

BACHELORARBEIT

Vergleich der Benutzerfreundlichkeit von Mehrfachauswahl-Methoden in Augmented Reality

vorgelegt 27. Juli 2022
Azad Amid

Erstprüfer: Prof. Dr. Eike Langbehn
Zweitprüfer: Simon Dewert

**HOCHSCHULE FÜR ANGEWANDTE
WISSENSCHAFTEN HAMBURG**
Department Medientechnik

Zusammenfassung

Mehrfachauswahl-Methoden können den Prozess der Manipulation von Objekten verschnellern, sei es auf einer Desktopumgebung oder in Virtual Reality. Jedoch ist die Mehrfachauswahl für AR beziehungsweise Handheld-AR nicht weiter erforscht wurden. Aus diesem Grund wurden für diese Arbeit vier Mehrfachauswahl-Methoden, auf der Grundlage bestehender Methoden und Metaphern, für Handheld-AR entworfen und implementiert. Genauer befasst sich diese Arbeit mit dem Vergleich zwischen Mehrfachauswahl-Methoden, die den 2D-Kontext und 3D-Kontext als Eingabe benutzen. Anhand einer Nutzerstudie wurde die Benutzerfreundlichkeit dieser Methoden untersucht. Die Ergebnisse zeigen das Mehrfachauswahl-Methoden, die den 2D Kontext benutzen, für Nutzer eine bessere Benutzerfreundlichkeit aufweisen können und es sich aber auch vermuten lässt, dass Mehrfachauswahl-Methoden, die den 3D-Kontext benutzen, durch Weiterentwicklungen, auch eine gute Benutzerfreundlichkeit haben können.

Abstract

Multiple selection methods can speed up the process of manipulating objects, whether on a desktop environment or in Virtual Reality. However, multiple selection for AR and Handheld-AR, respectively, has not been further explored. For this reason, four multiple selection methods, based on existing multiple selection methods and metaphors, were designed and implemented for Handheld-AR. More specifically, this thesis addresses the comparison between multiple selection methods that use the 2D context and 3D context as input. A user study was conducted to investigate the usability of these methods. The results show that multi-selection methods using the 2D context can have a better usability for users and it can also be assumed that multi-selection methods using the 3D context can also have a good usability through further developments.

Inhaltsverzeichnis

Abkürzungsverzeichnis	V
Abbildungsverzeichnis	VI
Tabellenverzeichnis	VIII
1 Einleitung.....	1
1.1 Zielsetzung.....	1
1.2 Struktur der Arbeit	2
2 Grundlagen	3
2.1 Augmented Reality	3
2.2 Tracking	4
2.2.1 Nicht visuelles Tracking.....	4
2.2.2 Visuelles Tracking.....	4
2.3 Displays.....	5
2.3.1 Video-See-Through.....	5
2.3.2 Optical-See-Through.....	7
2.4 Interaktion in AR	7
2.4.1 Interaktionsaufgaben	8
2.4.2 Interaktions-Typen in Handheld-AR.....	8
3 Auswahl in 3D.....	10
3.1 Auswahlaufgabe.....	10
3.1.1 Auswahlparameter.....	10
3.1.2 Einfluss der Eingabegeräte	10
3.1.3 Klassifizierung per Taxonomie	11
3.2 Auswahlmetapher	12
3.2.1 Zeige-Methoden	13
3.2.2 Oberflächenbasierte Methoden.....	14
3.3 Mehrfachauswahl-Methoden	14
3.3.1 Mehrfachauswahl-Methoden im 2D-Kontext.....	15
3.3.2 Mehrfachauswahl-Methoden im 3D-Kontext.....	15
4 Konzeption.....	17
4.1 Entwurfs-Richtlinien.....	17
4.2 Methoden im 3D Kontext	18

4.2.1	Sphere-Select.....	18
4.2.2	Box-Select	19
4.3	Methoden im 2D-Kontext	22
4.3.1	Drag-Select.....	22
4.3.2	Rect-Select.....	23
4.4	Manipulation der Objekte	24
5	Implementierung.....	25
5.1	AR-Framework	25
5.2	Entwicklungsumgebung.....	26
5.3	Anwendung.....	26
5.3.1	Platzierung der Szene	26
5.3.2	Ray-Casting.....	27
5.3.3	Implementierung der Mehrfachauswahl-Methoden	27
5.3.3.1	Sphere-Select.....	27
5.3.3.2	Box-Select	28
5.3.3.3	Rect-Select.....	29
5.3.3.4	Drag-Select.....	30
6	Nutzerstudie.....	31
6.1	Teilnehmer	31
6.2	Testplattform.....	31
6.3	Aufgaben.....	31
6.4	Methode	33
6.5	Auswertung.....	34
6.5.1	Subjektive Ergebnisse	34
6.5.2	Objektive Ergebnisse.....	35
6.6	Diskussion.....	36
7	Fazit und Aussicht	39
	Literaturverzeichnis.....	41
	Eigenständigkeitserklärung	45

Abkürzungsverzeichnis

DOF	Degree of Freedom
AR	Augmented Reality
VR	Virtual Reality
VE	Virtual Environment
HMD	Head Mounted Display
OST	Optical See Through

