorzuheben. Zusätzlich ist die Beobachtung essenziell, dass sich die kognitiven Fähigkeiten durch die Selbstrepräsentation mit einem Avatar verbessern. Dies ist insbesondere von Bedeutung für VReha, da dort die kognitiven Fähigkeiten der Nutzer\*innen getestet werden. Außerdem verdeutlichen die vorgestellten Studien die Wichtigkeit eines neutralen Aussehens sowie einer korrekten Skalierung des Avatars, um das natürliche Verhalten der Nutzenden nicht zu beeinflussen. Dies wird bei der Umsetzung in dieser Arbeit berücksichtigt.

### 2.5.2 Existierende IK-Lösungen

Es gibt verschiedenste Ansätze, die Darstellung durch einen Avatar umzusetzen. Zum einen können die Bewegungen und Positionen der Körpergelenke durch Motion Capture-Systeme verfolgt werden. Dabei gibt es einerseits Marker- oder Tracker-basierte Systeme, bei denen die Person zum Beispiel einen Anzug mit reflektierenden oder aktiven Markern tragen muss, welche dann von Infrarotkameras erfasst werden. Ein anderer Ansatz sind Systeme ohne Marker/Tracker, die durch Computer Vision-Algorithmen die Position und Rotation der Körpergelenke ermitteln. Ein Beispiel für solche Systeme ist die Kinect von Microsoft.

In dieser Arbeit soll allerdings für die Grundlösung kein Motion Capture-System verwendet werden – ein VR-Headset und gegebenenfalls eine Leap Motion sollen ausreichen. Dies ist von VReha gefordert, da die mobile Variante auch nur mit dieser Hardware nutzbar sein soll. Außerdem sollen die Versuche einfach und schnell durchführbar sein. Die Verwendung von Markern oder zusätzlichen Trackern würde zu einer Beeinträchtigung des/der Patients\*in führen und die Vorbereitungszeit der Versuche erhöhen.

Es wird eine IK-Lösung benötigt, um nur aus den Trackingdaten der Hände und des Kopfes einen Avatar passend zum/zur Benutzenden darzustellen.

Eine IK-Lösung kann sogar Vorteile gegenüber einer Motion Capture-Lösung bieten: Zum Beispiel verglichen Roth u. a. eine Motion Capture-Lösung mit einer IK-Lösung in Verbindung mit einem reduzierten Markerset. Dabei konnte kein signifikanter Unterschied im Gefühl, den Avatarkörper zu besitzen, festgestellt werden. Der IK-Ansatz reduzierte allerdings in ihrem Versuch die Latenz und die Belastung der Nutzer\*innen vor allem dadurch, dass kein Tracking-Anzug für den IK-Ansatz erforderlich war. [40]

Auch Parger u. a. stellten fest, dass ihre IK-Lösung im Vergleich zu einer Motion Capture-Lösung weniger Verzögerung und keine Trackingfehler aufweist und sogar ein größeres Verkörperungsgefühl hervorruft. [41].

Im Folgenden werden existierenden Lösungen vorgestellt. Alle Lösungen benötigen Trackingdaten der Hände und des Kopfes. Weitere Trackingpunkte werden nur von einigen benötigt oder sind optional. Die hier vorgestellten Lösungen werden im Kapitel 4 evaluiert und verglichen.

- Das Open Source Unity-Asset VRArmIK ist eine IK-Lösung, um Arme in VR darzustellen. Dazu wird eine kinematische Kette von den Händen zum Kopf verwendet. Die Schultern werden erst rotiert, wenn ein bestimmter Grenzwert erreicht ist. Dies kommt hauptsächlich vor, wenn die benutzende Person die Arme komplett ausstreckt. Die möglichen Positionen des Ellbogens werden anhand des errechneten Arm-Innenwinkels bestimmt. Anschließend wird die Orientierung der Ellbogen mittels heuristischer Verfahren ermittelt. Dabei wird vor allem darauf geachtet, eine unnatürliche Beugung zu verhindern. [41], [42]
- Das Unternehmen Deep Motion entwickelte eine Ganzkörper-Posenschätzung mithilfe von physikalischen Berechnungen und einem Machine Learning-Modell anstatt inverser Kinematik. Es können ein bis sechs Tracker verwendet werden. Der Avatar besteht aus starren Körperteilen, welche mit Gelenken verbunden sind. So können auf den Avatar physikalische Kräfte wie Gravitation und Kollision wirken. Es entsteht eine Ragdoll (Puppe). Durch Simulieren der Muskeln und Ausbalancieren des Avatars gemäß dem Schwerpunkt entsteht so eine Posenschätzung für den Avatar. Mithilfe von Machine Learning wird die Simulation der Muskulatur verbessert und vielfältigere Bewegungen werden ermöglicht. [43], [44]

- Das von Root Motion entwickelte Asset Final IK für Unity beinhaltet die Komponente VRIK. Dieses bietet eine Avatar-Pose-Schätzung für ein bis sechs Trackingpunkte. Es können neben Kopf- und Hand-Trackingpunkten noch Hüft- und Fuß-Trackingpunkte genutzt werden. Dabei werden die Position und Rotation der Armgelenke, ähnlich wie bei VRArmIK, mittels Trigonometrie gelöst. Um den Unterkörper zu schätzen, wird der Masseschwerpunkt errechnet. Ist der Avatar nicht ausbalanciert, wird ein Schritt ausgelöst. Die Beinposen werden ebenfalls durch Trigonometrie gelöst. [45]
- In der Arbeit „Avatar-Pose-Schätzung aus Kopf- und Handpositionen für kollaborative VR-Umgebungen“ von Scheurenbrand wird eine IK-Lösung mit dem FABRIK-Algorithmus vorgestellt.  
Der FABRIK-Algorithmus ist ein iterativer Ansatz und steht für Forward and Backward Reaching Inverse Kinematics. Die Gelenkkette wird vorwärts und rückwärts iterativ durchlaufen. In der Forward Reaching Phase wird eine neue Position für jedes Gelenk bestimmt, sodass der Endeffektor erreicht wird und in der Backward Reaching Phase geschieht dies umgekehrt. [46]  
Die Arme und Beine werden durch den FABRIK-Algorithmus positioniert, der für diese Anwendungsfälle angepasst wurde. Zusätzlich wird eine Gangsynthese vorgestellt, welche den Avatar beim Laufen animiert. [47]
- Das Paper „Real-time Full-body Motion Reconstruction and Recognition for Off-the-Shelf VR Devices“ von Jiang u. a. beschreibt den Ansatz, den Oberkörper mittels inverser Kinematik und den Unterkörper durch Animation-Blending zu konstruieren. [48]
- SAFullBodyIK ist eine kostenlose Ganzkörper-IK-Lösung für Unity, entwickelt von StereoArts. Es können unter anderem die Hände, Ellenbogen, Füße, Knie und Hüfte als Effektoren benutzt werden. [49]
- Unity beinhaltet eine IK-Lösung durch den Animator. Es können Hände, Füße und der Augen-Mittelpunkt als Effektoren verwendet werden. Des Weiteren kann festgelegt werden, wie stark sich der Oberkörper mit dem Kopf mitbewegen soll. Zusätzlich können die Position und Rotation des Masseschwerpunkts vorgegeben werden. Diese Lösung ist allerdings nur voll funktionsfähig mit Hand-, Fuß-, Augen-Mittelpunkt- und Masseschwerpunkt-Daten. [50]
- Das Animation Rigging Packet von Unity mit dem Constraint<sup>3</sup> Two Bone IK kann die Position und Orientierung von den Gelenken der Arm- und Bein-IK-Kette berechnen. Es kann ein Endeffektor (Hand/Fuß) und ein „Hinweis“ (Ellenbogen/Knie) angegeben werden. [52]

<sup>3</sup>Ein **Constraint** ist in Unity eine Komponente, die Transform-Eigenschaften (Position, Rotation, Skalierung) eines *GameObjects* mit einem anderen verknüpft. [51]

# 3 Zielsetzung und Anforderungen

In diesem Kapitel wird die Aufgabe dieser Arbeit definiert und deren Anforderungen festgelegt.

## 3.1 Zielsetzung

Das Ziel dieser Arbeit ist es, eine Avatar-Posenschätzungs Lösung mittels inverser Kinematik am Beispiel des Forschungsprojektes VReha umzusetzen. Dazu sollen im ersten Schritt existierende Lösungen verglichen und evaluiert werden, um die am besten geeignete Lösung für die Anwendung in VReha und für eine stimmige Positionierung aller Gelenke zu finden.

Im zweiten Schritt soll diese Lösung in das Forschungsprojekt VReha integriert werden und an dessen Anforderungen angepasst werden.

## 3.2 Anforderungen

Ausgehend von den Projektanforderungen von VReha wurden die Anforderungen definiert. Sie wurden in funktionale und nicht-funktionale Anforderungen unterteilt:

### Funktionale Anforderungen

- Es soll folgende Hardware unterstützt werden (vgl. Abschnitt 2.1):
  - Oculus Quest, Oculus Rift, Leap Motion und Azure Kinect
- Trackingpunkte für Kopf und Hände sollen ausreichen, um alle Gelenke des Avatars positionieren zu können.
- Die Lösung soll durch weitere Trackingpunkte, wie Ellenbogen und Füße, erweiterbar sein.
- Die Lösung soll sich an die Datenverfügbarkeit der Sensoren anpassen.
- Der Avatar soll gemäß der Körpergröße und den Körperproportionen skaliert werden.
  - Der Prozess soll über eine grafische Oberfläche gesteuert werden.
  - Die Skalierungsdaten sollen wiederverwendbar sein.
- Der Avatar soll eine Sitzpose einnehmen können.
- Über eine Konfigurationsdatei sollen Parameter der Lösung einfach anpassbar sein.

- Es sollen verschiedene Avatare unterstützt werden.
  - Es soll ein weiblicher oder männlicher Avatar zur Verfügung stehen.
  - Avatar-Modelle mit unterschiedlichen Gelenkorientierungen sollen unterstützt werden.
  - Die Avatar-Modelle sollen einfach austauschbar sein.

### **Nicht-Funktionale Anforderungen**

- Die Bewegungen des Avatars sollen ohne große Verzögerung den Bewegungen der benutzenden Person folgen.
- Die Lösung soll die Performance der VReha-Applikation nicht beeinflussen.

## 4 Evaluation existierender IK-Lösungen

In diesem Kapitel werden die existierenden IK-Lösungen (vgl. Abschnitt 2.5.2) verglichen und evaluiert, um die am besten geeignete IK-Lösung zu finden. Dazu wird als Erstes die von Parger u. a. bereits durchgeführte Evaluation von IK-Lösungen erläutert. Anschließend wird eine eigene Evaluation der IK-Lösungen durchgeführt und deren Implementierung erklärt. Im Anschluss werden die dabei gewonnenen Daten analysiert und evaluiert.

Parger u. a. verglichen in ihrem Paper bereits die folgenden IK-Lösungen:

- Limb IK<sup>4</sup>
- SAFullBodyIK
- den FABRIK-Algorithmus
- die Lösung von Jiang u. a.
- ihre eigene Lösung VRArmIK

Da die Lösung von Jiang u. a. nicht öffentlich verfügbar ist, implementierten Parger u. a. diese selbst nach den gegebenen Informationen. Allerdings war die Lösung nicht in der Lage, akzeptable Nacken- und Schulterpositionen zu errechnen. Dabei war nicht sicher, ob dies an der Implementierung oder an dem Ansatz selbst liegt. Um dieses Problem zu umgehen, wurden die Motion Capture-Schulterpositionen verwendet. Die Tools LimbIK und SAFullBodyIK berechnen nur die IK-Kette von der Schulter zur Hand, sodass auch hier die Motion Capture-Schulterposition benutzt wurde, um die Lösungen vergleichbar zu machen.

Für die Evaluation wurden vier Motion Capture-Datensätze verwendet und der Root Mean Square Error<sup>5</sup> zwischen den Motion Capture-Daten und den berechneten IK-Gelenkpositionen bestimmt.

Bei geringem Schulter-Positionierungsfehler schnitt VRArmIK besser oder ähnlich im Vergleich zu den anderen Lösungen bei der Ellenbogen-Positionierung ab. Bei vielfältigen Bewegungen war der Schulter-Positionierungsfehler von VRArmIK größer als 10 cm, trotzdem war der Ellenbogen-Positionierungsfehler im Vergleich zu den anderen IK-Lösungen, die die richtige Schulterposition benutzen, nur im Schnitt 2 cm größer. Insgesamt schließen Parger u. a. daraus, dass VRArmIK sich besser als die anderen IK-Lösungen eignet, um die komplette Gelenkkette von Kopf bis zur Hand zu berechnen. [41], [42]

<sup>4</sup>Limb IK ist eine IK-Lösung, welche im Asset Final IK enthalten ist. Sie ist für die Lösung von Bein- und Armgelenksketten ausgelegt. [53]

<sup>5</sup>Der **Root Mean Square Error** wird häufig für die Messung des Vorhersagefehlers eines Modells verwendet. Er gibt an, wie viel eine Schätzung oder Funktion im Durchschnitt von den Originalwerten abweicht. Berechnet wird er aus der Quadratwurzel des mittleren quadratischen Fehlers.

$$\left( RMSE = \sqrt{\sum_{i=1}^n \frac{error^2}{n}} \right)$$

Im Folgenden werden die Tools VRArmIK und VRIK sowie die beiden Lösungen von Unity verglichen. Die Lösungen von Deep Motion und Scheurenbrand waren zum Zeitpunkt dieser Arbeit nicht verfügbar und konnten daher nicht evaluiert werden. Es wird in dieser Arbeit nur die Arm-Gelenkkette evaluiert, da VRIK als Einziges eine Ganzkörper-Lösung bereitstellt.

## 4.1 Implementierung der Datensammlung

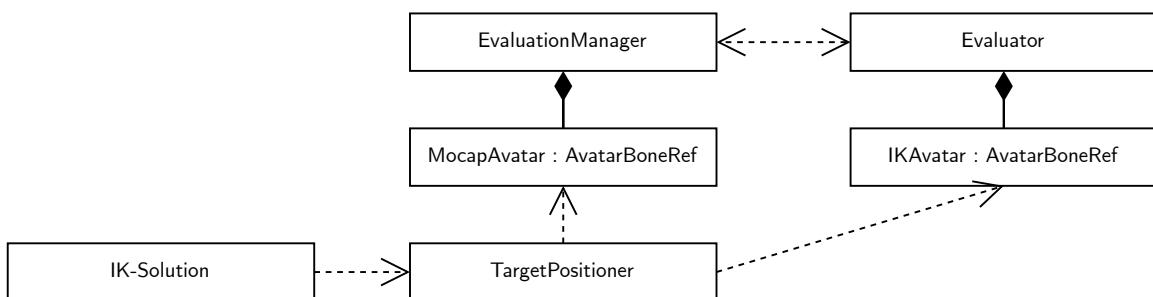
Um vergleichbare Daten der Gelenkpositionierung durch die IK-Lösungen sammeln zu können, wurden Motion Capture-Daten verwendet. So können Gelenk-Positionierungsfehler der IK-Lösungen unter den gleichen Voraussetzungen ermittelt werden, indem die IK-Lösungen dieselben Effektor-Daten als Grundlage erhalten.

Es wurden die Motion Capture-Daten von der Carnegie Mellon Universität [54] verwendet. Diese beinhalten eine Vielzahl an kostenlosen und unterschiedlichen Datensätzen. Für die Verwendung in Unity existieren die Assets *Huge FBX Mocap Library part 1-3* [55]–[57], welche die Motion Capture-Daten in dem vom Unity unterstützen Format *fbx* bereitstellen.

Zur Umsetzung in Unity wurde der enthaltene Avatar aus dem Asset *Huge FBX Mocap Library* verwendet. Um die Qualität der Arm-Posenschätzungen zu evaluieren, wurden 100 Animationen mit einer Gesamtdauer von ca. 25 Minuten ausgewählt. Diese beinhalten vielfältige Armbewegungen, wie zum Beispiel Winken, Unterhaltungsgesten, Essen sowie Sportbewegungen. Die Animationen wurden mithilfe eines Animation Controllers aneinandergereiht und aufeinanderfolgend abgespielt.

In Abbildung 4.1 ist ein vereinfachtes Objektdiagramm zur Veranschaulichung der Struktur der Implementierung abgebildet. In der Klasse *EvaluationManager* werden die Daten vom Motion Capture-Avatar bereitgestellt sowie einige generelle Größen für die Evaluation bestimmt. Nach dem Durchlauf aller Animationen werden hier die Daten aller *Evaluators* zusammengefasst und in eine Datei geschrieben.

Für jede IK-Lösung gibt es einen *Evaluator* und einen *TargetPositioner*. Im *TargetPositioner* werden die Effektoren für die IK-Lösung positioniert und rotiert. Je nach IK-Lösung können die Effektoren unterschiedlich sein, wie im nächsten Abschnitt beschrieben wird. Zu den Positions-



**Abbildung 4.1:** Vereinfachtes Objektdiagramm der Evaluation

daten des Motion Capture-Avatars, welche vom *EvaluationsManager* abgefragt werden, wird ein Offset addiert. Der Offset repräsentiert die Verschiebung vom IK-Avatar ausgehend vom Motion Capture-Avatar. Die Verschiebung ermöglicht die Darstellung aller Avatare nebeneinander, um auch optisch die IK-Lösungen vergleichen zu können. Um gleichzeitig Evaluationsdaten sammeln zu können, wird der Offset bei der Fehlerberechnung berücksichtigt.

Der *Evaluator* evaluiert die IK-Lösungsdaten mit den Motion Capture-Daten. In jedem *FixedUpdate* werden die Positionierungsfehler der Gelenke errechnet. Es wird die Distanz zwischen der IK-Gelenksposition und der Motion Capture-Gelenkposition bestimmt und in Zentimeter umgerechnet.

## 4.2 Implementierung der IK-Lösungen

Im Folgenden wird die Implementierung der IK-Lösungen erläutert. VRIK und VRArmIK sind im Asset Store erhältlich, daher mussten diese nur integriert und für die Evaluation angepasst werden. Two Bone IK und Unity IK wurden auf Grundlage ihrer Dokumentationen implementiert.

- **VRIK**

Bei der Lösung VRIK von Final IK werden die Standardeinstellungen verwendet, abgesehen von der Option *plantFeets*. Diese Option platziert die Füße auf dem Boden, auch wenn die Kopfposition zu hoch ist, beispielsweise beim Springen. Dadurch kann es sein, dass der echte Kopf höher ist als der Kopf des Avatars. Da die Animationen aus dem Motion Capture-Datensatz allerdings vereinzelte Sprünge enthalten, würde diese Option zu einem fehlerhaften Ergebnis führen.

- **VRArmIK**

VRArmIK wird ebenfalls mit den Standardeinstellungen verwendet. Allerdings benutzt VRArmIK einen eigenen Arm-Avatar, bei dem die einzelnen Körperteile durch Quader dargestellt werden (siehe Abbildung 4.2). Aus diesem Grund muss der VRArmIK-Avatar passend zum verwendeten Avatar skaliert werden, sodass die Körpergröße und die Körperproportionen identisch sind. Dazu werden die Avatarhöhe, die Armspannweite, die Schulterbreite, die Kopf-Nacken-Distanz und die Nacken-Schulter-Distanz als Werte in VRArmIK eingetragen. Zusätzlich wurde VRArmIK so angepasst, dass das Oberarm-Unterarmverhältnis dem des verwendeten Avatars entspricht. Des Weiteren werden die Gelenkpositionen und -rotationen von dem VRArmIK-Avatar auf den Ganzkörperavatar angewandt (siehe Abbildung 4.2), um die IK-Lösungen auch optisch miteinander vergleichen zu können.

- **Two Bone IK**

Die Unity Lösung mittels des AnimationRigging Pakets wurde auf Grundlage des enthaltenen Beispiel-Projekts sowie eines Tutorials [58] implementiert. Die Parameter wurden für eine bestmögliche Ellenbogen-Positionierung angepasst. Es wird ein *TwoBoneIK-Constraint* für beide Arme verwendet. Da nur die Gelenkposen der Gelenkkette vom Oberarm zur Hand berechnet werden, wurde die Oberarmposition gegeben. Es wird die Oberarmposition von VRIK verwendet, um die Lösung vergleichen zu können.

- **Unity IK**

Die Unity Lösung durch den Animator wurde auf Basis der Unity Dokumentation [50] implementiert und anschließend erweitert. Es werden die Hände, der Mittelpunkt der Augen sowie der Schwerpunkt des Avatars gegeben. Der Schwerpunkt des Avatars wird vom Motion Capture-Avatar bezogen.

In Abbildung 4.2 sind links die Pose des Motion Capture-Avatars und rechts die ermittelten Posen der IK-Lösungen dargestellt. Die Effektoren (z. B. Hände und Kopf) sind durch lila Kugeln markiert. Die Gelenke, welche durch die IK-Lösung platziert werden, sind durch türkise Kugeln gekennzeichnet.

**Abbildung 4.2:** Veranschaulichung der gegebenen und gelösten Gelenke der IK-Lösungen

## 4.3 Auswertung

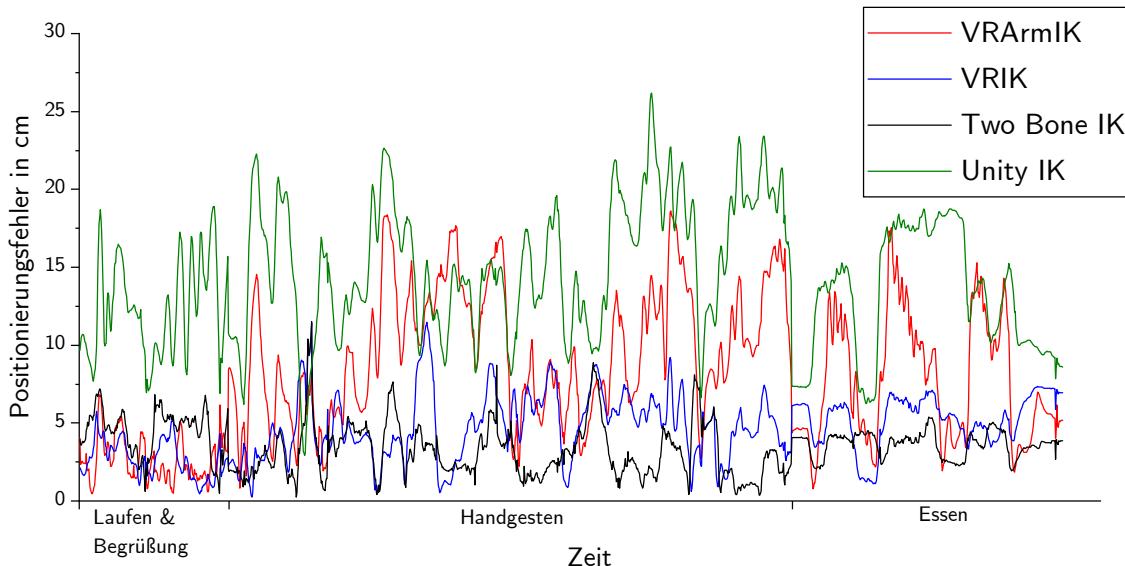
Die gewonnenen Evaluationsdaten werden im Folgenden analysiert.

Beim Testen in VR fielen beim subjektiven Betrachten keine großen Unterschiede zwischen den untersuchten IK-Lösungen auf. Wie zu erwarten war, zeigte sich bei der Analyse der Daten, dass der Ellenbogen-Positionierungsfehler vom Oberarm-Positionierungsfehler abhängig ist (siehe Abbildung A.4). Weiterhin wurde festgestellt, dass die Positionierungsfehler nicht von der Geschwindigkeit der Hände abhängig sind. Außerdem konnte beobachtet werden, dass der Ellenbogen-Positionierungsfehler bei Unity IK und VRArmIK bei kleiner werdendem Innenarmwinkel tendenziell größer wird (siehe Abbildung A.5). Mit Innenarmwinkel ist der Winkel zwischen dem Ober- und Unterarm gemeint. Weitere signifikante Abhängigkeiten vom Innenarmwinkel konnten nicht festgestellt werden.

Im Folgenden werden die Positionierungsfehler der IK-Lösungen verglichen, um die Lösung zu finden, welche die Gelenke mit den geringsten Fehlern positioniert. Dabei wird vor allem der Ellenbogen-Positionierungsfehler betrachtet, da dieser das kritischste Element ist. Er ist für den/die Nutzer\*in gut sichtbar. Fehlpositionierungen würden deshalb am stärksten auffallen.

Das Diagramm in Abbildung 4.3 stellt den Positionierungsfehler des rechten Ellenbogens der untersuchten IK-Lösungen während drei Beispiel-Animationen dar. Die Animationen beinhalten vielfältige Armbewegungen. In der ersten Animation läuft der Avatar einige Schritte und begrüßt jemanden mit Händeschütteln. In der zweiten Animation erklärt der Avatar gestenreich und in der Dritten bereitet er Essen zu und führt es zum Mund.

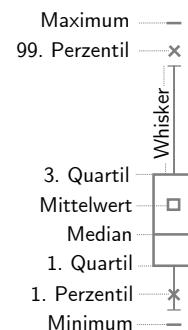
Dem Diagramm ist entnehmbar, dass die Fehler bei VRArmIK und Unity IK deutlich größer sind als bei VRIK und Two Bone IK. Zudem schwanken sie stärker. Diese Beobachtung ist wichtig,



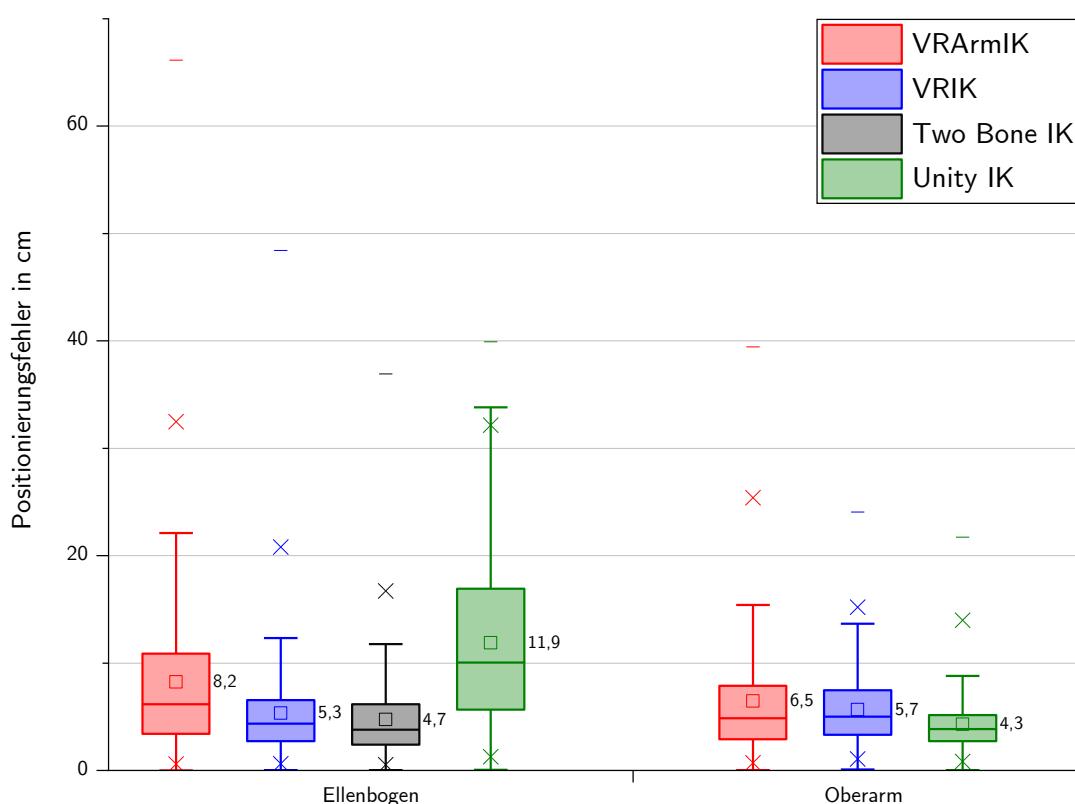
**Abbildung 4.3:** Positionierungsfehler des rechten Ellenbogens bei Bewegungen

da es für die Nutzung ungünstig ist, wenn sich die Ellenbogenposition zu abrupt ändert. VRIK und Two Bone IK haben zum Teil ähnliche Fehlerwerte, wobei Two Bone IK in den letzten beiden Animationen etwas geringere Fehler aufweist.

Der Box-Plot<sup>6</sup> in Abbildung 4.5 stellt die Daten von Ellenbogen- und Oberarm-Positionierungsfehler dar. Dazu wurden jeweils die Daten von linkem und rechtem Ellenbogen sowie der Oberarme zusammengefasst. Vergleicht man die Boxplots der Ellenbogen, ist ersichtlich, dass VRIK und Two Bone IK die Ellenbogen besser positionieren als VRArmIK und Unity IK. Deren Median, Mittelwert und auch der Interquartilsabstand ist geringer. Das bedeutet, dass durchschnittlich die Positionierungsfehler kleiner sind und die Streuung geringer ist. Dies wird auch durch die geringere Länge der Whiskers und den kleineren 99. Percentilen deutlich. Auch die Standardabweichung von Two Bone IK (3,4 cm) und VRIK (4,1 cm) ist geringer als die von VRArmIK (6,9 cm) und Unity



**Abbildung 4.4:**  
Box-Plot Legende



**Abbildung 4.5:** Box-Plot der Ellenbogen- und Oberarm-Positionierungsfehler

<sup>6</sup>Ein **Box-Plot** ist ein Diagramm, welches die Verteilung sowie wichtige Lage- und Streuungsmaße darstellt. Die Box wird durch das untere Quartil und das obere Quartil auch 1. und 3. Quartil genannt, begrenzt. Quartil bedeutet Viertel, somit gibt das 1. Quartil den Wert an, unter dem 25% der Werte liegen und das 3. Quartil den Wert unter dem 75% der Werte liegen. Eine Linie durch die Box zeigt den Median an. Von den Quartilen aus werden sogenannte Whisker gezeichnet. Die Länge der Whisker berechnet sich aus dem Datenwert, der noch innerhalb des 1,5-Fachen des Interquartilsabstands liegt. In Abbildung 4.4 ist eine Legende abgebildet.

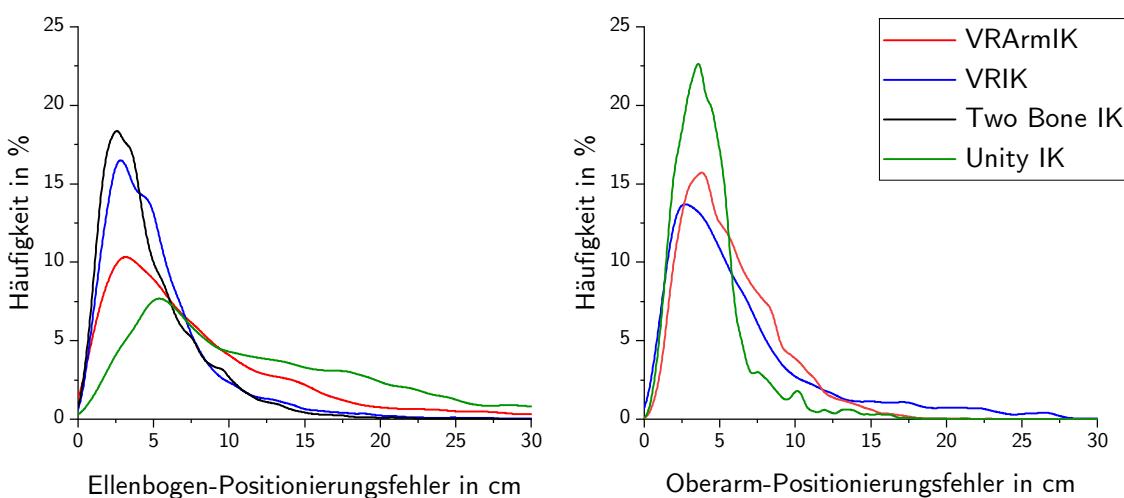
IK (7,6 cm).

Two Bone IK positioniert den Ellenbogen ein wenig besser als VRIK, im Durchschnitt um 6 mm. Der Median von VRIK liegt bei 4,4 cm, der von Two Bone IK bei 3,8 cm.

Bei den Oberarmfehlern sind die Unterschiede nicht ganz so erheblich wie bei den Ellenbogenfehlern. Da Two Bone IK die Oberarme nicht positioniert, gibt es keine Oberarm-Daten für diese Lösung. Unity IK hat einen großen Vorteil gegenüber den anderen IK-Lösungen bei der Oberarm-Positionierung. Es wird die korrekte Position und Rotation des Massenschwerpunktes vom Avatar gegeben, wodurch der Oberkörper einfacher gelöst werden kann. Dies spiegelt sich in den Daten wider, die Oberarm-Positionierung von Unity IK hat die niedrigsten Fehler. Die anderen Lösungen müssen die Position und Rotation des Masseschwerpunkts selber ermitteln, dadurch sind sie nicht direkt mit Unity IK vergleichbar. Beim Einsatz in VRReha ist der Massenschwerpunkt nicht bekannt, daher könnte Unity IK dort nur in Verbindung mit VRIK eingesetzt werden.

Die Qualität der Oberarm-Positionierung von VRArmIK und VRIK unterscheidet sich geringfügig: Der Median von VRArmIK ist ein wenig kleiner (VRArmIK: 4,9 cm, VRIK: 5,0 cm). Die Standardabweichung von VRIK ist hingegen deutlich geringer (VRArmIK: 5,3 cm, VRIK: 3,1 cm). Dies ist auch an den Werten des ersten und dritten Quartils ersichtlich, welche bei VRIK dichter am Median liegen.

In Abbildung 4.6 sind die Verteilungskurven des Positionierungsfehlers dargestellt. Um die Kurve zu approximieren, wurde das Kerndichteschätzer-Verfahren<sup>7</sup> verwendet. Es veranschaulicht, wie auch das Box-Plot, die geringere Streubreite und die kleineren Fehlerwerte von VRIK und Two Bone IK bei der Positionierung der Ellenbogen. Durch diese Darstellung wird deutlich, dass bei VRArmIK und Unity IK größere Fehlerwerte häufiger vorkommen. Auch die Erkenntnisse aus dem Box-Plot zur Oberarm-Positionierung werden in dieser Darstellung deutlich.



**Abbildung 4.6:** Kerndichteschätzung der Positionierungsfehler

<sup>7</sup>Das **Kerndichteschätzungs-Verfahren** wird dazu verwendet, eine stetige Wahrscheinlichkeitsverteilung zu approximieren.

Zusammenfassend lässt sich aus den erhobenen Daten erkennen, dass Two Bone IK die Ellenbogen und Unity IK die Oberarme am präzisesten positioniert. Da Unity IK allerdings die Ellenbogen nicht präzise positioniert und den Masseschwerpunkt des Avatars zum Lösen benötigt, ist es keine geeignete Lösung. Für die Benutzung von Two Bone IK müsste es mit VRIK kombiniert werden, da Two Bone IK nur die Ellenbogen positionieren kann. Da sich Two Bone IK und VRIK in der Qualität der Ellenbogen-Positionierung allerdings nur sehr geringfügig unterscheiden, wurde beschlossen, VRIK als Ausgangspunkt der Implementierung zu wählen.

Im Anhang A finden sich weitere Diagramme und deskriptive Statistiken.

# 5 Integration in VReha

In diesem Kapitel werden die Integration der IK-Lösung und die Implementierung zusätzlicher Erweiterungen beschrieben.

## 5.1 Grundlegende Softwarestruktur

Die Erstellung und Konfiguration der IK-Lösung und des Avatars wird durch die Klasse *AvatarManager* organisiert. Es wurden die Avatare aus der Microsoft Rocketbox Avatar Sammlung [59] verwendet. Diese beinhaltet 115 vielfältige Avatare, die direkt in Unity verwendet werden können.

In einer Konfigurationsdatei kann definiert werden, ob ein Avatar oder nur Hand-Modelle verwendet werden sollen sowie Parameter zur Spezifizierung der IK-Lösung. Soll ein Avatar verwendet werden, werden die ohne Avatar verwendeten Hand-Modelle deaktiviert und je nach Geschlecht des/der Benutzenden ein männlicher oder weiblicher Avatar erstellt.

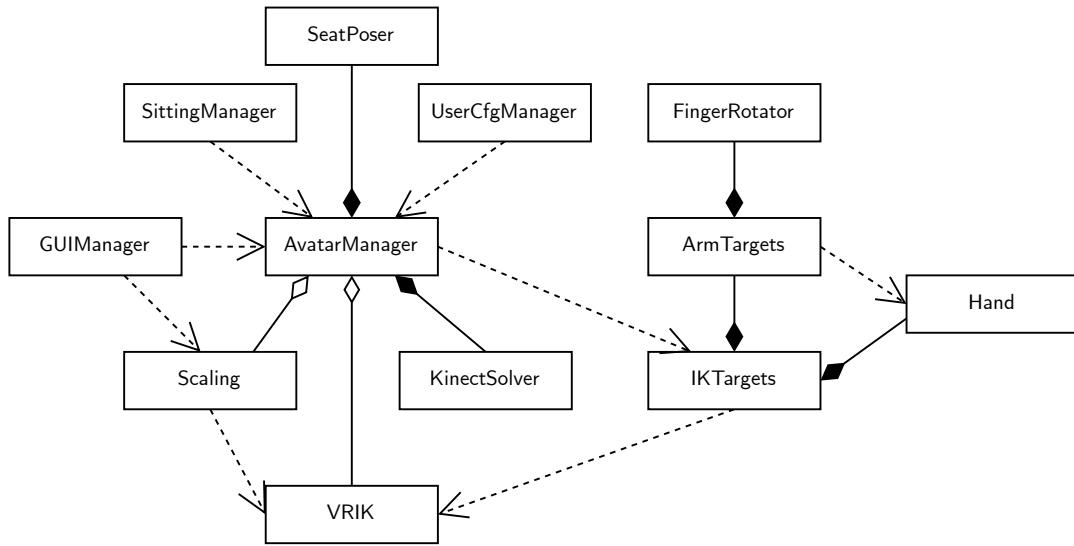
Benutzt der/die Nutzer\*in die Anwendung das erste Mal, werden Höhe, Armlänge und Hüfthöhe des/der Nutzers\*in mittels der Klasse *Scaling* ermittelt und der Avatar passend skaliert. Der Skalierungsprozess wird über den *GUIManager* gesteuert. Bestehende Skalierungsdaten werden über eine Konfigurationsdatei durch den *UserCfgManager* eingelesen. Der Skalierungsprozess wird in Abschnitt 5.3 näher erläutert.

Für VRIK müssen Effektoren positioniert werden. Der Kopf- und die Arm-Effektoren werden mittels der Klasse *IKTargets* gesetzt, näheres dazu in Abschnitt 5.6.

Zusätzlich werden Hüft- und Fuß-Effektoren vom *KinectSolver* positioniert. Es werden nur die Kinect-Daten des Unterkörpers verwendet, da beim Oberkörper Ungenauigkeiten, Trackingfehler und Latenzen deutlich stärker auffallen. Die Arm-Effektoren werden nur mit Kinect-Daten positioniert, wenn keine anderen Sensordaten zur Verfügung stehen. Die Verwendung der Kinect wird in Abschnitt 5.7 erklärt.

Wird die Anwendung sitzend ausgeführt, werden mit der Klasse *SeatPoser* der Hüft- und die Fuß-Effektoren für eine Sitzpose gesetzt (Abschnitt 5.5).

Die Kollaboration der verschiedenen Klassen ist im folgenden Diagramm dargestellt:



**Abbildung 5.1:** Klassenübersicht der Implementierung

## 5.2 Integration von VRIK

Die Evaluation hat ergeben, dass VRIK die am besten geeignete Lösung ist, weshalb diese für die Umsetzung gewählt wurde.

Zur Integration muss das Skript `VRIK.cs` dem *GameObject* des Avatars hinzugefügt werden. Hat der Avatar eine humanoide Animator-Komponente, werden die Referenzen automatisch gesetzt. Diese beinhalten die Transform-Komponenten der Gelenke.

Es können Effektoren für Kopf, Hände, Hüfte und Füße definiert werden. Zusätzlich können die Positionierungs- und Rotationswerte der Effektoren gewichtet werden. Für die Arme und Beine können zusätzlich noch sogenannte Bend-Goals angegeben werden. Diese können die Richtung der Beugung der Ellenbogen beziehungsweise der Knie beeinflussen.

Des Weiteren können zahlreiche Parameter angepasst werden, wie zum Beispiel der gewünschte Abstand der Füße zueinander oder die Distanz, ab welcher ein Schritt ausgelöst werden soll. Eine Dokumentation von VRIK ist auf der Website von Root Motion [60] zu finden.

Konkret wird VRIK vom `AvatarManager` nach einer erfolgreichen Skalierung des Avatars diesem hinzugefügt und konfiguriert.

Zusätzlich wird das Skript `PlatformManager` von Final IK verwendet. Ohne dieses Skript würden sich beispielsweise die Füße des Avatars bewegen, wenn sich das Elternelement des Avatars nach vorne bewegt. In VReha ist dies unerwünscht: Bewegt sich der/die Nutzer\*in beispielsweise durch Knopfdruck oder Beinbeugung vorwärts, soll der Avatar nicht dadurch beeinflusst werden. Dies wird durch den `PlatformManager` verhindert.

## 5.3 Skalierung des Avatars

Eine falsche Skalierung des Avatars führt zu unrealistischen Positionierungen der Gelenke. Sind beispielsweise die Arme des Avatars länger als die des/der Benutzenden, werden diese gebeugt, um die Handpositionen des/der Nutzers\*in zu erreichen, auch wenn diese\*r die Arme ausstreckt. Ähnliches gilt für die Körperhöhe: Ist der Avatar zu groß, werden die Beine und gegebenenfalls der Oberkörper gebeugt. Um einen möglichst realistischen Avatar zu erzeugen, muss dieser also passend zum/zur Benutzenden skaliert werden.

Die Skalierung wurde in der Klasse *Scaling* implementiert, sie ist in drei Teile gegliedert:

### 1. Skalierung der Körpergröße

Als Erstes wird der Avatar gemäß der Höhe des/der Benutzenden skaliert. Dazu wird aus der Kopfhöhe des/der Anwenders\*in, welche mittels VR-Headset bestimmt wird, und der Kopfhöhe des Avatars ein Skalierungsfaktor errechnet. Mit diesem Wert wird der Avatar in Höhe, Breite und Tiefe skaliert. Der Vorgang ist im Quellcode 5.1 dargestellt.

```
void scaleHeight(float heightUser)
{
    float scaleVal = heightUser / convertToLocal(head.position).y;
    avatar.transform.localScale *= scaleVal;
}
```

**Quellcode 5.1:** Skalierung des Avatars gemäß der Körpergröße

### 2. Skalierung der Arme

Durch die Skalierung der Körpergröße wurde die Armlänge bereits angepasst. Allerdings kann die Armlänge des Avatars im Vergleich zum/zur Benutzenden noch einige Zentimeter zu groß oder zu klein sein. Dies ist für die Posenschätzung ungünstig. Zum Beispiel würden die Arme bei einer zu großen Armlänge des Avatars fälschlicherweise gebeugt werden.

Um die Armlänge zu bestimmen, wurden verschiedene Ansätze getestet. Durch eine T-Pose<sup>8</sup> kann die Armspannweite bestimmt werden. Aus dieser kann durch Subtraktion der Schulterbreite die Armlänge errechnet werden. Dieser Ansatz ist allerdings nur mit Tracking durch die Kinect oder mit Controllern möglich, da beim Handtracking durch die Oculus Quest oder die Leap Motion die Hände nur im Sichtfeld erkannt werden können. Für die Integration in VReha wäre eine Skalierungslösung allerdings ohne Kinect oder Controller besser geeignet, da nicht immer eine Kinect vorhanden ist und ein Controller für die imVST-Anwendung an einer Lenkerstange montiert ist.

Für die Armlängenbestimmung mit der Oculus Quest oder Leap Motion müssen die Hände im Sichtfeld positioniert werden. Streckt der/die Benutzende die Arme gerade nach vorne aus, kann die Armlänge gemessen werden, wenn die Schulterpositionen bekannt sind. Dazu müssen

<sup>8</sup>**T-Pose:** Die Person steht gerade und steckt die Arme gerade zur Seite aus.

die Schulterpositionen approximiert werden. Es wurden dazu einige eigene Ansätze getestet – mit dem Ergebnis, dass VRIK die Schultern am präzisesten approximiert.

Für die Armlängenbestimmung wird kurzzeitig ein zweiter in der Höhe skalierter Avatar mit der VRIK-Komponente erzeugt und die Armlänge gemessen. Damit VRIK genug Zeit hat, die Gelenke zu positionieren, wird 50 Frames lang gewartet, bevor die Armlänge bestimmt wird. Der Vorgang ist im folgenden Quellcode dargestellt:

```
IEnumerator getAndSetArmlength()
{
    var ikAvatar = Instantiate(AvatarManager.Instance.avatar);
    //set avatar invisible
    ikAvatar.GetComponentInChildren<SkinnedMeshRenderer>().enabled = false;
    //set up IK and target anchors
    ikAvatar.SetActive(true);
    var ik = ikAvatar.AddComponent<VRIK>();
    ik.AutoDetectReferences();
    ik.GuessHandOrientations();
    yield return new WaitUntil(() => ik.componentInitiated == true);
    IKTargets.Instance.setUpHandsRotation(ik);
    var headTarget = Instantiate(IKTargets.Instance.headTargetAnchor,
        ↳ IKTargets.Instance.headTargetAnchor.position,
        ↳ IKTargets.Instance.headTargetAnchor.rotation);
    var lHand = Instantiate(IKTargets.Instance.leftArm.hand,
        ↳ IKTargets.Instance.leftArm.hand.position,
        ↳ IKTargets.Instance.leftArm.hand.rotation);
    var rHand = Instantiate(IKTargets.Instance.rightArm.hand,
        ↳ IKTargets.Instance.rightArm.hand.position,
        ↳ IKTargets.Instance.rightArm.hand.rotation);
    ik.solver.spine.headTarget = headTarget;
    ik.solver.leftArm.target = lHand;
    ik.solver.rightArm.target = rHand;
    for (int i = 0; i < 50; i++)
        yield return new WaitForEndOfFrame();
    //calculate arm length
    var armLengthL = Vector3.Distance(ik.references.leftUpperArm.position,
        ↳ ik.solver.leftArm.target.position);
    var armLengthR = Vector3.Distance(ik.references.rightUpperArm.position,
        ↳ ik.solver.rightArm.target.position);
    AvatarManager.Instance.avatarVals.armLength = (armLengthL + armLengthR) / 2;
    //destroy target anchors
    Destroy(ikAvatar.gameObject);
    Destroy(headTarget.gameObject);
    Destroy(lHand.gameObject);
    Destroy(rHand.gameObject);
}
```

**Quellcode 5.2:** Bestimmung der Armlänge

### 3. Skalierung des Ober- und Unterkörpers

Um die Beine vor allem in der Sitz-Variante realistisch darzustellen, ist es wichtig, dass die Hüfte auf der richtigen Höhe positioniert ist. Die Höhe der Hüftgelenke wird bestimmt, indem der/die Benutzende die Hände auf Hüftgelenkhöhe positioniert. Durch das Tracking der Hände kann somit die Hüftgelenkhöhe erfasst werden und die Hüfte des Avatars dementsprechend verschoben werden. Anschließend muss die Größe des Ober- und Unterkörpers angepasst werden. Die Skalierung kann nicht durch die Skalierung des *GameObjects* durchgeführt werden, da der Oberkörper kein übergeordnetes *GameObject* hat. Alle Knochen des Oberkörpers müssen deshalb entsprechend verschoben werden. Die Beine haben ein übergeordnetes *GameObject*, daher können sie mit dem gewohnten Verfahren skaliert werden.

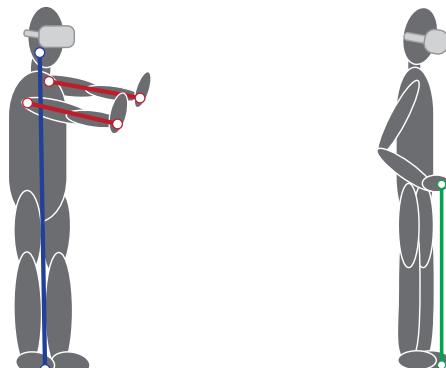
Zusätzlich wurde eine Skalierung der Schulterbreite in Betracht gezogen. Allerdings sind die Schultern des Avatars nicht gut sichtbar und daher deren Breite nicht so relevant. Außerdem ist es kompliziert und fehleranfällig, die Schulterbreite genau anhand der Hand- und Kopfposition zu messen.

#### Durchführung

Bei erstmaliger Benutzung muss die Person den beschriebenen Skalierungsprozess durchführen. Dazu wurde eine GUI entwickelt. Der/die Versuchsleiter\*in gibt als Erstes das Geschlecht des/der Patienten\*in an. Für diese Arbeit wurden ein weiblicher und ein männlicher Avatar integriert. In Zukunft sollen mehrere Avatare zur Auswahl stehen, beispielsweise mit unterschiedlicher Kleidung, Hautfarben und Alter.

Der/die Patient\*in muss anschließend die Skalierungsposen (siehe Abbildung 5.2) einnehmen, welche in der GUI veranschaulicht sind.

Als Erstes muss der/die Patient\*in gerade stehen und die Arme gerade nach vorne ausstrecken.



(a) Ermittlung der Körpergröße und Armlänge

(b) Ermittlung der Hüftgelenkhöhe

**Abbildung 5.2:** Posen für die Skalierung des Avatars

Auf Knopfdruck durch den/die Versuchsleiter\*in wird die Körpergröße und die Armlänge bestimmt. Anschließend muss der/die Patient\*in die Hände auf Höhe der Hüftgelenke platzieren und nach unten schauen, damit die Hände vom Handtracking der Leap Motion oder Oculus Quest erfasst werden können. Die Höhe der Hüftgelenke wird wieder auf Knopfdruck gespeichert und der Avatar daraufhin skaliert.

Der/die Patient\*in kann nun seinen Avatar sehen und testen, ob die Skalierung erfolgreich war. Sollte dies nicht der Fall sein, kann der Skalierungsprozess wiederholt werden.

Die Skalierungsdaten werden in eine Datei geschrieben und können so bei der nächsten Sitzung wiederverwendet werden. Die Datei wird mit der Studien-ID, die beim Start der Anwendung eingegeben wird, einer Person zugeordnet.

## 5.4 Konfiguration des Avatars

Nach der Skalierung wird der Avatar konfiguriert und ist sichtbar.

VRIK wird dem Avatar hinzugefügt, konfiguriert und die Effektoren werden definiert. Wenn die Daten des Kinect-Trackings verwendet werden sollen und der Avatar nicht sitzt, wird ein *KinectSolver* erzeugt, der den Hüft- und die Fuß-Effektoren durch die Kinect positioniert.

Damit die Finger des Avatars durch den/die Nutzer\*in bewegbar sind, müssen auf diese entweder die Leap Motion-Daten oder Oculus Quest-Daten angewandt werden. Für die Leap Motion gibt es ein bereitgestelltes Skript, welches die Fingerrotationsdaten auf Hand-Modelle anwendet. Für die Oculus Quest wurde eine eigene Lösung implementiert. Die Rotationsdaten der Fingergelenke werden damit auf die des Avatars angewandt.

Zusätzlich wird ein *Dictionary* erzeugt, welches die lokalen Positions- und Rotationsdaten der Knochen in der T-Pose enthält. Dies wird für die Sitz-Lösung sowie für die Platzierung der Fuß-Effektoren durch die Kinect benötigt. Die Erstellung ist im folgenden Quellcode dargestellt:

```
var skeleton = animator.avatar.humanDescription.skeleton;
var human = animator.avatar.humanDescription.human;
skeletonData = skeleton.Join(human, skeletonVals => skeletonVals.name, humanVals =>
    → humanVals.boneName, (skeletonVals, humanVals) => new { humanVals.humanName,
    → skeletonVals }).ToDictionary(x => x.humanName, x => new
    → Bone(x.skeletonVals.rotation, x.skeletonVals.position));
//set hip world rotation (some avatars have a rotated parent object)
var hipBone = new Bone(
    animator.GetBoneTransform(HumanBodyBones.Hips).rotation,
    skeletonData["Hips"].position);
skeletonData["Hips"] = hipBone;
```

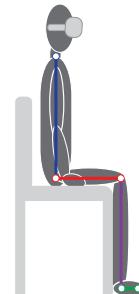
**Quellcode 5.3:** Erzeugung eines Dictionarys für die lokalen Positions- und Rotationsdaten der Knochen in der T-Pose

## 5.5 Positionierung und Ausrichtung der Effektoren für die Sitzpose

In VReha gibt es die Option, die Anwendung sitzend durchzuführen. Für diesen Fall muss die Möglichkeit implementiert werden, dass der Avatar sitzen kann. Da VRIK den Avatar anhand seines Schwerpunktes ausrichtet, kniet der Avatar, wenn der/die Benutzer\*in sitzt und nur Kopf- und Hand-Effektoren gegeben sind. Sind Hüft- und Fuß-Effektoren gegeben, kann VRIK den Avatar korrekt sitzend positionieren. Diese Effektoren werden basierend auf der Kopfposition fest platziert. Es werden keine Kinect-Daten verwendet, da die Kinect die Gelenkpositionen bei sitzenden Personen nicht gut erkennt.

Für die Positions berechnung der Effektoren muss der/die Nutzer\*in gerade sitzen. Mit der Oberkörperhöhe, welche bei der Skalierung bestimmt wurde, wird der Hüft-Effektor unter dem Nacken des Avatars positioniert. Anschließend werden die Hüftgelenke als Hilfestellung für die Fuß-Effektor-Positionierung platziert. Als Kindelemente werden die Fuß-Effektoren erzeugt. Diese werden ausgehend vom Hüftgelenk um die Oberschenkellänge und die Fußlänge nach vorne verschoben. Die Höhe der Fuß-Effektoren wird entsprechend dem Abstand von den Zehen zum Boden gesetzt. Dieser wurde vor der Initialisierung von VRIK bestimmt, sodass die Füße zu diesem Zeitpunkt auf dem Boden standen. Zusätzlich wird die Rotation der Zehen in der T-Pose des Avatars auf die Effektoren angewandt, sodass die Füße nach vorne ausgerichtet sind. Alle Berechnungen werden mit lokalen Transformationswerten durchgeführt, um die Effektoren auch bei einem gedrehten Elternelement korrekt zu positionieren. Die Platzierung der Fuß-Effektoren ist in Abbildung 5.3 veranschaulicht.

In der imVST-Anwendung wird durch eine Konfigurationsdatei festgelegt, ob die Anwendung sitzend oder stehend durchgeführt wird. Bei der imVMT-Anwendung ist dies nicht festgelegt und es wird anhand der Kopfhöhe bestimmt, ob der/die Nutzer\*in sitzt. Wenn dies der Fall ist, wird der Avatar sitzend positioniert und ein Stuhl platziert. Die Sitzpose wird auf Knopfdruck eingenommen. In Abbildung 5.5 sind sitzende Avatare abgebildet.



**Abbildung 5.3:**  
Positionierung  
der Effektoren



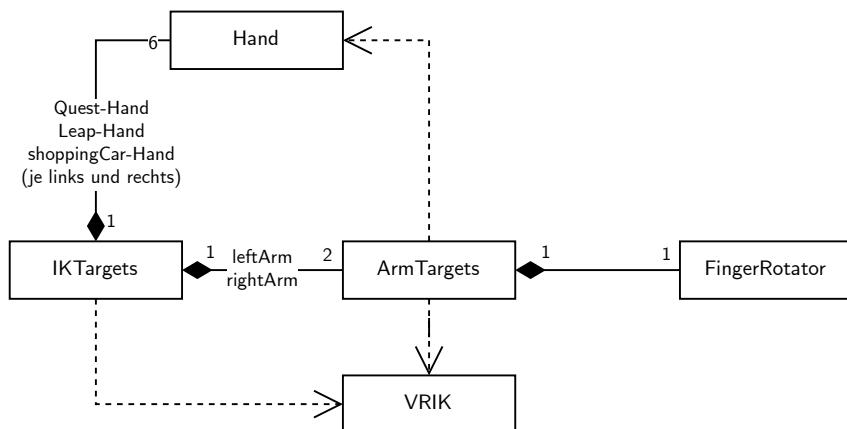
**Abbildung 5.4:** Sitzende Avatare in VReha

## 5.6 Positionierung und Ausrichtung der Kopf- und Arm-Effektoren

In diesem Abschnitt werden die Positionierung und Ausrichtung der Kopf-, Hand- und Ellenbogen-Effektoren erläutert.

Abbildung 5.5 gibt einen Überblick über die Klassenstruktur. Die Klasse *IKTTargets* verwaltet die Platzierung und Ausrichtung der Effektoren. Diese ermittelt die Rotationsoffsets der Kopf- und Handgelenke vom Avatar zu den Gelenkorientierungen der verschiedenen Sensordaten (siehe Abschnitt 5.6.1). *IKTTargets* besitzt sechs *Hand*-Objekte. Diese definieren für jeden Sensor die Ausrichtung des Handgelenks sowie den beim Programmstart ermittelten Rotationsoffset zum Avatar-Handgelenk. In *IKTTargets* werden zusätzlich der Kopf-Effektor sowie beide *ArmTargets* aktualisiert. Die Klasse *ArmTargets* positioniert den Ellenbogen- und den Hand-Effektor je nach Datenverfügbarkeit, was in Abschnitt 5.6.3 näher beschrieben wird.

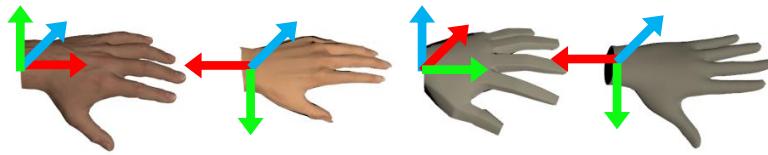
Alle Effektoren werden lokal zum Elternelement des Avatars positioniert und rotiert. Dies ermöglicht eine Verschiebung und Rotation des Elternelements, bei der sich der Avatar mitbewegt.



**Abbildung 5.5:** Klassendiagramm für die Positionierung und Ausrichtung der Kopf- und Arm-Effektoren

### 5.6.1 Ermittlung der Rotationsoffsets

Die Achsenorientierungen der Gelenke verschiedener Avatare beziehungsweise verschiedener Sensordaten können unterschiedlich sein, wie in Abbildung 5.6 am Beispiel mehrerer Hände veranschaulicht ist. Für die Nutzung von VRIK müssen die Achsenorientierungen der Effektoren mit denen des Avatars übereinstimmen. Will man beispielsweise die Hand-Daten der Oculus Quest auf eine Avatarhand anwenden, müssen die Daten so rotiert werden, dass die Achsenorientierung der Orientierung der Avatarhand entspricht. Das heißt, die Achsen müssen dieselben Richtungen definieren wie die der Avatarhand, beispielsweise die Richtung vom Handgelenk zum Daumen.



**Abbildung 5.6:** Achsenorientierungen der Handgelenke bei verschiedenen Hand-Modellen

### Ermittlung der Hand-Rotationsoffsets

Für jeden Handtrackingsensor (Oculus Quest und Leap Motion) gibt es für die linke und rechte Hand ein Objekt des Typs *Hand*, welches die Achsenorientierung definiert. Dazu werden die Achsenrichtungen vom Handgelenk zur Handfläche und von der Handfläche zum Daumen verwendet. Für die Avatarhände können diese Werte von VR IK abgefragt werden. Aus diesen Werten kann der Rotationoffset zwischen der Orientierung der Avatarhand und der einer durch einen Sensor gegebenen Hand mittels folgender Formel berechnet werden:

$$Rot = \begin{pmatrix} \vec{wtp}_{Input}^T \\ \vec{ptt}_{Input}^T \\ \vec{wtp}_{Input}^T \times \vec{ptt}_{Input}^T \end{pmatrix} \cdot \begin{pmatrix} \vec{wtp}_{Avatar}^T \\ \vec{ptt}_{Avatar}^T \\ \vec{wtp}_{Avatar}^T \times \vec{ptt}_{Avatar}^T \end{pmatrix}^{-1}$$

$\vec{wtp} = \text{wristToPalm}, \vec{ptt} = \text{palmToThumb}$

Die Rotationoffsets werden für jeden Sensor jeweils für die linke und rechte Hand ermittelt und in dem dazugehörigen *Hand*-Objekt gespeichert. Beim Rotieren der Hand-Effektoren wird dann der entsprechende Offsetwert angewandt.

Der Vorgang ist im folgenden Quellcode dargestellt:

```

var avatarHandLeft = getMatrix(ik.solver.leftArm.wristToPalmAxis,
    ik.solver.leftArm.palmToThumbAxis);

setHandRotOffset(avatarHandLeft, leftQuestHand);

void setHandRotOffset (Matrix4x4 avatarHand, Hand hand) {
    var handMatrix = getMatrix(hand.wristToPalm, hand.palmToThumb);
    hand.offset = getRotationOffset(avatarHand, handMatrix);
}

Quaternion getRotationOffset(Matrix4x4 avatar, Matrix4x4 input){
    var rot = input * avatar.inverse;
    return rot.rotation;
}

```

**Quellcode 5.4:** Beispiel der Rotationoffsetberechnung der linken Oculus Quest Hand

### Ermittlung des Kopf-Rotationsoffsets

Für die Ermittlung des Kopf-Rotationsoffsets wird die gleiche Vorgehensweise wie zur Ermittlung der Hand-Rotationsoffsets verwendet. Zum Aufstellen der Matrix werden die Achsenausrichtungen vom Kopf zum Nacken und vom Kopf zum linken Oberarm verwendet. Für die Rotation des Kopfes gibt es nur einen Sensor, sodass in diesem Fall nur ein Offset berechnet werden muss.

### 5.6.2 Positionierung und Ausrichtung des Kopf-Effektors

Für VRIK wird ein Effektor benötigt, der der Position des Kopfes vom Avatar entspricht. Das Kopfgelenk des Avatars entspricht dem oberen Kopfgelenk, also der Gelenkfläche des ersten Halswirbels. Vom Headset wird allerdings nur der Mittelpunkt der Augen bereitgestellt. Daher muss der Effektor so verschoben werden, dass er der Position des Kopfes entspricht. Dazu wird ein *GameObject* erzeugt, welches in jedem Frame entsprechend der Headset-Daten transformiert wird. Zusätzlich wird für dieses ein Kind-*GameObject* erzeugt, welches so transformiert wird, dass es der Kopfposition des Avatars entspricht. Darauf wird auch der ermittelte Rotationoffset angewandt. Dieses *GameObject* wird als Kopf-Effektor verwendet.

### 5.6.3 Positionierung und Ausrichtung der Arm-Effektoren

Die Arm-Effektoren werden durch die Klasse *ArmTargets* positioniert und ausgerichtet. Die Klasse *IKTargets* beinhaltet zwei *ArmTargets*, je eines pro Arm. Diese positioniert die Hand- und Ellenbogen-Effektoren des jeweiligen Arms.

#### Adaption an Datenverfügbarkeit

Je nach Datenverfügbarkeit werden die Hand- und Ellenbogen-Effektoren gesetzt.

Die Sensordaten werden in folgender Reihenfolge priorisiert:

1. Leap Motion/Oculus Quest-Daten
2. Kinect-Daten

Die Leap Motion und die Oculus Quest werden nicht zusammen verwendet, daher haben sie dieselbe Priorisierung. Wird die Hand nicht durch diese Sensoren getrackt, werden die Daten der Kinect nach einer kurzen Wartezeit verwendet. Die Wartezeit ermöglicht, dass die Hände bei kurzzeitigen Trackingproblemen nicht sofort zur Kinect-Position springen. Kurzzeitige Trackingprobleme treten größtenteils auf, wenn die Hände verdeckt oder zu dicht beieinander sind, beispielsweise wenn man in die Hände klatscht. Zusätzlich muss die Hand kurzzeitig durch die Leap Motion oder Oculus Quest getrackt worden sein, bevor die Daten verwendet werden. Somit werden zu Beginn des Trackings auftretende Fehler gefiltert. Dieser Prozess ist im folgenden Quellcode dargestellt:

```

void setTrackingStatusFirstSensor() {
    bool tracked = false;
    bool highConfidence = false;
    if (SensorManager.Instance.UpdateLeapData) {
        tracked = leapHand.isTracked;
        highConfidence = leapHand.Confidence > 0.6f;
    }
    else { //Quest
        tracked = questHand.IsTracked;
        if (questHand.HandConfidence == OVRHand.TrackingConfidence.High)
            highConfidence = true;
        else
            highConfidence = false;
    }
    if (tracked && highConfidence) {
        timeFirstSensorTracked += Time.deltaTime;
        if (timeFirstSensorTracked > MaxTrackingTime) {
            //filter tracking errors when starting to track
            isFirstSensorTracked = true;
            noTrackingTimeFirstSensor = 0;
        }
        else
            isFirstSensorTracked = false;
    }
    else {
        //stay in old position for a while to filter short tracking errors e.g. clapping
        noTrackingTimeFirstSensor += Time.deltaTime;
        if (noTrackingTimeFirstSensor > MaxTimeOutFirstSensor) {
            isFirstSensorTracked = false;
            timeFirstSensorTracked = 0;
        }
    }
}

```

**Quellcode 5.5:** Ermittlung, ob die Daten der Leap Motion/Oculus Quest verwendet werden sollen

Von der Kinect werden nur die Positionsdaten verwendet, da die Rotationsdaten unpräzise sind. Zur Anwendung der Sensordaten werden die Daten vom *SensorManager* benutzt und mit dem jeweils passenden Rotationsoffset multipliziert.

Wird die Hand durch keinen Sensor getrackt, wird sie nach einer kurzen Wartezeit in einer Grundpose positioniert. Diese entspricht der Pose, in der die Hände des Avatars beim Start des Programms waren. So können beliebige Grundposen definiert werden. Ist die Hand in der Grundpose, werden auch die Finger so rotiert, dass sie in der Grundpose sind. Dazu dient die Klasse *FingerRotator*. Dort werden zum Programmstart die Fingerrotationswerte der jeweiligen Hand gespeichert und können bei Bedarf auf die Finger angewandt werden. Die Finger werden auch in ihre Grundpose rotiert, wenn der Hand-Effektor durch die Kinect platziert wird, da die Kinect die Finger nicht trackt.

Nach demselben Prinzip werden die Ellenbogen-Effektoren platziert – diese können mit der Leap Motion oder der Kinect erfasst werden.

Die Platzierung der Effektoren ist pro Arm eigenständig, was die Positionierung der Effektoren mit unterschiedlichen Sensoren ermöglicht. So können beispielsweise die Effektoren des rechten Arms mit der Kinect platziert werden und die des linken Armes mit der Leap Motion.

#### 5.6.4 Platzierung der Hände am Lenker

In der imVST-Anwendung von VReha verwendet der/die Nutzer\*in eine Lenkerstange zum Steuern, welche in VR der Stange eines Einkaufswagens oder dem Lenker eines Elektromobils entspricht. Sind die Hände an der Lenkerstange, kann der/die Nutzer\*in sich vorwärtsbewegen. Um dem/der Nutzer\*in eindeutig Rückmeldung zu geben, wann dies der Fall ist, werden die Hände an der Lenkerstange positioniert, wenn sie sich im nahen Umfeld der Stange befinden. Zusätzlich werden die Finger fest positioniert, sodass sie um die Stange greifen. Dies hat den Vorteil, dass die Finger nicht unlogisch durch das Tracking positioniert werden. In Abbildung 5.7 sind die greifenden Hände an der Lenkerstange dargestellt.

##### Umsetzung

Es werden zwei unsichtbare „Helper-Hände“ an der Lenkerstange platziert. Kommt eine Hand dicht genug an die Lenkerstange, wird sie mittels den Transformationswerten von der Helper-Hand an der Lenkerstange platziert. Die Position und Rotation der Helper-Hand werden auf den Hand-Effektor angewandt. Die Finger werden wie die Finger der Helper-Hand rotiert. Dies ist in der Klasse *FingerRotator* implementiert. Dort werden die Rotationswerte der Finger der Helper-Hand gespeichert. Ist die Hand an der Lenkerstange, werden diese auf die Finger angewandt.

Für eine optisch ansprechende Lösung für verschiedene Handgrößen wird zusätzlich ein Offset errechnet, um welchen die Hand verschoben werden muss, um die Lenkerstange anatomisch korrekt zu greifen. Dies ist wichtig für Hände, die größer oder kleiner als die Helper-Hände sind. Für den Offset wird der Abstand zwischen dem Mittelgelenk des Mittelfingers der Avatarhand und der Helper-Hand bestimmt.



(a) Lenkerstange in der Realität    (b) Lenkerstange im Stehen    (c) Lenkerstange im Sitzen

**Abbildung 5.7:** Platzierte Hände an der Lenkerstange

Damit Avatare mit unterschiedlichen Gelenkorientierungen verwendet werden können, wird auch für die Helfer-Hände ein Hand-Rotationsoffset ermittelt und angewandt.

## 5.7 Positionierung und Ausrichtung der Hüft- und Fuß-Effektoren

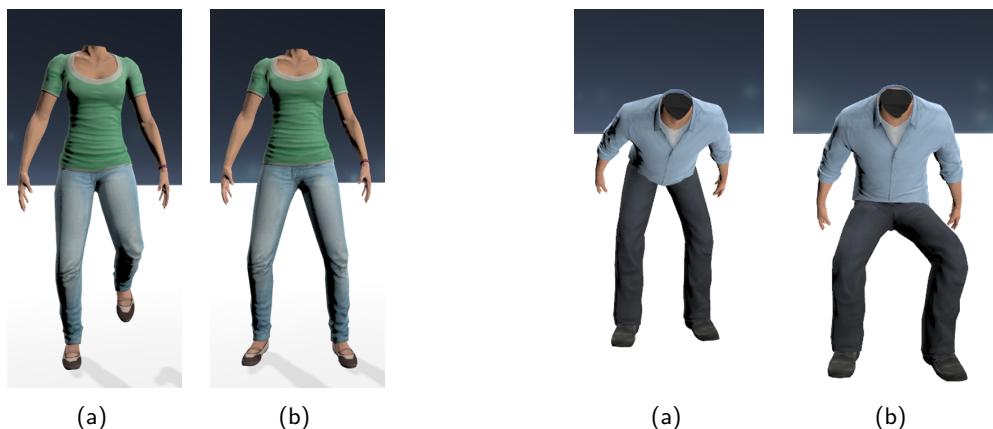
In diesem Abschnitt wird die Verwendung der von der Kinect 2.0 und Azure Kinect empfangenen Skelettdaten beschrieben.

Die Skelettdaten der Kinect werden durch eine vom Fraunhofer HHI entwickelte Applikation ausgelesen und über die ebenfalls vom HHI entwickelte Middleware an die Unity Applikation gesendet. Dort werden sie empfangen und können von dem Objekt *SensorManager* abgefragt werden. Pro Gelenk liefert die Kinect einen Positions- und einen Rotationswert.

Wird keine Kinect verwendet, sind die Füße fest am Boden platziert und die Beine werden bei Bedarf gebeugt. Die Hüfte wird unter dem Kopf platziert, beugt sich der/die Nutzer\*in nach vorne, werden daher die Beine gebeugt. Die Verwendung der Kinect-Daten erlaubt die Positionierung der Gelenke von der Hüfte und den Beinen gemäß dem/der Benutzenden. Ein Vergleich des Avatars mit und ohne Kinect-Tracking ist in Abbildung 5.8 dargestellt.

Für die Positionierung der Beine mithilfe der Kinect-Daten gibt es zwei verschiedene Ansätze. Die Beine können durch die IK-Lösung positioniert werden, wenn die Füße oder Zehen und optional die Hüfte als Effektoren gegeben werden. Die zweite Möglichkeit besteht darin, den Oberkörper durch die IK-Lösung zu positionieren und die Beine eigenständig zu setzen.

Beide Ansätze wurden implementiert und verglichen. Das Ergebnis ist sehr ähnlich. Allerdings hat die erste Lösung den Vorteil, dass schneller und flüssiger zwischen der Positionierung der Beine mit den Kinect-Daten und der Positionierung durch die IK-Lösung umgeschaltet werden kann, wenn es zu Trackingproblemen der Kinect kommt. Zusätzlich können die Effektoren gewichtet werden. Dies ist insbesondere für den Hüft-Effektor nützlich. Durch die Latenz der Kinect ist



**Abbildung 5.8:** Vergleich des Avatars mit (a) und ohne Kinect-Tracking (b)

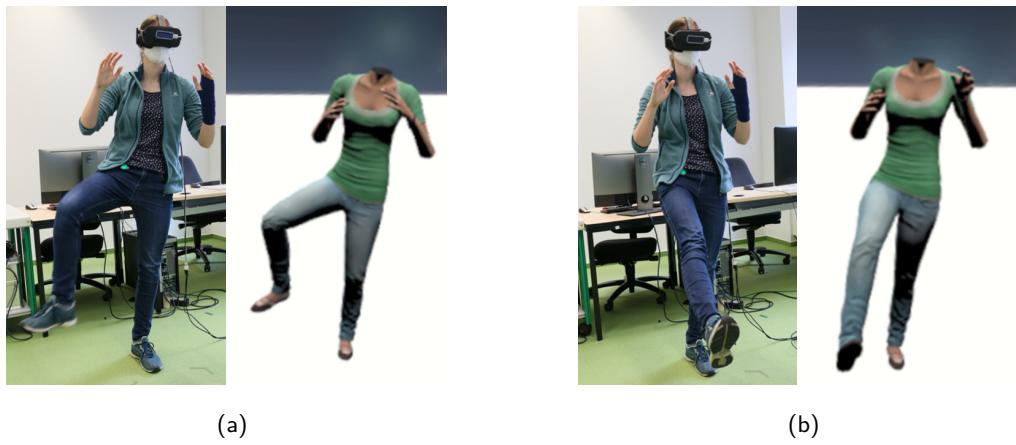
die Position der Hüfte zum Beispiel beim Laufen immer ein wenig hinter beziehungsweise vor der echten Position. Dies führt zu der Positionierung eines schiefen Oberkörpers. Wird der Hüft-Effektor aber nur mit 50% gewichtet, kann die Latenz verringert werden.

Für die Positionierung der Fuß-Effektoren wird statt der Verwendung der Kinect-Positionsdaten die Position mittels der Kinect-Rotationsdaten der Gelenke berechnet. Auf alle Gelenke in der Gelenkkette von der Hüfte zum Fuß werden die jeweiligen Kinect-Rotationswerte angewandt. Diese Lösung hat den Vorteil, dass die Beinbeugung unabhängig von der Beinlänge gleich bleibt. Hat zum Beispiel der Avatar längere Beine als der/die Benutzende, würde die Verwendung der Fußposition dazu führen, dass die Beine gebeugt sind, auch wenn der/die Nutzer\*in die Beine gestreckt hat. Berechnet man die Position vom Fuß mithilfe der Rotationswerte, bleibt die Beinbeugung korrekt und die Fußposition ändert sich je nach Beinlänge.

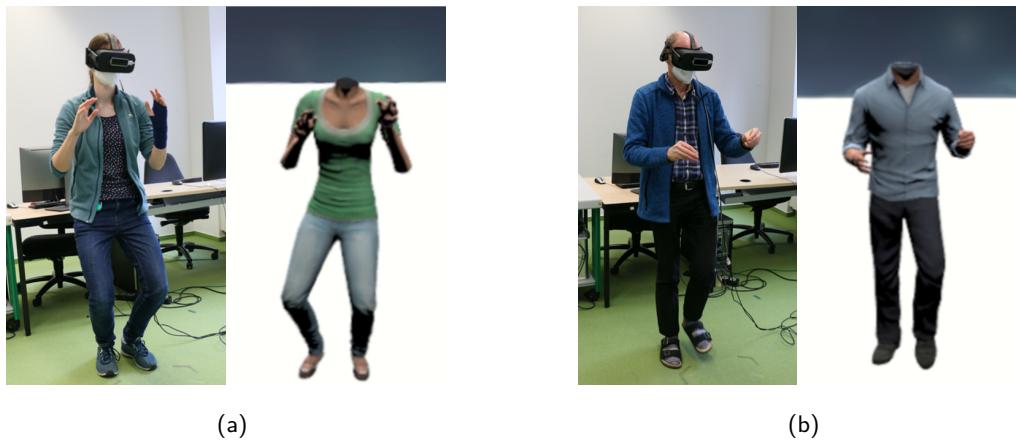
Zusätzlich wird geprüft, ob bei der errechneten Fußposition der Fuß im Boden versinkt. Dies kann durch fehlerhaftes Tracking vorkommen. Anhand des Abstands von den Zehen zum Boden wird errechnet, ob dies der Fall ist. Ist dieser zu klein, wird der Fuß nach oben verschoben. Durch die Benutzung der Füße als Effektoren werden die Zehen von der IK-Lösung nicht mehr rotiert. Es wurde eine eigene Lösung entwickelt, um die Zehen zu beugen. Allerdings stellte der Entwickler von Final IK später eine Lösung bereit. Da diese einfacher ist und sich besser in die Struktur von Final IK einfügt, wurde diese verwendet.

Je nach Datenverfügbarkeit werden die Beine mit den Kinect-Daten oder rein durch die IK-Lösung positioniert. Werden mehrere Skelette durch die Kinect erfasst, werden die Daten des Skeletts verwendet, bei dem der Kopf die geringste Distanz zur Headsetposition aufweist.

In Abbildung 5.9 und 5.10 sind der Avatar und der/die Nutzer\*in im Vergleich zum Avatar bei verschiedenen Posen dargestellt.



**Abbildung 5.9:** Avatar mit Kinect-Tracking im Vergleich zum/zur Nutzer\*in (1)



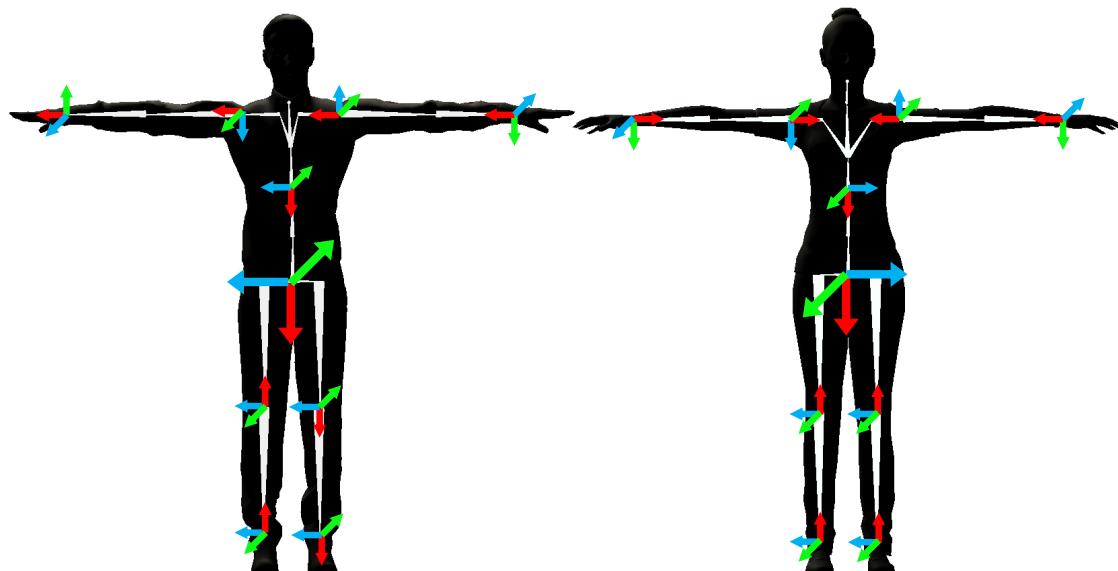
**Abbildung 5.10:** Avatar mit Kinect-Tracking im Vergleich zum/zur Nutzer\*in (2)

#### Anwendung der Rotationsdaten auf unterschiedliche Avatare

Unterschiedliche Avatare können verschiedene Gelenkorientierungen haben. Die Gelenk-Rotationsdaten müssen daher entsprechend umgerechnet werden, um auf beliebige Avatare angewandt werden zu können. Ein Beispiel für die unterschiedlichen Orientierungen der Gelenke ist in Abbildung 5.11 abgebildet.

Für die Anwendung der Rotationsdaten auf einen beliebigen Avatar muss der Rotationsoffset ermittelt werden, welcher die Kinect-Rotationsdaten so dreht, dass diese auf Gelenke mit einer anderen Orientierung anwendbar sind. Am Beispiel von Abbildung 5.11 müsste beispielsweise der Hüftrotationswert des linken Avatars 180 Grad um die X-Achse rotiert werden, um ihn auf den rechten Avatar anzuwenden. Zur Bestimmung der Rotationoffsets werden die lokalen Rotationswerte der Avatare in der T-Pose benutzt.

Die Kinect-Gelenkorientierungen werden aus einer Datei eingelesen. Die Gelenkorientierungen



**Abbildung 5.11:** Unterschiedliche Gelenkorientierungen bei Avataren

unterscheiden sich bei der Kinect 2.0 und der Azure Kinect, daher gibt es je eine eigene Datei. Für die Ermittlung der Azure Kinect-Gelenkorientierungen wurde ein Avatar verwendet, der dieselben Orientierungen hat. Die Gelenkorientierungen der Kinect 2.0 wurden manuell in eine Datei geschrieben, da kein Avatar vorhanden war, der dieselben Gelenkorientierungen wie die der Kinect 2.0-Daten hat.

Mit den Gelenkorientierungen des Avatars und der Kinect können anschließend die Rotationsoffsets ermittelt werden. Dieser Vorgang ist in Quellcode 5.6 am Beispiel des Hüft- und des linken Oberschenkel-Rotationsoffsets dargestellt.

Die Offsets werden dann mit den jeweiligen Kinect-Rotationswerten multipliziert, sodass diese auf den gegebenen Avatar angewandt werden können.

```
offsetHip = Quaternion.Inverse(kinectJointRots["Hips"]) *
    → skeletonData["Hips"].rotation;
offsetLegL = Quaternion.Inverse(kinectJointRots["Hips"]) *
    → kinectJointRots["LeftUpperLeg"]) * (skeletonData["Hips"].rotation *
    → skeletonData["LeftUpperLeg"].rotation);
```

**Quellcode 5.6:** Beispiel der Rotationoffsetberechnung für die Hüfte und den linken Oberschenkel

### Probleme

Ursprünglich sollte nur die Azure Kinect verwendet werden. Diese sollte als Ersatz für die nicht mehr im Handel erhältliche Kinect 2.0 dienen, welche im aktuellen Projekt eingesetzt wird. Allerdings fiel bei ihrer Benutzung auf, dass sie eine deutlich höhere Latenz als die Kinect 2.0 hat. Dies fällt beispielsweise beim Laufen auf der Stelle besonders auf. Verwendet man die Daten der Azure Kinect für die Positionierung der Fuß-Effektoren, sind die Fußpositionen so verzögert, dass sich das jeweils andere Bein beim Laufen hebt. Die Verzögerung der Daten fällt auch stark auf, wenn man die durch einen Controller gegebene Handposition mit der der Azure Kinect bei Bewegungen vergleicht. Dies wurde auch durch Messungen der Distanz zwischen der Controller-Handposition zur Azure Kinect-Handposition bei Bewegungen ersichtlich. Anhand von Umkehrpunkten bei der Handbewegung wurde eine Latenz von 200 ms ermittelt. Der Glättungsfaktor hat dabei Einfluss auf die Latenz, je höher der Glättungsfaktor, desto höher ist die Latenz. Die Messdaten sind im Anhang B zu finden.

Die Kinect 2.0 hat eine geringere Latenz, daher wurde zu dieser gewechselt. Sie wird momentan bei den klinischen Partnern eingesetzt, sodass die Lösung dort verwendet werden kann.

Ein weiteres Problem ist, dass die Kinect-Daten zum Teil zittern, beispielsweise zittert die Fuß-Position. Dies könnte man durch ein Filtern oder Glätten der Daten ändern, allerdings hätte dies Auswirkungen auf die Latenz. Da es wichtig ist, dass die Bewegungen des/der Nutzers\*in mit möglichst geringer Verzögerung beim Avatar sichtbar sind, werden die Daten nicht geglättet beziehungsweise gefiltert.

## 5.8 Kalibrierung der Kinect

Um die Kinect-Daten in Unity zu benutzen, müssen diese in das von Unity benutzte Koordinatensystem transformiert werden. Der Koordinatenursprung der Kinect-Daten liegt im Tiefensor, der von Unity an einem speziellen Punkt, der bei der Kalibrierung der Oculus HMDs festgelegt wird.

### Vorgehensweise

Für die Kalibrierung gab es bereits eine früher am HHI entwickelte Lösung für die Kinect 2.0 in Verbindung mit einem Asset, welches die Kinect-Daten direkt in Unity abfragt. Auf dieser baut die implementierte Kalibrierung auf. Sie wurde insbesondere für die Verwendung der Azure Kinect sowie die Nutzung der Kinect-Daten von der Middleware angepasst.

Für die Transformation der Kinect-Daten in das Koordinatensystem von Unity wird eine Transformationsmatrix benötigt. Um diese ermitteln zu können, werden Koordinaten der gleichen Position im Koordinatensystem der Kinect und dem von Unity gesammelt. Dazu wurde die Handgelenkposition verwendet, da diese sowohl bei den Kinect-Daten als auch bei den Controller-Daten vorliegt. Von den Controllern wird die lokale Position zum Elternelement des Avatares verwendet. Dies ermöglicht, dass eine Transformation des *GameObjects* bei der Kalibrierung nicht berücksichtigt wird.

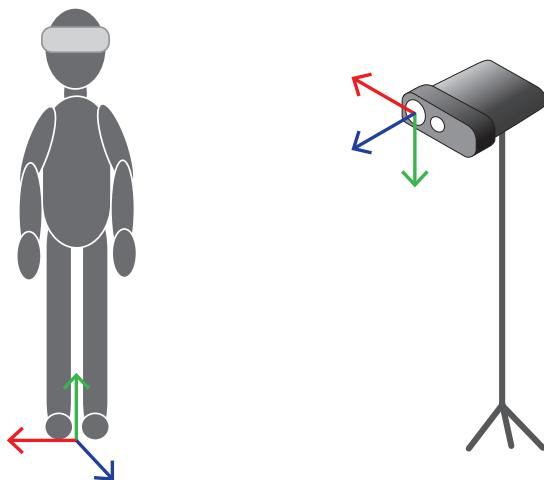
Für die Kalibrierung kann der/die Nutzer\*in durch Knopfdruck auf dem Controller Daten sammeln. Um möglichst genaue und unterschiedliche Werte zu erhalten, sollten folgende Kriterien erfüllt werden:

- Entfernung zum vorherigen Wert > 10 cm
- keine Bewegung des Handgelenks bei und kurz vor der Aufnahme

Treffen beide Kriterien zu, werden die Daten gespeichert. Wenn die Hand bewegt wird, sollen keine Daten aufgezeichnet werden, da durch die Latenz der Kinect deren Handposition verzögert ist. Die Anzahl der zu sammelnden Datenpunkte, der Mindestabstand und die Zeit, in der sich das Handgelenk nicht bewegen soll, können in einer Konfigurationsdatei festgelegt werden.

Der/die Nutzer\*in muss die angegebene Anzahl von Datenpunkten jeweils mit der rechten und linken Hand sammeln. Über den Fortschritt informiert eine GUI mit zwei Fortschrittsbalken. Sind ausreichend Werte gesammelt, werden diese an die Middleware geschickt. Dort wird eine Transformationsmatrix errechnet, welche an die Unity-Anwendung geschickt wird. In Unity wird diese Nachricht empfangen und eingelesen. Aus der Transformationsmatrix wird eine Quaternion errechnet, welche den Rotationsunterschied zwischen der Kinect und dem Unity-Ursprung darstellt.

Für die Anwendung der Kinect-Daten werden die Positions- und Rotationsdaten der Kinect mithilfe der Transformationsmatrix beziehungsweise der Quaternion für die Rotation umgerechnet. Es ergeben sich die lokalen Positions- und Rotationswerte. Die Rechnung mit den lokalen Werten hat den Vorteil, dass der Avatar in der virtuellen Welt gedreht und bewegt werden kann und sich die Kinect-Daten dabei dementsprechend mitbewegen. Diese werden dann für



**Abbildung 5.12:** Koordinatensysteme von Unity (links) und der Azure Kinect (rechts)

die einfache Benutzung in Weltkoordinaten umgerechnet.

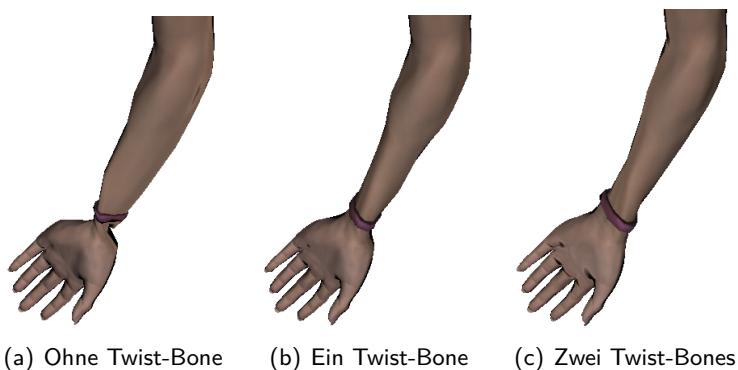
Die ermittelte Transformationsmatrix wird in eine JSON-Datei geschrieben, um die Kalibrierungsdaten bei einem erneuten Start der Applikation wiederverwenden zu können. So kann sie fortlaufend verwendet werden. Über die Kalibrierungs-Konfigurationsdatei kann angegeben werden, ob eine neue Kalibrierung durchgeführt werden soll oder, wenn vorhanden, eine alte Kalibrierung angewandt werden soll. Zusätzlich kann definiert werden, welche Kinect verwendet wird. Für die Azure Kinect und die Kinect 2.0 werden unterschiedliche Kalibrierungsdateien angelegt.

## 5.9 Optimierung der Avatare

Für diese Arbeit wurden die Rocketbox-Avatare [59] verwendet. Für die optimale Verwendung müssen diese noch angepasst werden, was in Blender erfolgte.

Der Kopf muss als separates Mesh vorliegen, sodass man dieses deaktivieren kann. Dies ist für die IK-Lösung nötig, da es sonst in seltenen Fällen dazu kommen kann, dass man das Innere des Kopfes sieht. Für die Integration in VReha muss außerdem jede Hand ein eigenes Material haben. So kann man es beim Greifen beispielsweise einfärben, um dem/der Nutzer\*in Feedback zu geben, ob etwas greifbar ist.

Zusätzlich müssen Twist-Bones<sup>9</sup> im Unterarm hinzugefügt werden. Ohne diese wird das Mesh vom Avatar verdreht, wenn man seine Hand rotiert, wie in Abbildung 7.1(a) zu sehen ist. Dies liegt daran, dass sich das Ellenbogen-Gelenk dabei nicht mitdreht. Stattdessen liegt der Drehpunkt mittig im Unterarm. Um das Problem zu beheben, wurden verschiedene Ansätze getestet. Es lässt sich durch das Hinzufügen zweier Twist-Bones am besten beheben. Ein Vergleich zwischen einem Arm ohne Twist-Bone und Armen mit Twist-Bones ist in Abbildung 5.13 dargestellt. Mit dem Skript *TwistRelaxer* von Final IK werden die Twist-Bones so rotiert, dass die Verdrehung des Meshs ausgeglichen wird.



**Abbildung 5.13:** Einsatz von Twist-Bones

<sup>9</sup>**Twist-Bones** sind Hilfsknochen im Avatar-Skelett, die Verdrehungen im Mesh ausgleichen können.

## 6 Nutzerstudie

Es wurde eine Nutzerstudie durchgeführt, um herauszufinden, ob die Selbstrepräsentation durch einen Avatar in VReha den gewünschten Effekt eines höheren Präsenz- und Verkörperungsgefühls erzielt sowie von Nutzer\*innen gegenüber einer Repräsentation durch Hand-Modelle bevorzugt wird.

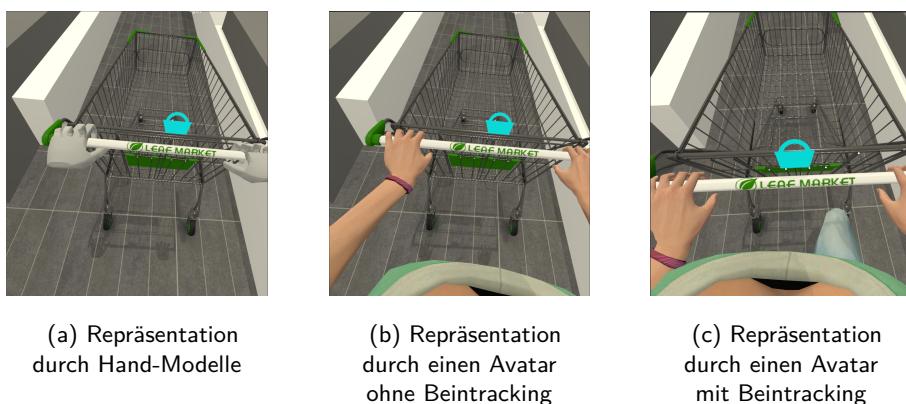
Dazu wurde der Fragebogen von Gonzalez-Franco u. a. [61] zum Gefühl der Verkörperung bei der Repräsentation durch einen Avatar beziehungsweise durch Hand-Modelle verwendet. Es wurden nur die Fragen aus den Kategorien Körperbesitz, motorische Steuerung, Position und Aussehen verwendet, da die anderen Kategorien (Reaktion auf externe Stimulation und Tastempfindung) für diesen Versuch nicht relevant sind. Die Fragen wurden in Bezug auf die Hände und auf den Körper bei Repräsentation durch einen Ganzkörperavatar gestellt.

Zusätzlich wurden Fragen in Bezug auf die physische Präsenz verwendet. Dazu wurden die aus dem Englischen übersetzten Fragen der Multimodal Presence Scale [62] verwendet. Der Fragebogen ist im Anhang C.1 zu finden.

Es wurden drei Versuche pro Person durchgeführt. Für die Versuche wurde die imVST-Anwendung gewählt. Die Teilnehmer\*innen mussten fünf Minuten durch den Supermarkt laufen und nach Sachen greifen. Dabei wurden sie repräsentiert durch

- Hand-Modelle,
- einen Avatar, bei dem auf die Beine keine Tracking-Daten angewandt wurden,
- einen Avatar, bei dem die Beine mittels Kinect-Tracking bewegt wurden.

Dies ist in Abbildung 6.1 dargestellt. Bei den Hand-Modellen handelt es sich um die von der Leap Motion bereitgestellten Standardmodelle, diese werden derzeit in VReha verwendet. Der Avatar wurde gemäß dem/der Benutzenden skaliert und hinsichtlich des Geschlechts gewählt. Die Versuche wurden in zufälliger Reihenfolge durchgeführt.



**Abbildung 6.1:** Darstellung der im Rahmen der Nutzerstudie durchgeföhrten Versuche

In VR kann es zu Motion Sickness<sup>10</sup> kommen. Zwischen den Versuchen wurden deshalb ausreichend lange Pausen eingelegt, um das Erlebnis in VR nicht durch Motion Sickness aus dem vorherigen Versuch zu beeinflussen. Zusätzlich wurde nach jedem Versuch die Motion Sickness des/der Teilnehmers\*in anhand der Fast Motion Sickness Scale [64] erfasst.

Insgesamt haben 11 Personen an der Nutzerstudie teilgenommen. Davon wurden die Daten einer Person verworfen, da es zu Fehlern bei der Steuerung für die Fortbewegung kam.

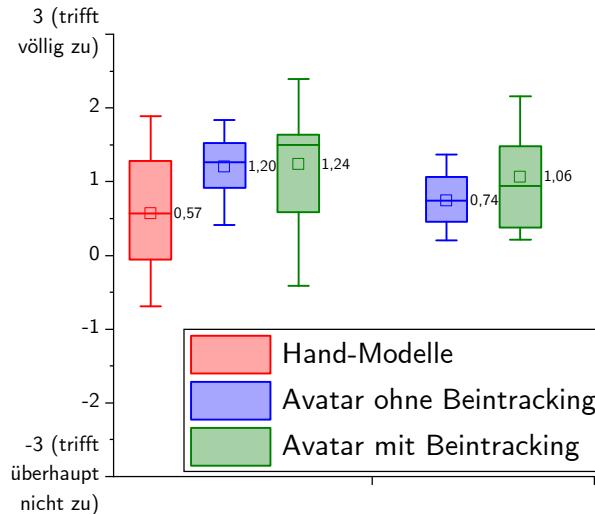
40 % der Teilnehmer\*innen waren weiblich, 60 % der Teilnehmer\*innen männlich. 60 % der Teilnehmenden nutzen VR-Anwendungen selten bis nie, 40 % verwenden sie häufig. Das Durchschnittsalter betrug 37 Jahre. Die gewonnenen Daten wurden mittels den vorgestellten Formeln (siehe Anhang C.2.1) von Gonzalez-Franco u. a. [61] in Bezug auf das Verkörperungsgefühl und Volkmann u. a. [62] in Bezug auf die physische Präsenz ausgewertet. Die Daten wurden im Rahmen einer kleinen Stichprobe erhoben, daher müssen sie statistisch vorsichtig betrachtet werden.

Um zu ermitteln, ob signifikante Unterschiede zwischen den erhobenen Daten der Versuche bestehen, wurden gepaarte t-Tests<sup>11</sup> durchgeführt. Dazu wurde die Software *jamovi* [66] verwendet. Es wurde der *Robuste Paired Samples T-Test* gewählt, da die für andere Tests nötigen Voraussetzungen, wie eine Normalverteilung oder eine Homogenität der Varianz, für die Daten nicht zutreffen. Es wurde der gepaarte t-Test verwendet, weil abhängige Stichproben vorliegen, da jede Person an allen drei Versuchen teilnahm. Die Ergebnisse der t-Tests sind im Anhang C.2.3 dargestellt. Vergleicht man die Repräsentation durch Hand-Modelle mit der durch einen Avatar, besteht ein signifikanter Unterschied für das Verkörperungsgefühl in Bezug auf die Hände. Für die physische Präsenz und den Körperbesitz zeichnet sich eine Tendenz ab, dass diese bei Repräsentation durch einen Avatar stärker ist. Vergleicht man die Repräsentation durch einen Avatar mit Beintracking mit einer ohne Beintracking, zeigt sich eine Tendenz für ein verstärktes Gefühl der Verkörperung und des Körperbesitzes in Bezug auf den gesamten Körper bei dem Avatar mit Beintracking. Signifikante Unterschiede liegen allerdings nicht vor. Die Ergebnisse werden im Folgenden anhand von Box-Plots näher betrachtet.

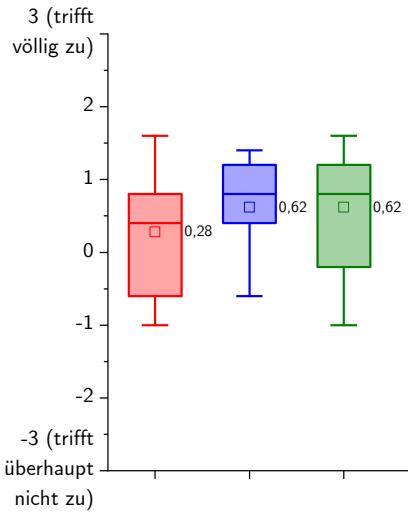
In Abbildung 6.2 ist links das Verkörperungsgefühl in Bezug auf die Hände dargestellt. Es ist ersichtlich, dass es in Bezug auf die Hände bei der Repräsentation durch Hand-Modelle am geringsten ist. Das Verkörperungsgefühl in Bezug auf die Hände setzt sich aus Körperbesitz, motorischer Steuerung und der korrekten Positionierung der Hände in VR zusammen. Im Anhang C.1 ist dargestellt, aus welchen Fragen sich die einzelnen Faktoren zusammensetzen.

<sup>10</sup> Stimmt die Wahrnehmung von Bewegungen mit der tatsächlichen Bewegung nicht überein, kann es zur **Motion Sickness** kommen. Dies kann in VR beispielsweise dann auftreten, wenn der/die Nutzer\*in in der Realität auf der Stelle läuft, sich in der virtuellen Welt aber vorwärtsbewegt. Es kann dabei zu Schwindel und Übelkeit kommen. [63]

<sup>11</sup> Mit einem **t-Test** kann geprüft werden, ob sich zwei Stichproben signifikant voneinander unterscheiden. Es wird überprüft, ob die Nullhypothese zutrifft, in diesem Fall, dass die Stichproben gleich sind. Als Ergebnis kann man den p-Wert (Signifikanzwert) erhalten, liegt dieser unter dem festgelegten Signifikanzniveau, meist 0,05, besteht ein signifikanter Unterschied zwischen den Stichproben und die Nullhypothese wird abgelehnt. [65] Der gepaarter t-Test wird bei abhängigen Stichproben verwendet.



**Abbildung 6.2:** Verkörperungsgefühl in Bezug auf die Hände und den Körper



**Abbildung 6.3:** Physisches Präsenzgefühl

Die Ergebnisse der einzelnen Faktoren sind in Abbildung C.1 dargestellt. Spannend ist, dass die Ergebnisse für die motorische Steuerung und die korrekte Positionierung tendenziell positiver ausfallen, wenn eine Repräsentation durch einen Avatar erfolgt, obwohl die Hände auf dieselbe Weise getrackt und positioniert werden.

Das Verkörperungsgefühl in Bezug auf den ganzen Körper ist bei einer Repräsentation durch einen Avatar mit Beintracking ein wenig besser, dies ist in Abbildung 6.2 rechts dargestellt. Das Verkörperungsgefühl in Bezug auf den ganzen Körper setzt sich aus denselben Faktoren zusammen wie das Verkörperungsgefühl der Hände, allerdings bezogen auf den Körper zuzüglich dem Aussehen. Die Ergebnisse der einzelnen Faktoren sind in Abbildung C.2 abgebildet. Insbesondere die motorische Steuerung wurde bei einem Avatar mit Beintracking tendenziell besser bewertet. Allerdings gibt es hier keine signifikanten Unterschiede, wie oben beschrieben. Die Repräsentation durch einen Avatar mit Beintracking wurde aber als beste Variante von den Teilnehmer\*innen im abschließenden Fragebogen ausgewählt. Auf diesen wird später näher eingegangen. Die Fragen zum Verkörperungsgefühl wurden in Bezug auf den ganzen Körper und nicht in Bezug auf die Beine gestellt. Dies könnte erklären, warum keine signifikanten Unterschiede im Verkörperungsgefühl bestehen, die Teilnehmer\*innen aber dennoch die Variante mit Beintracking klar bevorzugen.

In Abbildung 6.3 sind die Ergebnisse zur physischen Präsenz dargestellt. Es herrscht ein tendenziell höheres Präsenzgefühl bei einer Darstellung durch einen Avatar. Das Präsenzgefühl bei einer Darstellung durch einen Avatar mit beziehungsweise ohne Beintracking ist ähnlich.

Zusammenfassend hat sich gezeigt, dass durch die Repräsentation durch einen Avatar ein höheres Verkörperungsgefühl entsteht. Dies bestätigte sich auch durch die am Ende der Versuchsreihe erfolgte Befragung. Die Teilnehmenden wurden gebeten, die Variante zu wählen, welche ihnen

am besten zusagte und diese Wahl zu begründen. 90 % der Teilnehmenden wählten die Variante „Avatar mit Beintracking“. Nur eine Person wählte die Variante „Avatar ohne Beintracking“. Allerdings begründete diese es damit, dass das Greifen bei dem Avatar ohne Beintracking am besten funktionierte. Der/die Teilnehmende fügte jedoch hinzu, dass sich der Avatar mit Beintracking ihm/ihr ähnlicher anfühlte.

Eine teilnehmende Person äußerte: „Mir ist weniger schlecht geworden, weil ich mich selbst sehen konnte und meine Bewegungen mit denen des Avatars übereinstimmten. Dadurch fiel es mir auch leichter, auf der Stelle zu laufen und nicht nach vorne oder zur Seite zu laufen.“ Dies ist ein weiterer positiver Effekt des Avatars.

Ein/eine anderer Teilnehmer\*in beschrieb, dass sich durch die bessere Bewegung des Avatars mit Beintracking der Einkauf realistischer/immersiver anfühlte.

Die Darstellung der Beine mittels Beintracking wurde als besser empfunden: „Die Beine wurden beim Beugen des Oberkörpers und bei anderen Beinbewegungen besser positioniert bzw. in VR dargestellt.“

Die Variante der Darstellung durch Hand-Modelle wurde als „unreal – wie zwei schwebende Handschuhe“ empfunden. Ein/eine Teilnehmer\*in beschrieb diese Variante folgendermaßen: „Bei fehlender Körperrepräsentation hat man das Gefühl,körperlos zu sein, zu schweben und etwas zu hoch zu sein, nicht passend zur eigenen Position“.

Die Begründungen aller Teilnehmer\*innen sind im Anhang C.3.4 zusammengestellt.

Insgesamt stellt die Repräsentation durch einen Avatar eine deutliche Verbesserung zu der Repräsentation durch Hand-Modelle dar. Sie führt zu einem ausgeprägteren Verkörperungsgefühl. Auch für das physische Präsenzgefühl zeigt sich eine positive Tendenz. Zusätzlich wurde ersichtlich, dass die Teilnehmenden die Repräsentation durch einen Avatar mit Beintracking klar bevorzugen.

Ein Einfluss der verschiedenen Repräsentationen auf die Motion Sickness konnte nicht festgestellt werden.

Die konkret verwendeten Fragen und die Antworten der Teilnehmer\*innen sind im Anhang C.1 und C.3 zu finden. Deskriptive Statistiken zu den Antworten sind im Anhang C.2.4 zusammengestellt.

# 7 Zusammenfassung und Ausblick

## Zusammenfassung

In dieser Arbeit wurde eine inverse Kinematik-Lösung für eine Selbstrepräsentation durch einen Ganzkörperavatar in VR am Beispiel des Forschungsprojekts VReha erfolgreich umgesetzt.

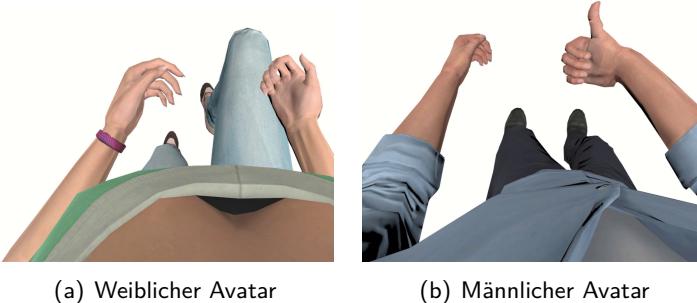
Die Evaluation von vier existierenden IK-Lösungen zeigte, dass VRIK die Arm-Gelenke am genauesten positioniert und am besten für die Umsetzung einer Avatar-Posenschätzung geeignet ist. VRIK wurde daher in das Forschungsprojekt VReha integriert.

Zusätzlich wurde eine Skalierung des Avatars passend zur Größe des/der Benutzenden entwickelt. Hierbei werden die Körpergröße, die Armlänge und die Hüfthöhe skaliert und die Skalierungsdaten zur Wiederverwendung gespeichert. Die Nutzung verschiedener Sensorsdaten und die Adaption an deren Verfügbarkeit wurde umgesetzt. Es werden die Oculus Rift (Kopftracking), die Oculus Quest (Hand- und Kopftracking), die Leap Motion (Handtracking), die Kinect 2.0 und die Azure Kinect (jeweils Ganzkörpertracking) unterstützt. Ursprünglich sollte nur die Azure Kinect verwendet werden. Allerdings fiel bei ihrer Benutzung eine hohe Latenz auf, daher wurde zur Kinect 2.0 gewechselt. Weiterhin wurden die spezifischen Anforderungen von VReha umgesetzt: Der Avatar kann eine Sitzpose einnehmen und die Hände können an einer Lenkerstange einrasten.

Die Lösung unterstützt Avatare mit verschiedenen Gelenkorientierungen und neue Avatare können einfach hinzugefügt werden. Über eine Konfigurationsdatei sind die Parameter der Anwendung anpassbar, zum Beispiel ob Kinect-Daten verwendet werden sollen.

Die Forschungsergebnisse zu den Auswirkungen der Selbstrepräsentation durch einen Avatar auf die nutzende Person wurden berücksichtigt. Es wurde auf die korrekte Skalierung der Avatare geachtet und möglichst durchschnittliche, neutrale Avatare gewählt.

Eine abschließende Nutzerstudie zeigte, dass der erwartete Effekt eines höheren Verkörperungsgefühls durch die Repräsentation mit einem Avatar eintritt. Es zeigte sich zudem eine Tendenz für ein höheres physisches Präsenzgefühl in der virtuellen Welt. Aus der Nutzerstudie ging hervor, dass Nutzer\*innen die Repräsentation durch einen Avatar mit Beintracking gegenüber einer Repräsentation durch Hand-Modelle oder einen Avatar ohne Beintracking klar bevorzugen. In Abbildung 7.1 sind Beispiele für die Selbstrepräsentation durch einen Avatar mit Kinect-Tracking dargestellt.



**Abbildung 7.1:** Beispiele zur Selbstdarstellung durch einen Avatar mit Kinect-Tracking

Zusammenfassend wurde die Zielstellung, eine Avatar-Posenschätzungslosung mittels inverser Kinematik am Beispiel des Forschungsprojektes VReha umzusetzen, erfolgreich realisiert. Es wurden alle Anforderungen, welche in Kapitel 3.2 spezifiziert wurden, erfolgreich umgesetzt.

## Ausblick

In Zukunft sollen zusätzlich diverse Avatare eingesetzt werden, beispielsweise mit unterschiedlicher Hautfarbe, Kleidung und Alter. Momentan werden nur ein männlicher und ein weiblicher hellhäutiger Avatar verwendet. Das Feedback der Anwender\*innen soll bei der Weiterentwicklung eingearbeitet werden. Außerdem wäre die Verwendung einer anderen Hardware als die Kinect 2.0 zum Tracken der Beine denkbar. Die Daten zittern zum Teil und ein gutes Tracking ist nur bei frontal stehenden Personen möglich. Außerdem kann es zu Latenzen kommen. Momentan wird am HHI an einer eigenen Lösung gearbeitet, die diese Probleme beheben soll. Denkbar wäre auch, nach einer Lösung zu suchen, die die Daten so filtert, dass das Zittern der Daten abgeschwächt wird, dabei aber die Daten nicht merkbar verzögert werden. Beispielsweise könnte man die Daten nur filtern, wenn die Füße auf dem Boden stehen. So würden die Positionsdaten von auf dem Boden stehenden, ruhigen Füßen gefiltert werden und von Füßen, die sich über dem Boden bewegen, nicht. Das Problem, dass die Skelettdaten der Kinect 2.0 nur für frontal zur Kinect 2.0 stehende Personen korrekt sind, könnte man teilweise beheben: Steht die Person beispielsweise mit dem Rücken zur Kinect 2.0, werden die Daten gespiegelt, das linke Bein entspricht dann dem Rechten. Dies könnte man anhand der Drehung der Person erkennen und die Daten entsprechend spiegeln.

Die Lösung kann auch in zukünftigen Projekten und Anwendungen Verwendung finden. Sie ist ohne Weiteres in andere Projekte integrierbar, nur einige Sonderfunktionen und die GUI sind speziell auf VReha ausgelegt und müssten entsprechend angepasst werden.

# Literaturverzeichnis

- [1] M. Slater, »Place Illusion and Plausibility Can Lead to Realistic Behaviour in Immersive Virtual Environments,« *Philosophical Transactions of the Royal Society B: Biological Sciences*, Jg. 364, Nr. 1535, S. 3549–3557, 12. Dez. 2009. DOI: 10.1098/rstb.2009.0138.
- [2] M. Gonzalez-Franco, P. Abtahi und A. Steed, »Individual Differences in Embodied Distance Estimation in Virtual Reality,« in *2019 IEEE Conference on Virtual Reality and 3D User Interfaces (VR)*, Osaka, Japan: IEEE, März 2019, S. 941–943, ISBN: 978-1-72811-377-7. DOI: 10.1109/VR.2019.8798348.
- [3] A. Steed, Y. Pan, F. Zisch und W. Steptoe, »The impact of a self-avatar on cognitive load in immersive virtual reality,« in *2016 IEEE Virtual Reality (VR)*, Greenville, SC, USA: IEEE, März 2016, S. 67–76, ISBN: 978-1-5090-0836-0. DOI: 10.1109/VR.2016.7504689.
- [4] N. Ogawa, T. Narumi, H. Kuzuoka und M. Hirose, »Do You Feel Like Passing Through Walls?: Effect of Self-Avatar Appearance on Facilitating Realistic Behavior in Virtual Environments,« in *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*, Honolulu HI USA: ACM, 21. Apr. 2020, S. 1–14, ISBN: 978-1-4503-6708-0. DOI: 10.1145/3313831.3376562.
- [5] S. Venator. »Therapie mit virtuellen Welten - W wie Wissen - ARD | Das Erste,« Adresse: <https://www.daserste.de/information/wissen-kultur/w-wie-wissen/Therapie-mit-virtuellen-Welten-100.html> (besucht am 01.11.2020).
- [6] Barmer. »Virtuelle Realität in der Medizin - neue Möglichkeiten bei der Behandlung von Krankheiten,« Adresse: <https://www.barmer.de/gesundheit-verstehen/gesellschaft/virtuelle-realitaet-in-der-medizin-106024> (besucht am 01.11.2020).
- [7] A. Haselhoff. »Wie funktioniert eine VR-Brille? Stereoskopisches Sehen und räumliche Darstellung,« VR-Guides. (10. Dez. 2018), Adresse: <https://www.vr-guides.de/vr-brille-funktionsweise/> (besucht am 24.01.2021).
- [8] World of VR GmbH. »Oculus Quest: Kurz & Knapp,« Adresse: <https://worldofvr.net/oculus-quest/> (besucht am 04.12.2020).
- [9] »Oculus Rift: VR-Headset Für VR-Fähige PCs | Oculus,« Adresse: <https://www.oculus.com/rift/> (besucht am 26.02.2021).
- [10] Ultraleap. »Leap Motion Controller Data Sheet,« Adresse: [https://www.ultraleap.com/datasheets/Leap\\_Motion\\_Controller\\_Datasheet.pdf](https://www.ultraleap.com/datasheets/Leap_Motion_Controller_Datasheet.pdf) (besucht am 04.12.2020).
- [11] Y. Bloch. »Kinect v2,« VRNerds. (8. März 2014), Adresse: <https://www.vrnerds.de/kinect-v2/> (besucht am 26.02.2021).

- [12] Microsoft. »Azure Kinect DK,« Adresse: <https://azure.microsoft.com/de-de/services/kinect-dk/> (besucht am 26.10.2020).
- [13] Fraunhofer HHI. »VReha,« Adresse: <https://www.hhi.fraunhofer.de/abteilungen/vit/projekte/vreha.html> (besucht am 26.07.2020).
- [14] MPI für Kognitions- und Neurowissenschaften. »VReha - „Virtuelle Welten Für Digitale Diagnostik Und Kognitive Rehabilitation“.« (28. Mai 2020), Adresse: [https://www.youtube.com/watch?v=4U80\\_05-iFo](https://www.youtube.com/watch?v=4U80_05-iFo) (besucht am 19.03.2021).
- [15] DocCheck Medical Services. »Exekutivfunktion,« DocCheck Flexikon, Adresse: <https://flexikon.doccheck.com/de/Exekutivfunktion> (besucht am 17.03.2021).
- [16] K. Kilteni, R. Grotens und M. Slater, »The Sense of Embodiment in Virtual Reality,« *Presence: Teleoperators and Virtual Environments*, Jg. 21, Nr. 4, S. 373–387, Nov. 2012. DOI: 10.1162/PRES\_a\_00124.
- [17] M. Botvinick und J. Cohen, »Rubber Hands ‘Feel’ Touch That Eyes See,« *Nature*, Jg. 391, Nr. 6669, S. 756–756, 1. Feb. 1998. DOI: 10.1038/35784.
- [18] M. Slater, D. Perez-Marcos, H. H. Ehrsson und M. V. Sanchez-Vives, »Towards a Digital Body: The Virtual Arm Illusion,« *Frontiers in Human Neuroscience*, Jg. 2, 20. Aug. 2008. DOI: 10.3389/neuro.09.006.2008.
- [19] M. Slater, D. Pérez Marcos, H. Ehrsson und M. V. Sanchez-Vives, »Inducing illusory ownership of a virtual body,« *Frontiers in Neuroscience*, Jg. 3, 2009. DOI: 10.3389/neuro.01.029.2009.
- [20] M. Gonzalez-Franco, D. Perez-Marcos, B. Spanlang und M. Slater, »The Contribution of Real-Time Mirror Reflections of Motor Actions on Virtual Body Ownership in an Immersive Virtual Environment,« Ser. *IEEE Virtual Reality*, Apr. 2010, S. 111–114. DOI: 10.1109/VR.2010.5444805.
- [21] K. Kilteni, J.-M. Normand, M. V. Sanchez-Vives und M. Slater, »Extending Body Space in Immersive Virtual Reality: A Very Long Arm Illusion,« *PLOS ONE*, Jg. 7, Nr. 7, e40867, 19. Juli 2012. DOI: 10.1371/journal.pone.0040867.
- [22] B. van der Hoort, A. Guterstam und H. H. Ehrsson, »Being Barbie: The Size of One’s Own Body Determines the Perceived Size of the World,« *PLOS ONE*, Jg. 6, Nr. 5, e20195, 25. Mai 2011. DOI: 10.1371/journal.pone.0020195.
- [23] J.-M. Normand, E. Giannopoulos, B. Spanlang und M. Slater, »Multisensory Stimulation Can Induce an Illusion of Larger Belly Size in Immersive Virtual Reality,« *PLOS ONE*, Jg. 6, Nr. 1, e16128, 19. Jan. 2011. DOI: 10.1371/journal.pone.0016128.
- [24] W. Steptoe, A. Steed und M. Slater, »Human Tails: Ownership and Control of Extended Humanoid Avatars,« *IEEE transactions on visualization and computer graphics*, Jg. 19, S. 583–90, 1. Apr. 2013. DOI: 10.1109/TVCG.2013.32.

- [25] M. Slater und S. Wilbur, »A Framework for Immersive Virtual Environments (FIVE): Speculations on the Role of Presence in Virtual Environments,« *Presence: Teleoperators and Virtual Environments*, Jg. 6, Nr. 6, S. 603–616, Dez. 1997. DOI: 10.1162/pres.1997.6.6.603.
- [26] M. Slater, B. Spanlang und D. Corominas, »Simulating Virtual Environments within Virtual Environments as the Basis for a Psychophysics of Presence,« *ACM Transactions on Graphics*, Jg. 29, Nr. 4, 92:1–92:9, 26. Juli 2010. DOI: 10.1145/1778765.1778829.
- [27] M. Slater, M. Usoh und A. Steed, »Taking Steps: The Influence of a Walking Technique on Presence in Virtual Reality,« *ACM Transactions on Computer-Human Interaction*, Jg. 2, Nr. 3, S. 201–219, 1. Sep. 1995. DOI: 10.1145/210079.210084.
- [28] M. Usoh, K. Arthur, M. C. Whitton u. a., »Walking > Walking-in-Place > Flying, in Virtual Environments,« in *Proceedings of the 26th Annual Conference on Computer Graphics and Interactive Techniques*, Ser. SIGGRAPH '99, USA: ACM Press/Addison-Wesley Publishing Co., 1. Juli 1999, S. 359–364, ISBN: 978-0-201-48560-8. DOI: 10.1145/311535.311589.
- [29] J. N. Bailenson und A. C. Beall, »Transformed Social Interaction: Exploring the Digital Plasticity of Avatars,« in *Avatars at Work and Play*, Ser. Computer Supported Cooperative Work, R. Schroeder und A.-S. Axelsson, Hrsg., Bd. 34, Berlin/Heidelberg: Springer-Verlag, 2006, S. 1–16, ISBN: 978-1-4020-3883-9. DOI: 10.1007/1-4020-3898-4\_1.
- [30] N. Yee und J. Bailenson, »The Proteus Effect: The Effect of Transformed Self-Representation on Behavior,« *Human Communication Research*, Jg. 33, Nr. 3, S. 271–290, Juli 2007. DOI: 10.1111/j.1468-2958.2007.00299.x.
- [31] K. Kilteni, I. Bergstrom und M. Slater, »Drumming in Immersive Virtual Reality: The Body Shapes the Way We Play,« *IEEE Transactions on Visualization and Computer Graphics*, Jg. 19, Nr. 4, S. 597–605, Apr. 2013. DOI: 10.1109/TVCG.2013.29.
- [32] D. Banakou, S. Kishore und M. Slater, »Virtually Being Einstein Results in an Improvement in Cognitive Task Performance and a Decrease in Age Bias,« *Frontiers in Psychology*, Jg. 9, 2018. DOI: 10.3389/fpsyg.2018.00917.
- [33] N. Yee und J. Bailenson, »Walk A Mile in Digital Shoes: The Impact of Embodied Perspective-Taking on The Reduction of Negative Stereotyping in Immersive Virtual Environments,« 2006.
- [34] T. C. Peck, S. Seinfeld, S. M. Aglioti und M. Slater, »Putting yourself in the skin of a black avatar reduces implicit racial bias,« *Consciousness and Cognition*, Jg. 22, Nr. 3, S. 779–787, Sep. 2013. DOI: 10.1016/j.concog.2013.04.016.
- [35] V. Groom, J. N. Bailenson und C. Nass, »The influence of racial embodiment on racial bias in immersive virtual environments,« *Social Influence*, Jg. 4, Nr. 3, S. 231–248, Juli 2009. DOI: 10.1080/15534510802643750.

- [36] R. S. Renner, B. M. Velichkovsky und J. R. Helmert, »The perception of egocentric distances in virtual environments - A review,« *ACM Computing Surveys*, Jg. 46, Nr. 2, S. 1–40, Nov. 2013. DOI: 10.1145/2543581.2543590.
- [37] B. Ries, V. Interrante, M. Kaeding und L. Anderson, »The effect of self-embodiment on distance perception in immersive virtual environments,« in *Proceedings of the 2008 ACM Symposium on Virtual Reality Software and Technology - VRST '08*, Bordeaux, France: ACM Press, 2008, S. 167, ISBN: 978-1-59593-951-7. DOI: 10.1145/1450579.1450614.
- [38] B. J. Mohler, H. H. Bülthoff, W. B. Thompson und S. H. Creem-Regehr, »A Full-Body Avatar Improves Egocentric Distance Judgments in an Immersive Virtual Environment,« in *Proceedings of the 5th Symposium on Applied Perception in Graphics and Visualization*, Ser. APGV '08, New York, NY, USA: Association for Computing Machinery, 9. Aug. 2008, S. 194, ISBN: 978-1-59593-981-4. DOI: 10.1145/1394281.1394323.
- [39] Q. Lin, J. Rieser und B. Bodenheimer, »Affordance Judgments in HMD-Based Virtual Environments: Stepping over a Pole and Stepping off a Ledge,« *ACM Transactions on Applied Perception*, Jg. 12, Nr. 2, S. 1–21, 10. Apr. 2015. DOI: 10.1145/2720020.
- [40] D. Roth, J. Lugrin, J. Büser, G. Bente, A. Fuhrmann und M. E. Latoschik, »A Simplified Inverse Kinematic Approach for Embodied VR Applications,« in *2016 IEEE Virtual Reality (VR)*, März 2016, S. 275–276. DOI: 10.1109/VR.2016.7504760.
- [41] M. Parger, J. H. Mueller, D. Schmalstieg und M. Steinberger, »Human upper-body inverse kinematics for increased embodiment in consumer-grade virtual reality,« in *Proceedings of the 24th ACM Symposium on Virtual Reality Software and Technology*, Tokyo Japan: ACM, 28. Nov. 2018, S. 1–10, ISBN: 978-1-4503-6086-9. DOI: 10.1145/3281505.3281529.
- [42] M. Parger, *Inverse Kinematics for Virtual Reality*, Juli 2018. Adresse: <https://diglib.tugraz.at/download.php?id=5c4a48dc5a282&location=browse>.
- [43] K. He. »Procedural Animation Physics Simulation AI and Virtual Beings.« (10. Sep. 2019), Adresse: <https://blog.deepmotion.com/2019/09/10/procedural-animation-physics-simulation-and-virtual-beings/> (besucht am 30.07.2020).
- [44] Deepmotion. »Virtual Reality Tracking,« Adresse: <https://deepmotion.com/virtual-reality-tracking> (besucht am 02.08.2020).
- [45] P. Lang. »Inverse Kinematics in Dead and Buried,« RootMotion. (28. Juni 2016), Adresse: <http://root-motion.com/2016/06/inverse-kinematics-in-dead-and-buried/> (besucht am 26.07.2020).
- [46] A. Aristidou und J. Lasenby, »FABRIK: A fast, iterative solver for the Inverse Kinematics problem,« *Graphical Models*, Jg. 73, Nr. 5, S. 243–260, Sep. 2011. DOI: 10.1016/j.gmod.2011.05.003.
- [47] P. Scheurenbrand, »Avatar-Posen-Schätzung aus Kopf- und Handpositionen für kollaborative VR-Umgebungen,« 2019. DOI: 10.18419/opus-10730.

- [48] F. Jiang, X. Yang und L. Feng, »Real-time full-body motion reconstruction and recognition for off-the-shelf VR devices,« in *Proceedings of the 15th ACM SIGGRAPH Conference on Virtual-Reality Continuum and Its Applications in Industry - VRCAI '16*, Zhuhai, China: ACM Press, 2016, S. 309–318, ISBN: 978-1-4503-4692-4. DOI: 10.1145/3013971.3013987.
- [49] Stereoarts, *Stereoarts/SAFullBodyIK*, 26. Okt. 2020. Adresse: <https://github.com/Stereoarts/SAFullBodyIK> (besucht am 06.12.2020).
- [50] Unity Technologies. »Unity - Manual: Inverse Kinematics,« Adresse: <https://docs.unity3d.com/Manual/InverseKinematics.html> (besucht am 22.01.2021).
- [51] ——, »Unity - Manual: Constraints,« Adresse: <https://docs.unity3d.com/Manual/Constraints.html> (besucht am 22.01.2021).
- [52] ——, »Two Bone IK | Animation Rigging | 0.2.7-Preview,« Adresse: <https://docs.unity3d.com/Packages/com.unity.animation.rigging@0.2/manual/constraints/TwoBoneIKConstraint.html> (besucht am 22.01.2021).
- [53] Root Motion. »Limb IK,« Adresse: <http://www.root-motion.com/finalikdok/html/page12.html> (besucht am 06.12.2020).
- [54] Carnegie Mellon University. »CMU Graphics Lab Motion Capture Database,« Adresse: <http://mocap.cs.cmu.edu/> (besucht am 27.07.2020).
- [55] cMonkeys. »Huge FBX Mocap Library part 1,« Adresse: <https://assetstore.unity.com/packages/3d/animations/huge-fbx-mocap-library-part-1-19991> (besucht am 27.07.2020).
- [56] ——, »Huge FBX Mocap Library part 2,« Adresse: <https://assetstore.unity.com/packages/3d/animations/huge-fbx-mocap-library-part-2-20282> (besucht am 29.07.2020).
- [57] ——, »Huge FBX Mocap Library part 3,« Adresse: <https://assetstore.unity.com/packages/3d/animations/huge-fbx-mocap-library-part-3-20285> (besucht am 29.07.2020).
- [58] Tony Foster. »How to Make a Body in VR - PART 1.« (16. Dez. 2019), Adresse: <https://www.youtube.com/watch?v=tBYl-aSxUe0> (besucht am 24.01.2021).
- [59] Microsoft-Rocketbox, Microsoft. Adresse: <https://github.com/microsoft/Microsoft-Rocketbox> (besucht am 12.02.2021).
- [60] Root Motion. »VRIK,« Adresse: <http://www.root-motion.com/finalikdok/html/page16.html> (besucht am 23.11.2020).
- [61] M. Gonzalez-Franco und T. C. Peck, »Avatar Embodiment. Towards a Standardized Questionnaire,« *Frontiers in Robotics and AI*, Jg. 5, 2018. DOI: 10.3389/frobt.2018.00074.

- [62] T. Volkmann, D. Wessel, N. Jochems und T. Franke, »German Translation of the Multimodal Presence Scale,« 2018. DOI: 10.18420/MUC2018-MCI-0428. Adresse: <http://dl.gi.de/handle/20.500.12116/16655>.
- [63] World of VR GmbH. »Motion Sickness: Definition und einfache Erklärung,« World of VR. (10. Sep. 2019), Adresse: <https://worldofvr.net/motion-sickness/> (besucht am 21.03.2021).
- [64] B. Keshavarz und H. Hecht, »Validating an Efficient Method to Quantify Motion Sickness,« *Human Factors: The Journal of the Human Factors and Ergonomics Society*, Jg. 53, Nr. 4, S. 415–426, Aug. 2011. DOI: 10.1177/0018720811403736. Adresse: <http://journals.sagepub.com/doi/10.1177/0018720811403736> (besucht am 19.03.2021).
- [65] R. Bender und S. Lange, »Was ist der  $p$ -Wert?« *DMW - Deutsche Medizinische Wochenschrift*, Jg. 132, e15–e16, S 01 2007. DOI: 10.1055/s-2007-959030. Adresse: <http://www.thieme-connect.de/DOI/DOI?10.1055/s-2007-959030> (besucht am 19.03.2021).
- [66] The jamovi project, *Jamovi*, 2021. Adresse: <https://www.jamovi.org>.

# Bildquellenverzeichnis

- [1] M. Paget, *Oculus Quest*. Adresse: <https://gamespot1.cbsistatic.com/uploads/original/1551/15511094/3655007-oculus-quest.jpg> (besucht am 26.10.2020).
- [2] Windows Central, *Oculus Rift Mit Leap Motion*. Adresse: <https://www.windowscentral.com/sites/wpcentral.com/files/styles/w830/public/field/image/2018/03/leap-motion-hero-01.jpg> (besucht am 26.10.2020).
- [3] Evan-Amos, *Kinect-Sensorleiste für Xbox One*, 5. Juni 2014. Adresse: <https://de.wikipedia.org/w/index.php?title=Kinect&oldid=209421759> (besucht am 11.03.2021).
- [4] Microsoft, *Azure Kinect DK*. Adresse: <https://img-prod-cms-rt-microsoft-com.akamaized.net/cms/api/am/imageFileData/RWqOsq> (besucht am 26.10.2020).

# **Abkürzungsverzeichnis**

HHI	Heinrich-Hertz-Institut
HMD	Head-Mounted-Display
IK	Inverse Kinematik
imVMT	immersive Virtual Memory Task
imVST	immersive Virtual Supermarket Task
VR	Virtuelle Realität

# Abbildungsverzeichnis

2.1	Verwendete Hardware . . . . .	3
2.2	Beispieldarstellung der Aufgaben in VReha . . . . .	4
2.3	Übersicht des technischen Aufbaus von VReha . . . . .	5
2.4	Aufbau eines humanoiden Avatars in Unity . . . . .	6
4.1	Vereinfachtes Objektdiagramm der Evaluation . . . . .	15
4.2	Veranschaulichung der gegebenen und gelösten Gelenke der IK-Lösungen . . . . .	17
4.3	Positionierungsfehler des rechten Ellenbogens bei Bewegungen . . . . .	18
4.4	Box-Plot Legende . . . . .	19
4.5	Box-Plot der Ellenbogen- und Oberarm-Positionierungsfehler . . . . .	19
4.6	Kerndichteschätzung der Ellenbogen- und Oberarm-Positionierungsfehler . . . . .	20
5.1	Klassenübersicht der Implementierung . . . . .	23
5.2	Posen für die Skalierung des Avatars . . . . .	26
5.3	Positionierung der Hüft- und Fuß-Effektoren für die Sitzpose . . . . .	28
5.5	Klassendiagramm für die Positionierung und Ausrichtung der Kopf- und Armeffektoren . . . . .	29
5.6	Achsenorientierungen der Handgelenke bei verschiedenen Hand-Modellen . . . . .	30
5.7	Platzierte Hände an der Lenkerstange in VReha . . . . .	33
5.9	Avatar mit Kinect-Tracking im Vergleich zum/zur Nutzer*in (1) . . . . .	35
5.10	Avatar mit Kinect-Tracking im Vergleich zum/zur Nutzer*in (2) . . . . .	36
5.11	Unterschiedliche Gelenkorientierungen bei Avataren . . . . .	36
5.12	Koordinatensysteme von Unity (links) und der Azure Kinect (rechts) . . . . .	39
5.13	Vergleich der Auswirkung von Twist Bones auf die Verdrehung des Arm-Meshs . . . . .	40
6.1	Darstellung der im Rahmen der Nutzerstudie durchgeführten Versuche . . . . .	41
6.2	Verkörperungsgefühl in Bezug auf die Hände und den Körper . . . . .	43
6.3	Physisches Präsenzgefühl . . . . .	43
A.1	Standardnormalverteilung der Ellenbogen und Oberarm-Positionierungsfehler . . . . .	60
A.2	Histogramm mit Kerndichteschätzung der Ellenbogen-Positionierungsfehler . . . . .	60
A.3	Histogramm mit Kerndichteschätzung der Oberarm-Positionierungsfehler . . . . .	61
A.4	Ellenbogen-Positionierungsfehler in Abhängigkeit des Oberarm-Positionierungsfehlers . . . . .	61
A.5	Ellenbogen-Positionierungsfehler in Abhängigkeit des Innenarmwinkels . . . . .	62
B.1	Vergleich der Trackingdaten der Azure Kinect mit denen des Oculus Controller zur Bestimmung der Latenz . . . . .	63
C.1	Vergleich der Ergebnisse für Körperbesitz, motorischer Steuerung und Position in Bezug auf die Hände . . . . .	67

---

*Abbildungsverzeichnis*

C.2 Vergleich der Ergebnisse für Körperbesitz, motorischer Steuerung, Position und Aussehen in Bezug auf den Körper . . . . .	67
---	----

# **Quellcodeverzeichnis**

5.1	Skalierung des Avatars gemäß der Körpergröße . . . . .	24
5.2	Bestimmung der Armlänge . . . . .	25
5.3	Erzeugung eines Dictionarys für die lokalen Positions- und Rotationsdaten der Knochen in der T-Pose . . . . .	27
5.4	Beispiel der Rotationoffsetberechnung der linken Oculus Quest Hand . . . . .	30
5.5	Ermittlung, ob die Daten der Leap Motion/Oculus Quest verwendet werden sollen	32
5.6	Beispiel der Rotationoffsetberechnung für die Hüfte und den linken Oberschenkel	37

# Eigenständigkeitserklärung

Ich erkläre hiermit, dass

- ich die vorliegende wissenschaftliche Arbeit selbständig und ohne unerlaubte Hilfe angefertigt habe,
- ich andere als die angegebenen Quellen und Hilfsmittel nicht benutzt habe,
- ich die den benutzten Quellen wörtlich oder inhaltlich entnommenen Stellen als solche kenntlich gemacht habe,
- die Arbeit in gleicher oder ähnlicher Form noch keiner anderen Prüfbehörde vorgelegen hat.

Berlin, den 26.03.2021

L. Wagnerberger

Larissa Wagnerberger

# Anhang

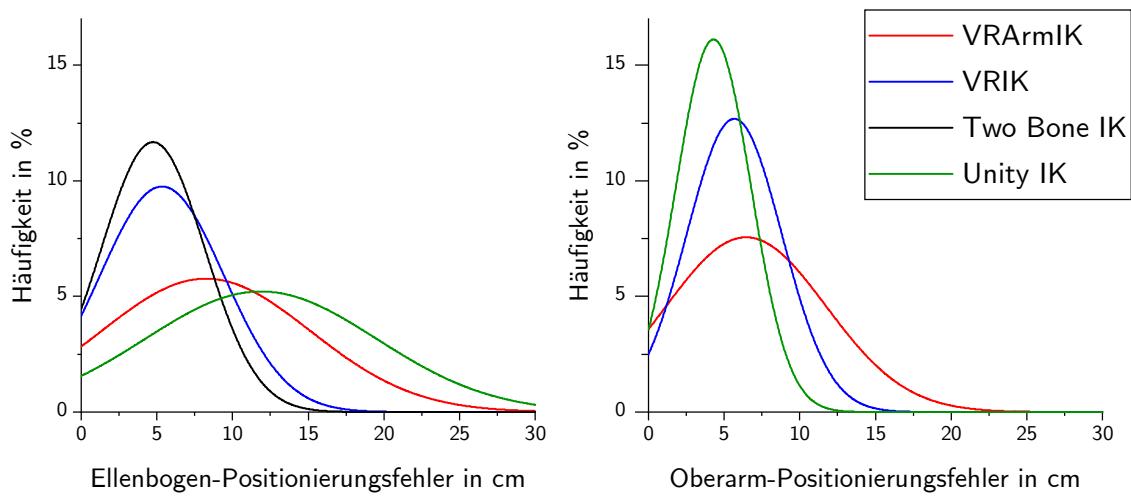
## A Evaluation existierender IK-Lösungen

**Tabelle A.1:** Deskriptive Statistik zu den Ellenbogen-Positionierungsfehlern (in cm)

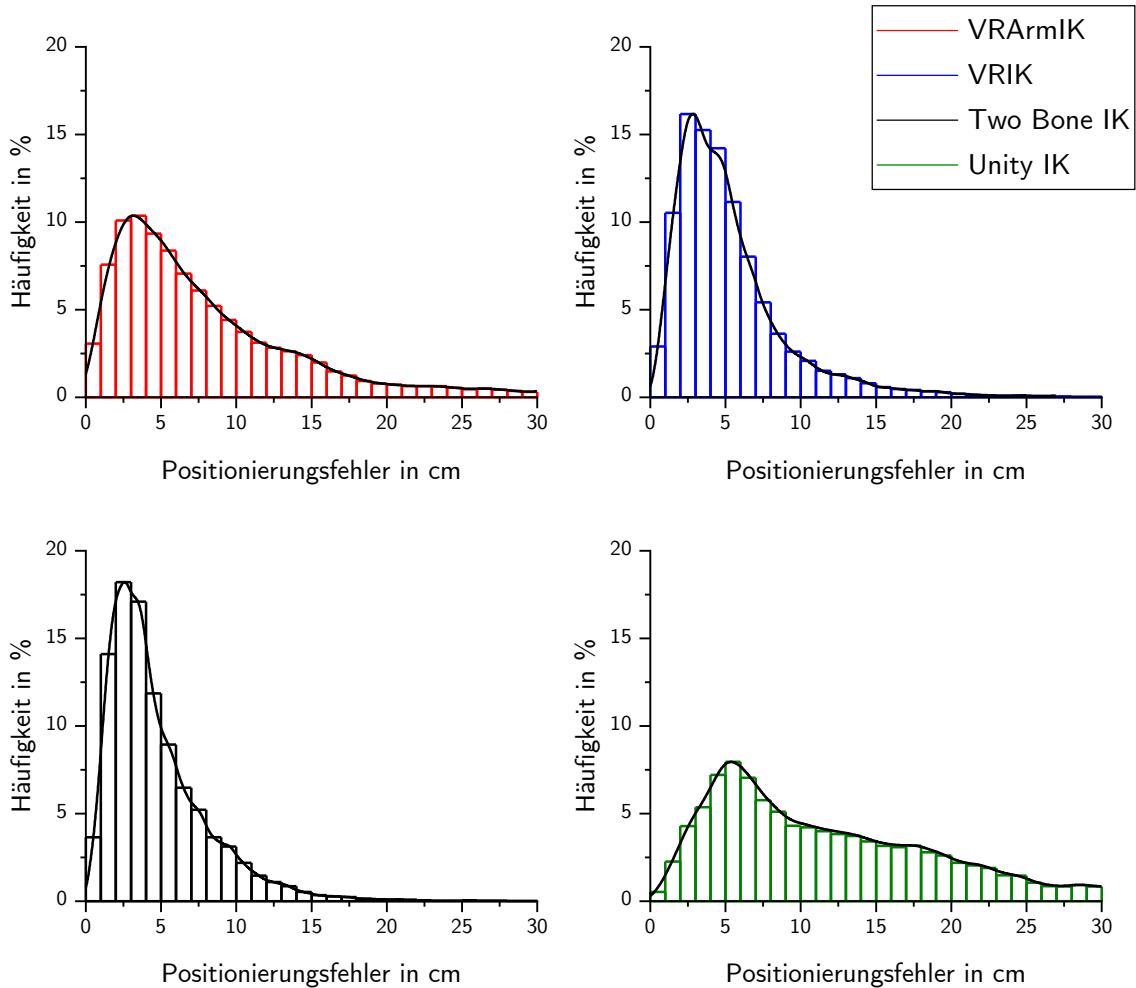
	VRArmIK	VRIK	Two Bone IK	Unity IK
Mittelwert	8,24	5,34	4,75	11,88
Median	6,15	4,36	3,80	10,04
Standardabweichung	6,92	4,09	3,42	7,65
Minimum	0,01	0,04	0,02	0,07
Maximum	66,15	48,44	36,94	39,91
1. Quartil	3,40	2,73	2,41	5,66
3. Quartil	10,88	6,56	6,15	16,92
1. Perzentile	0,59	0,64	0,54	1,27
99. Perzentile	32,50	20,81	16,73	32,17

**Tabelle A.2:** Deskriptive Statistik zu den Oberarm-Positionierungsfehlern (in cm)

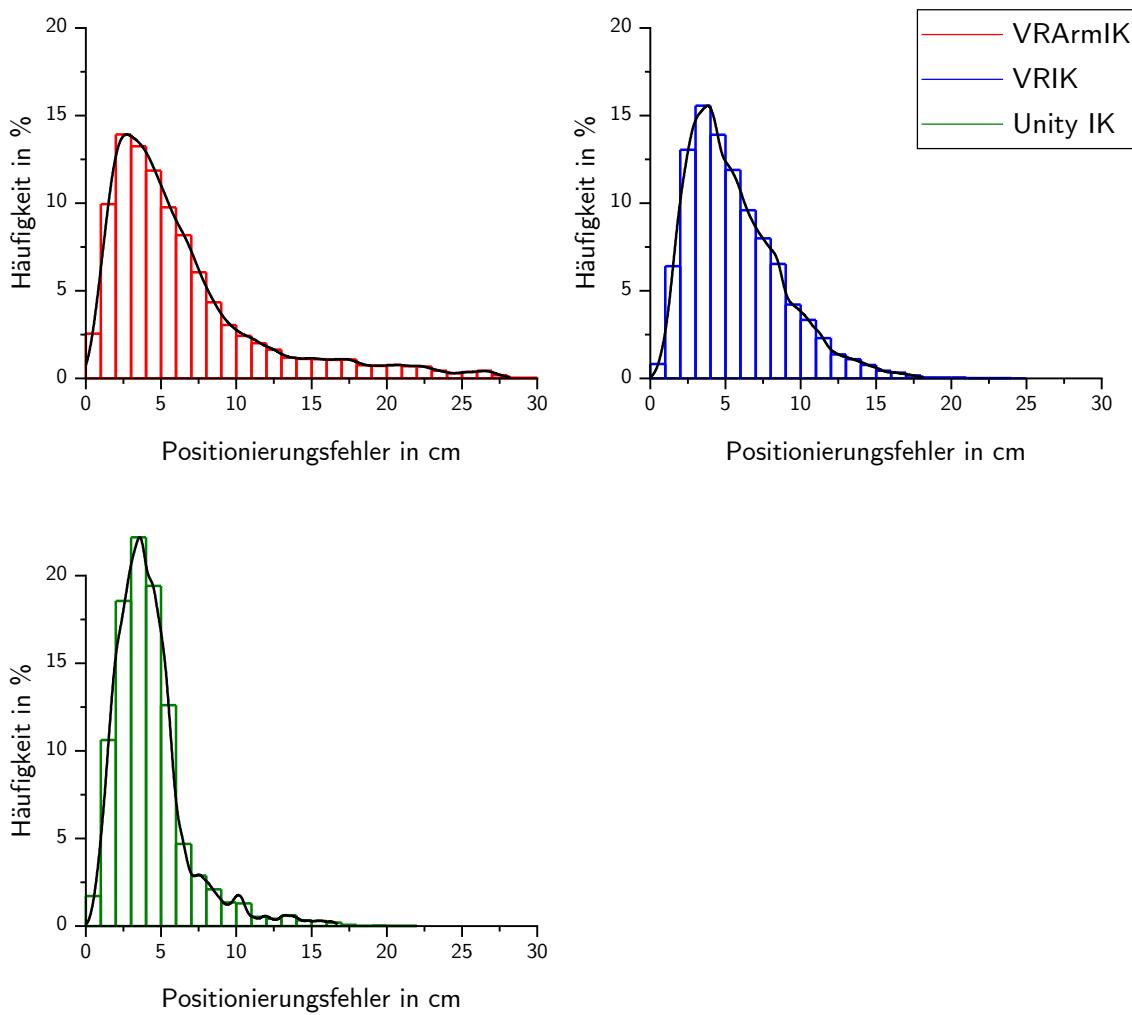
	VRArmIK	VRIK	Unity IK
Mittelwert	6,45	5,67	4,30
Median	4,86	5,02	3,85
Standardabweichung	5,28	3,15	2,48
Minimum	0,02	0,08	0,04
Maximum	39,46	24,08	21,73
1. Quartil	2,90	3,31	2,71
3. Quartil	7,90	7,46	5,15
1. Perzentile	0,71	1,06	0,84
99. Perzentile	25,40	15,20	13,99



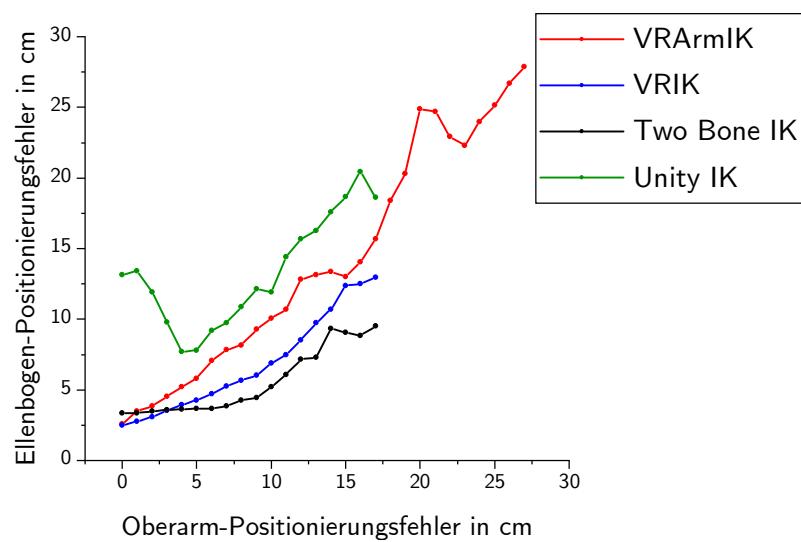
**Abbildung A.1:** Standardnormalverteilung der Ellenbogen und Oberarm-Positionierungsfehler



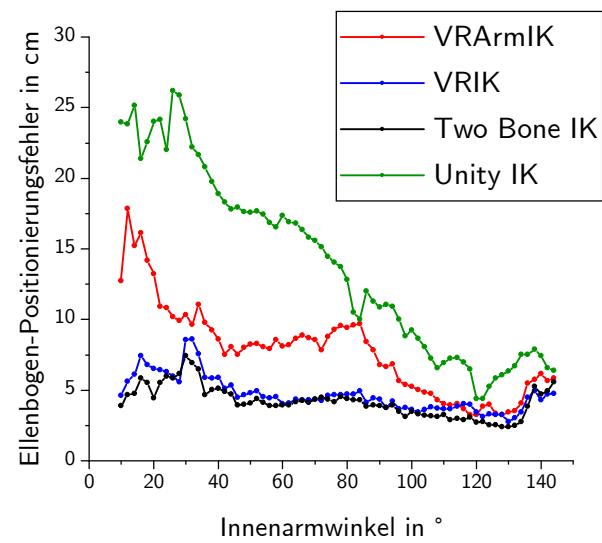
**Abbildung A.2:** Histogramm mit Kerndichteschätzung der Ellenbogen-Positionierungsfehler



**Abbildung A.3:** Histogramm mit Kerndichteschätzung der Oberarm-Positionierungsfehler

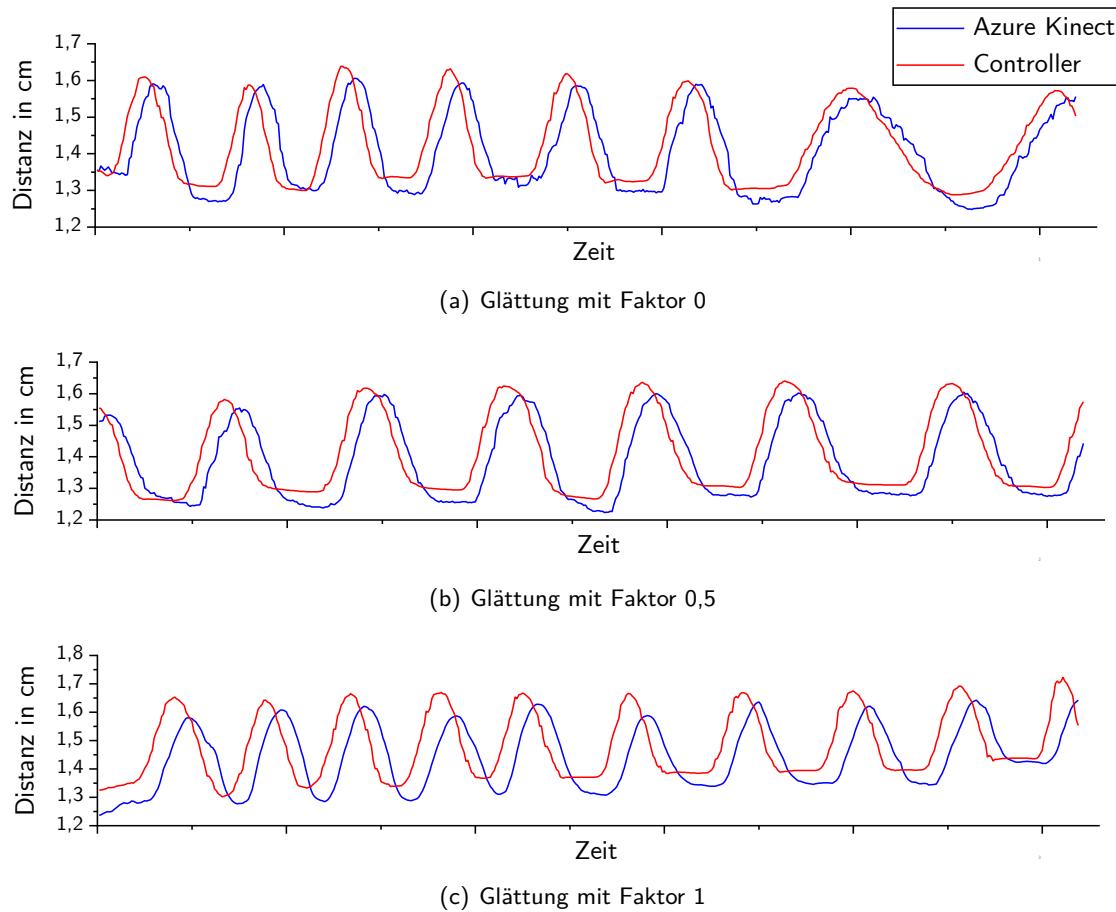


**Abbildung A.4:** Ellenbogen-Positionierungsfehler in Abhängigkeit des Oberarm-Positionierungsfehlers



**Abbildung A.5:** Ellenbogen-Positionierungsfehler in Abhängigkeit des Innenarmwinkels

## B Latenz der Azure Kinect



**Abbildung B.1:** Vergleich der Trackingdaten der Azure Kinect mit denen des Oculus Controller zur Bestimmung der Latenz

**Tabelle B.1:** Ergebnis der Latenzmessungen

Glättungsfaktor	Latenz
0,0	~ 200 ms
0,5	~ 240 ms
1, 0	~ 300 ms

## C Nutzerstudie

### C.1 Fragen

Die Fragen wurden pro Versuch in einer festen zufälligen Reihenfolge gestellt.

#### C.1.1 Fragen bezüglich des Verkörperungsgefühls

Es wurden die Fragen aus dem Paper „Avatar Embodiment. Towards a Standardized Questionnaire“ von Gonzalez-Franco u. a. [61]) verwendet uns ins Deutsche übersetzt. Für die Antworten wurde, wie im Paper vorgeschlagen, eine 7-Punkte Likert-Skala verwendet:

- trifft überhaupt nicht zu (-3), trifft nicht zu (-3), trifft eher nicht zu (-1), teils-teils (0), trifft eher zu (1), trifft zu (2), trifft völlig zu (3)

#### Körperbesitz in Bezug auf die Hände

- Q1: Es fühlte sich so an, als ob die virtuellen Hände meine eigenen Hände wären.  
Q2: Es fühlte sich so an, als ob die Hände, die ich sah, von jemand anderem wären.  
Q3: Es schien, als ob ich mehr als zwei Hände hätte.

#### Motorische Steuerung in Bezug auf die Hände

- Q4: Es fühlte sich so an, als ob ich die virtuellen Hände so steuern konnte, als wären es meine eigenen Hände.  
Q5: Die Bewegungen der virtuellen Hände wurden durch meine Bewegungen verursacht.  
Q6: Ich hatte das Gefühl, als würden die Bewegungen der virtuellen Hände meine eigenen Bewegungen beeinflussen.  
Q7: Ich hatte das Gefühl, als würden sich die virtuellen Hände von selbst bewegen.

#### Position in Bezug auf die Hände

- Q8: Ich hatte das Gefühl, dass sich meine Hände dort befanden, wo ich die virtuellen Hände sah.  
Q9: Ich fühlte mich außerhalb meines Körpers.

#### Körperbesitz in Bezug auf den Körper

- Q10: Es fühlte sich so an, als ob der virtuelle Körper mein eigener Körper wäre.  
Q11: Es fühlte sich so an, als ob der Körper, den ich sah, von jemand anderem wäre.  
Q12: Es schien, als ob ich mehr als einen Körper hätte.

### **Motorische Steuerung in Bezug auf den Körper**

- Q13: Es fühlte sich so an, als ob ich den virtuellen Körper so steuern könnte, als wäre es mein eigener Körper.
- Q14: Die Bewegungen des virtuellen Körpers wurden durch meine Bewegungen verursacht.
- Q15: Ich hatte das Gefühl, als würden die Bewegungen des virtuellen Körpers meine eigenen Bewegungen beeinflussen.
- Q16: Ich hatte das Gefühl, als würde sich der virtuelle Körper von selbst bewegen.

### **Position in Bezug auf den Körper**

- Q17: Ich hatte das Gefühl, dass sich mein Körper dort befand, wo ich den virtuellen Körper sah.

### **Aussehen**

- Q18: Es fühlte sich an, als würde sich mein (realer) Körper in einen 'Avatar'-Körper verwandeln.
- Q19: Irgendwann fühlte es sich so an, als würde mein realer Körper beginnen, die Haltung oder Form des virtuellen Körpers anzunehmen, den ich sah.
- Q20: An einem gewissen Punkt hatte ich das Gefühl, dass der virtuelle Körper meinem eigenen (realen) Körper ähnelte, in Bezug auf Form, Hautton oder andere visuelle Merkmale.
- Q21: Ich hatte das Gefühl, dass ich andere Kleidung während des Versuches trug im Vergleich zu der Kleidung, die ich trug, als ich das Labor betrat.

### **C.1.2 Fragen bezüglich der physischen Präsenz**

Es wurden die übersetzten Fragen der Multimodal Presence Scale [62] benutzt. Für die Antwortmöglichkeiten wurde ebenfalls die 7-Punkte Likert-Skala verwendet.

- Q22: Die virtuelle Umgebung erschien mir real.
- Q23: Ich hatte das Gefühl, in der virtuellen Umgebung zu agieren, anstatt etwas von außen zu kontrollieren.
- Q24: Mein Erleben in der virtuellen Umgebung erschien konsistent mit meinem Erleben in der realen Welt.
- Q25: Während ich in der virtuellen Umgebung war, hatte ich ein Gefühl des 'Dort-Seins'.
- Q26: Ich war komplett gefesselt von der virtuellen Welt.

### **C.1.3 Frage bezüglich der Motion Sickness**

In Bezug auf die Motion Sickness wurde eine Frage gestellt. Es wurde die Fast Motion Sickness Scale [64] für die Antwortmöglichkeiten verwendet:

- 0 (überhaupt keine Übelkeit) - 20 (schwere Übelkeit)

Q27: Fühlen Sie kurz in sich hinein und achten insb. auf Empfindungen von Übelkeit, allgemeinem Unwohlsein und Magenbeschwerden. Ignorieren Sie aber bitte andere Gefühle wie Nervosität, Langeweile oder Müdigkeit.

Wie unwohl bzw. übel fühlen Sie sich gerade?

#### C.1.4 Abschlussfragen

Nachdem alle Versuche durchgeführt wurden, wurden folgende abschließende Fragen gestellt:

1. Alter
2. Geschlecht
3. Wie oft benutzen Sie VR-Anwendungen?
  - nie, selten, mehrmals im Jahr, mehrmals im Monat, mehrmals in der Woche
4. Welche Variante fanden Sie am besten?
5. Warum fanden Sie diese Variante besser als die anderen?

## C.2 Auswertung

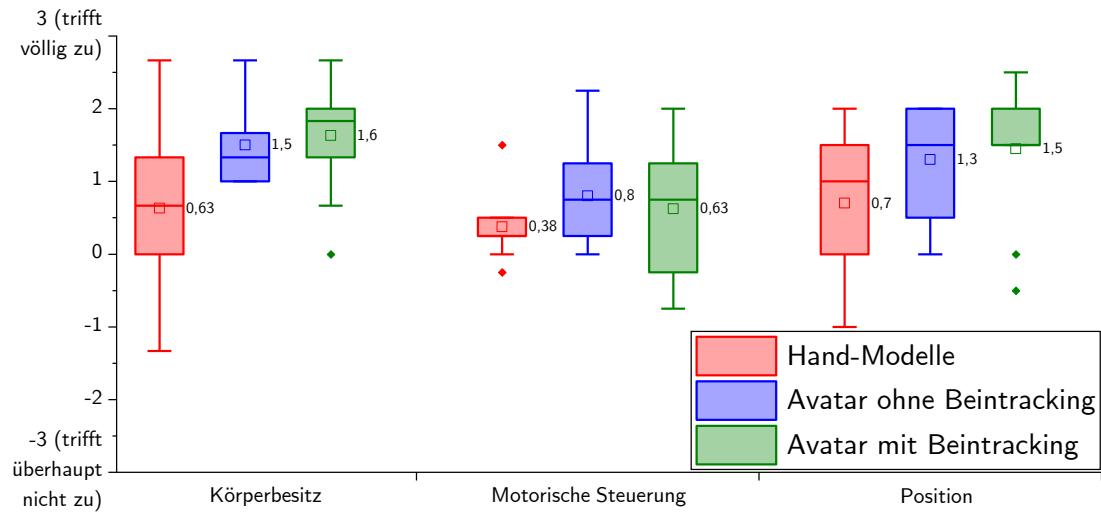
### C.2.1 Verwendete Formeln

Für das Zusammenfassen der Antworten in die Faktoren Körperbesitz, motorische Steuerung, Position und Aussehen wurden die von Gonzalez-Franco u. a. vorgestellten Formeln [61] verwendet. Zur Ermittlung der physischen Präsenz wird die vorgestellte Formel von Volkmann u. a. [62] angewandt.

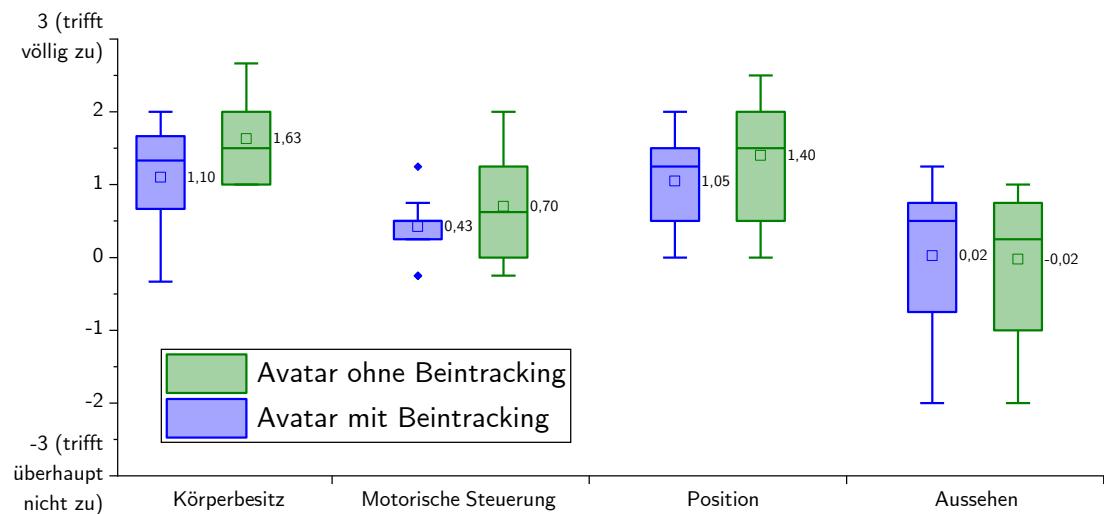
**Tabelle C.1:** Verwendete Formeln

	in Bezug auf die/den	Formel
Körperbesitz	Hände	$(Q1 - Q2) - Q3) / 3$
	Körper	$(Q10 - Q11) - Q12) / 3$
Motorische Steuerung	Hände	$(Q4 + Q5 + Q6 - Q7) / 4$
	Körper	$(Q13 + Q14 + Q15 - Q16) / 4$
Position	Hände	$(Q8 - Q9) / 2$
	Körper	$(Q17 - Q9) / 2$
Aussehen	Körper	$(Q18 + Q19 + Q20 + Q21) / 4$
Verkörper- ungsgefühl	Hände	$(Körperbesitz + Motorische Steuerung + Position) / 3$
	Körper	$(Körperbesitz * 2 + Motorische Steuerung * 2 + Position * 2 + Aussehen) / 7$
Physische Präsenz		$(Q22 + Q23 + Q24 + Q25 + Q26) / 6$

### C.2.2 Ergebnisse der Faktoren des Verkörperungsgefühls



**Abbildung C.1:** Vergleich der Ergebnisse für Körperbesitz, motorischer Steuerung und Position in Bezug auf die Hände



**Abbildung C.2:** Vergleich der Ergebnisse für Körperbesitz, motorischer Steuerung, Position und Aussehen in Bezug auf den Körper

### C.2.3 Ergebnisse der t-Tests

#### Vergleich der Ergebnisse der Verkörperungsfaktoren in Bezug auf die Hände und der physischen Präsenz

**Tabelle C.2:** Vergleich der Ergebnisse von Hand-Modellen mit denen vom Avatar

	t	df	p	Mittlere Differenz	SE
Körperbesitz	1,64	12,50	0,13°	-0,42	0,26
motorische Steuerung	1,32	21,70	0,20°	-0,34	0,36
Position	1,27	20,30	0,22	-0,38	0,40
Verkörperungsgefühl	2,29	11,20	0,04*	-0,69	0,23
Physische Präsenz	1,34	10,30	0,21	-0,56	0,11

Robuster T-Test für unabhängige Stichproben - Yuen's Test

\* signifikanter Unterschied

° Tendenz zu erkennen

**Tabelle C.3:** Vergleich der Ergebnisse von Hand-Modellen mit denen vom Avatar mit Beintracking

	t	df	p	Mittlere Differenz	SE
Körperbesitz	-2,12	7	0,07°	-1,10	0,52
motorische Steuerung	-0,78	7	0,46	-0,31	0,40
Position	-2,07	7	0,08°	-0,81	0,39
Verkörperungsgefühl	-2,91	7	0,02*	-0,75	0,26
Physische Präsenz	-1,81	7	0,11°	-0,43	0,23

Robuster T-Test für gepaarte Stichproben

\* signifikanter Unterschied

° Tendenz zu erkennen

**Tabelle C.4:** Vergleich der Ergebnisse von Hand-Modellen mit denen vom Avatar ohne Beintracking

	t	df	p	Mittlere Differenz	SE
Körperbesitz	-1,64	7	0,15°	-0,80	0,49
motorische Steuerung	-1,94	7	0,09°	-0,41	0,21
Position	-1,43	7	0,20	-0,63	0,44
Verkörperungsgefühl	-2,24	7	0,06*	-0,65	0,29
Physische Präsenz	-2,13	7	0,07°	-0,40	0,19

Robuster T-Test für gepaarte Stichproben

\* signifikanter Unterschied

° Tendenz zu erkennen

**Tabelle C.5:** Vergleich der Ergebnisse von Avatar ohne Beintracking mit denen vom Avatar mit Beintracking

	t	df	p	Mittlere Differenz	SE
Körperbesitz	-1,17	7	0,28	-0,30	0,26
motorische Steuerung	0,28	7	0,79	0,10	0,36
Position	-0,47	7	0,66	-0,19	0,40
Verkörperungsgefühl	-0,44	7	0,67	-0,10	0,23
Physische Präsenz	-0,22	7	0,83	-0,03	0,11

Robuster T-Test für gepaarte Stichproben

### Vergleich der Ergebnisse der Verkörperungsfaktoren in Bezug auf den Körper

**Tabelle C.6:** Vergleich der Ergebnisse vom Avatar ohne Beintracking mit denen vom Avatar mit Beintracking

	t	df	p	Mittlere Differenz	SE
Körperbesitz	-2,07	7	0,08°	-0,43	0,21
motorische Steuerung	-1,05	7	0,33	-0,26	0,25
Position	-1,10	7	0,31	-0,38	0,34
Aussehen	0,10	7	0,93	0,03	0,26
Verkörperungsgefühl	-1,54	7	0,17°	-0,30	0,20

Robuster T-Test für gepaarte Stichproben

° Tendenz zu erkennen

#### C.2.4 Deskriptive Statistiken

**Tabelle C.7:** Deskriptive Statistik zu Faktoren des Verkörperungsgefühls in Bezug auf die Hände und zur physischen Präsenz

	Versuchs typ	Körper- besitz	motorische Steuerung	Position	Verkörper- ungsgefühl	Physische Präsenz
Mittelwert	HM	0,63	0,38	0,70	0,57	0,28
	oBt	1,36	0,66	1,14	1,05	0,62
	mBt	1,81	0,78	1,67	1,42	0,62
Median	HM	0,67	0,25	1,00	0,57	0,40
	oBt	1,33	0,75	1,50	1,25	0,80
	mBt	2,00	0,75	1,50	1,53	0,80
Standard- abweichung	HM	1,31	0,46	1,03	0,80	0,86
	oBt	0,67	0,83	0,95	0,67	0,63
	mBt	0,58	0,91	0,75	0,60	0,89
Minimum	HM	-1,33	-0,25	-1,00	-0,69	-1,00
	oBt	0,00	-0,75	-0,50	-0,42	-0,60
	mBt	0,67	-0,75	0,00	0,47	-1,00
Maximum	HM	2,67	1,50	2,00	1,89	1,60
	oBt	2,67	2,25	2,00	1,83	1,40
	mBt	2,67	2,00	2,50	2,39	1,60

HM = Hand-Modelle

oBt = Avatar ohne Beintracking

mBt = Avatar mit Beintracking

**Tabelle C.8:** Deskriptive Statistik zu den Faktoren des Verkörperungsgefühls in Bezug auf den Körper

	Versuchs typ	Körper- besitz	motorische Steuerung	Position	Verkörper- ungsgefühl	Physische Präsenz
Mittelwert	mBt	1,6	0,7	1,4	0,0	1,1
	oBt	1,1	0,4	1,1	0,0	0,7
Median	mBt	1,5	0,6	1,5	0,3	0,9
	oBt	1,3	0,5	1,3	0,5	0,7
Standard- abweichung	mBt	0,6	0,7	0,9	1,0	0,7
	oBt	0,7	0,4	0,6	1,1	0,4
Minimum	mBt	1,0	-0,3	0,0	-2,0	0,2
	oBt	-0,3	-0,3	0,0	-2,0	0,2
Maximum	mBt	2,7	2,0	2,5	1,0	2,2
	oBt	2,0	1,3	2,0	1,3	1,4

oBt = Avatar ohne Beintracking

mBt = Avatar mit Beintracking

### C.3 Antworten

#### C.3.1 Antworten bezüglich des Verkörperungsgefühls

Q1: Es fühlte sich so an, als ob die virtuellen Hände meine eigenen Hände wären.

Person ID	Hand- Modelle	ohne Bein- tracking	mit Bein- tracking
1	-2	2	2
2	0	0	1
3	0	1	-2
4	-3	1	1
5	0	0	2
7	2	2	2
8	0	1	1
9	-1	0	2
10	-1	1	1
11	0	1	-2

Q2: Es fühlte sich so an, als ob die Hände, die ich sah, von jemand anderem wären.

Person ID	Hand- Modelle	ohne Bein- tracking	mit Bein- tracking
1	1	-2	-2
2	1	-1	-1
3	-2	-2	-2
4	3	0	-2
5	-2	-2	-2
7	-3	-3	-3
8	-3	-1	-2
9	-1	0	-2
10	0	-1	-2
11	1	0	0

Q3: Es schien, als ob ich mehr als zwei Hände hätte.

Person ID	Hand- Modelle	ohne Bein- tracking	mit Bein- tracking
1	1	-2	-1
2	-1	-2	-2
3	-2	-2	-2
4	-2	-3	-3
5	-2	-3	-2
7	-3	-3	-3
8	-3	-2	-2
9	-3	-3	-3
10	-2	-2	-3
11	-2	-2	-2

Q4: Es fühlte sich so an, als ob ich die virtuellen Hände so steuern konnte, als wären es meine eigenen Hände.

Person ID	Hand- Modelle	ohne Bein- tracking	mit Bein- tracking
1	0	2	2
2	0	1	1
3	1	1	1
4	0	2	1
5	2	0	2
7	2	2	1
8	1	2	2
9	0	2	1
10	0	1	1
11	2	1	-1

Q5: Die Bewegungen der virtuellen Hände wurden durch meine Bewegungen verursacht.

Person ID	Hand- Modelle	ohne Bein- tracking	mit Bein- tracking
1	0	2	2
2	1	1	2
3	1	2	-1
4	0	2	1
5	2	-1	2
7	2	2	2
8	2	2	2
9	1	1	2
10	1	1	1
11	2	1	-2

Q6: Ich hatte das Gefühl, als würden die Bewegungen der virtuellen Hände meine eigenen Bewegungen beeinflussen.

Person ID	Hand- Modelle	ohne Bein- tracking	mit Bein- tracking
1	2	0	0
2	-2	-2	0
3	-1	-2	-2
4	1	-1	-2
5	-2	-1	-2
7	2	2	2
8	-2	-1	1
9	0	-3	-2
10	-1	-2	-2
11	-2	-2	-2

Q7: Ich hatte das Gefühl, als würden sich die virtuellen Hände von selbst bewegen.

Person ID	Hand- Modelle	ohne Bein- tracking	mit Bein- tracking
1	0	-2	-1
2	-1	-1	-2
3	1	-2	1
4	0	-1	1
5	-2	-2	-1
7	-3	-3	-3
8	-2	-2	-2
9	-1	-3	-2
10	0	0	-1
11	-1	-1	-2

Q8: Ich hatte das Gefühl, dass sich meine Hände dort befanden, wo ich die virtuellen Hände sah.

Person ID	Hand- Modelle	ohne Bein- tracking	mit Bein- tracking
1	1	2	1
2	1	1	2
3	1	1	1
4	1	2	1
5	2	2	2
7	2	2	2
8	2	2	1
9	1	2	2
10	1	0	1
11	0	1	-1

Q9: Ich fühlte mich außerhalb meines Körpers.

Person ID	Hand-Modelle	ohne Bein-tracking	mit Bein-tracking
1	1	-2	-2
2	-1	-2	-2
3	1	-2	-2
4	3	-1	1
5	-2	-2	-2
7	-1	1	-3
8	-2	-2	-2
9	-1	-2	-3
10	2	0	-2
11	-2	1	0

Q10: Es fühlte sich so an, als ob der virtuelle Körper mein eigener Körper wäre.

Person ID	ohne Beintracking	mit Beintracking
1	1	1
2	-1	0
3	2	1
4	0	0
5	1	2
7	0	2
8	0	1
9	1	2
10	-1	1
11	1	0

Q11: Es fühlte sich so an, als ob der Körper, den ich sah, von jemand anderem wäre.

Person ID	ohne Beintracking	mit Beintracking
1	-1	-1
2	-1	-1
3	-2	-2
4	2	0
5	-3	-2
7	-1	-3
8	-1	-2
9	-1	-3
10	-1	-1
11	-1	-2

Q12: Es schien, als ob ich mehr als einen Körper hätte.

Person ID	ohne Beintracking	mit Beintracking
1	-2	-1
2	-1	-2
3	-2	-2
4	-1	-3
5	-1	-2
7	-3	-3
8	-2	-2
9	-3	-3
10	-2	-2
11	-2	-2

Q13: Es fühlte sich so an, als ob ich den virtuellen Körper so steuern könnte, als wäre es mein eigener Körper.

Person ID	ohne	mit
	Beintracking	Beintracking
1	1	1
2	0	-1
3	1	1
4	0	1
5	0	1
7	-2	1
8	1	1
9	1	2
10	1	2
11	1	0

Q14: Die Bewegungen des virtuellen Körpers wurden durch meine Bewegungen verursacht.

Person ID	ohne	mit
	Beintracking	Beintracking
1	2	2
2	1	0
3	1	1
4	-2	1
5	2	2
7	1	2
8	1	2
9	1	2
10	1	1
11	2	1

Q15: Ich hatte das Gefühl, als würden die Bewegungen des virtuellen Körpers meine eigenen Bewegungen beeinflussen.

Person ID	ohne	mit
	Beintracking	Beintracking
1	-1	0
2	0	-1
3	-2	-2
4	0	-3
5	-1	-1
7	-3	2
8	-2	0
9	0	-1
10	-1	-1
11	-2	-2

Q16: Ich hatte das Gefühl, als würde sich der virtuelle Körper von selbst bewegen.

Person ID	ohne	mit
	Beintracking	Beintracking
1	-1	-1
2	-1	-1
3	-2	-2
4	-1	-1
5	-1	0
7	-3	-3
8	-2	-2
9	-3	-2
10	0	-1
11	-1	-1

Q17: Ich hatte das Gefühl, dass sich mein Körper dort befand, wo ich den virtuellen Körper sah.

Person ID	ohne	mit
	Beintracking	Beintracking
1	2	1
2	-1	-1
3	1	1
4	1	1
5	1	2
7	2	2
8	1	2
9	1	2
10	1	1
11	1	0

Q18: Es fühlte sich an, als würde sich mein (realer) Körper in einen 'Avatar'-Körper verwandeln.

Person ID	ohne	mit
	Beintracking	Beintracking
1	-1	0
2	-1	-2
3	-2	-2
4	3	2
5	0	2
7	2	2
8	0	0
9	1	2
10	1	-1
11	1	2

Q19: Irgendwann fühlte es sich so an, als würde mein realer Körper beginnen, die Haltung oder Form des virtuellen Körpers anzunehmen, den ich sah.

Person ID	ohne	mit
	Beintracking	Beintracking
1	-1	1
2	-1	-1
3	-2	-2
4	-2	-2
5	1	-1
7	-1	2
8	-1	-1
9	1	-1
10	-1	-2
11	-1	-2

Q20: An einem gewissen Punkt hatte ich das Gefühl, dass der virtuelle Körper meinem eigenen (realen) Körper ähnelte, in Bezug auf Form, Hautton oder andere visuelle Merkmale.

Person ID	ohne	mit
	Beintracking	Beintracking
1	0	0
2	-1	1
3	-2	-2
4	0	0
5	0	1
7	2	2
8	1	1
9	-1	1
10	1	1
11	-1	-1

Q21: Ich hatte das Gefühl, dass ich andere Kleidung während des Versuches trug im Vergleich zu der Kleidung, die ich trug, als ich das Labor betrat.

Person ID	ohne	mit
	Beintracking	Beintracking
1	-1	-1
2	-2	-2
3	-2	-2
4	3	2
5	2	2
7	2	-3
8	2	2
9	2	2
10	1	-2
11	-1	1

### C.3.2 Antworten bezüglich der physischen Präsenz

Q22: Die virtuelle Umgebung erschien mir real.

Person ID	Hand-Modelle	ohne Bein-tracking	mit Bein-tracking
		Beintracking	Beintracking
1	0	1	1
2	-1	-1	-2
3	1	1	1
4	-3	-2	-2
5	1	1	1
7	0	0	0
8	0	0	1
9	1	1	1
10	1	1	1
11	0	1	0

Q23: Ich hatte das Gefühl, in der virtuellen Umgebung zu agieren, anstatt etwas von außen zu kontrollieren.

Person ID	Hand-Modelle	ohne Bein-tracking	mit Bein-tracking
		Beintracking	Beintracking
1	-1	1	2
2	0	0	0
3	1	1	-2
4	1	2	0
5	0	2	2
7	2	2	3
8	1	1	1
9	1	1	2
10	1	2	1
11	-2	-1	0

Q24: Mein Erleben in der virtuellen Umgebung erschien konsistent mit meinem Erleben in der realen Welt.

Person ID	Hand- Modelle	ohne Bein- tracking	mit Bein- tracking
1	-1	0	0
2	-1	0	-1
3	1	1	1
4	-2	-1	0
5	0	1	1
7	1	0	0
8	0	0	0
9	0	0	1
10	1	1	1
11	0	0	-1

Q25: Während ich in der virtuellen Umgebung war, hatte ich ein Gefühl des 'Dort-Seins'

Person ID	Hand- Modelle	ohne Bein- tracking	mit Bein- tracking
1	0	1	1
2	-1	-1	-1
3	1	1	1
4	0	1	2
5	1	2	2
7	3	2	3
8	2	1	1
9	1	2	2
10	2	1	2
11	2	2	2

Q26: Ich war komplett gefesselt von der virtuellen Welt.

Person ID	Hand- Modelle	ohne Bein- tracking	mit Bein- tracking
1	-1	0	0
2	-1	-1	-1
3	-2	-2	-2
4	-1	-2	-1
5	0	1	2
7	2	0	0
8	1	2	1
9	0	2	1
10	1	1	1
11	1	2	2

### C.3.3 Antworten bezüglich der Motion Sickness

Q27: Fühlen Sie kurz in sich hinein und achten insb. auf Empfindungen von Übelkeit, allgemeinem Unwohlsein und Magenbeschwerden. Ignorieren Sie aber bitte andere Gefühle wie Nervosität, Langeweile oder Müdigkeit. – Wie unwohl bzw. übel fühlen Sie sich gerade?

Person ID	Hand-Modelle	ohne Bein-tracking	mit Bein-tracking
1	5	2	4
2	7	10	8
3	3	5	8
4	15	15	14
5	7	9	7
7	3	2	0
8	4	0	0
9	0	0	0
10	3	3	4
11	3	8	8

### C.3.4 Antworten der Abschlussfragen

Allgemeine Informationen über die Teilnehmer\*innen

Person ID	Alter	Geschlecht	Nutzung von VR-Anwendungen
1	20	weiblich	selten
2	27	männlich	selten
3	31	weiblich	selten
4	51	männlich	mehrmals in der Woche
5	31	männlich	mehrmals im Jahr
7	40	männlich	selten
8	48	weiblich	mehrmals im Monat
9	34	männlich	nie
10	57	männlich	mehrmals im Monat
11	34	weiblich	selten

Welche Variante fanden Sie am besten?

Warum fanden Sie diese Variante besser als die anderen?

---

Person ID	Antwort
1	Avatar mit Beintracking Mir ist weniger schlecht geworden, weil ich mich selbst sehen konnte und meine Bewegungen mit denen des Avatars übereinstimmten. Dadurch viel es mir auch leichter auf der Stelle zu laufen und nicht nach vorne oder zur Seite zu laufen
2	Avatar mit Beintracking Die Hände waren besser
3	Avatar mit Beintracking You can see the Movement of your feet.
4	Avatar mit Beintracking Es wirkt realistischer wenn alle meiner Bewegung auch in die VR abgebildet werden, besonders der Beinen und klar der armen. Die Darstellung der Haut, die Struktur und Klamotten sollten ähnlich meiner sein, nicht unbedingt identisch aber ähnlich.
5	Avatar mit Beintracking Durch die bessere Bewegung des Avatars fühlte sich der Einkauf realistischer/immersiver an.
7	Avatar ohne Beintracking Beim Avatar ohne Kinect, hatte ich das Gefühl, dass die Steuerung bzw. das Tracking akkurater war, mit Kinect-Tracking hatte ich deutlich mehr Probleme beim Greifen der Produkte, dafür aber eher das Gefühl, dass der Avatar mir ähnlicher ist.
8	Avatar mit Beintracking Körper hat am besten reagiert, inklusive Füße, keine Probleme mit in den Körper schauen. Bei fehlender Körperrepräsentation hat man das Gefühl, körperlos zu sein, zu schweben und etwas zu hoch zu sein, nicht passend zur eigenen Position.
9	Avatar mit Beintracking Die Beine wurden beim Beugen des Oberkörpers und bei anderen Beinbewegungen besser positioniert bzw. in VR dargestellt. Die Variante ohne Arme fühlte sich etwas unreal an - wie zwei schwebende Handschuhe.
10	Avatar mit Beintracking Die Körperbewegung war realistischer. Ich habe allerdings nicht sehr auf meine Beine geachtet, da ich zu sehr mit dem Einkaufen/Greifen beschäftigt war. Die realistische Darstellung der Hände ist viel besser als die grauen Roboterhände.
11	Avatar mit Beintracking Die Hände waren mehr realistisch in der zweiter Version. Die Steuerung der Hände war besser und der Effekt von der Beine-Bewegung.

---