Abbildungsverzeichnis

Abbildung 2.1 – Das Reality-Virtuality Continuum nach Milgram et. al [6].....	3
Abbildung 2.2 – Typische Marken, wie sie für kamerabasiertes Tracking eingesetzt werden	5
Abbildung 2.3 – Ein Video-See-Through System. Die Umgebung wird von einer Kamera aufgenommen und wird dann, mit virtuellen Inhalten, auf einem Display angezeigt [8].	6
Abbildung 2.4 – Ein Optical-See-Through System in Anlehnung an Azuma R. [8]	7
Abbildung 3.1 – Die Taxonomie für Mehrfachauswahl-Methoden in 3D von Lucas J. [17].....	12
Abbildung 3.2 – Ray-Casting auf einem virtuellen Objekt mit einem Handheld Eingabegerät. Der Schnidepunkt des Vektors wird hier mit einem 3D-Cursor visualisiert.	13
Abbildung 4.1 - Darstellung der Sphere-Cast Methode. Der Schnidepunkt wird mit einem 3D-Cursor dargestellt.	18
Abbildung 4.2 – (a) Darstellung der Größenänderung der Kugel durch einer Wisch-Geste auf der X-Achse. (b) Darstellung der Höhenänderung durch eine Wisch-Geste auf der Y-Achse.....	19
Abbildung 4.3 – Eine abstrakte Darstellung vom ersten Modus, mit Ausschnitt der Benutzeroberfläche. Die erkannte Fläche wird mit einer rot gepunkteten Ebene visualisiert.....	20
Abbildung 4.4 – Abstrakte Darstellung des zweiten Modus von der Box-Select Methode	20
Abbildung 4.5 – Abstrakte Darstellung des dritten Modus der Box-Select Methode	21
Abbildung 4.6 – Bildschirmaufnahme der Box-Select Methode mit „Auswahl“-Button (unten links) und „Beweg“-Button (unten rechts).....	21
Abbildung 4.7 – Auswahl von 3D-Objekten mir der Drag-Select Methode. Innerhalb der weißen Umrandung wird die Wisch-Geste visualisiert.....	22
Abbildung 4.8 – Auswahl von 3D-Objekten mit der Rect-Select Methode. Der Halte-Geste wird mit einem Kreis visualisiert.....	23
Abbildung 5.1 - Darstellung der erkannten Merkmale und Ebenen in der 3D-Szene (a) Darstellung (b) zeigt diese in Bezug zum realen Raum.....	25
Abbildung 5.2 - Darstellung der transparent und weiß gepunkteten Fläche.	27
Abbildung 5.3 – Kollision des Anzeigeobjektes der Sphere-Select Methode mit einem auswählbaren 3D-Objekt.....	28
Abbildung 5.4 – Normalisierter Kubus in einem Koordinatensystem. Der Ursprungspunkt des Koordinatensystems ist am Schnidepunkt des virtuellen Strahles.....	28
Abbildung 5.5 –Skalierung des Kubus in einem Koordinatensystem. Der normalisierte Kubus wird durch die Veränderung von E2 bis zu der gewünschten Größe skaliert.....	29
Abbildung 6.1 - Darstellung der Starteraufgabe. (a) zeigt eine Draufsicht der Aufgabe und (b) eine Ansicht von vorne.	32
Abbildung 6.2 – Darstellung der Reihen-Aufgaben in Unity. (a1) zeigt eine schiefe Ansicht der Aufgabe. (a2), (b) und (c) zeigen eine Draufsicht der drei Reihen-Aufgaben.....	32

Abbildung 6.3 – Darstellung	33
Abbildung 6.4 - Mittlere Lösungszeiten der Aufgaben pro Methode. Die x-Achse stellt die Methoden pro Aufgabe dar. Die y-Achse die Zeit für das lösen der Aufgabe.	36

Tabellenverzeichnis

Tabelle 4.1 - Einordnung der Methoden in die Taxonomie für Mehrfachauswahl-Methoden in 3D....	17
Tabelle 6.1 - SUS Punktzahlen der Methoden. (M) Zeigt den Mittelwert der Ergebnisse pro Methode an. (SD) zeigt die Standardabweichung der Punktezahlen an.	34
Tabelle 6.2 – Rangordnung der Methoden basierend auf der Präferenz der Teilnehmer.....	35

1 Einleitung

Der Einsatz von AR ist in den letzten Jahren, aufgrund von voranschreitenden Technologien von Smartphones und Tablets (Handheldgeräte), zugänglicher und dadurch beliebter geworden. Nutzer können mithilfe dieser Handheldgeräte schnell und einfach in AR-Erlebnisse eintauchen, indem sie beispielsweise eine App herunterladen. Den Einsatz von Augmented Reality in Smartphones und Tablets bezeichnet diese Arbeit als Handheld-AR.

Augmented Reality (AR) vermischt virtuelle Inhalte mit der realen Welt und erlaubt es virtuelle Objekte in den eigenen vier Wänden zu platzieren, um diese wie reale Gegenstände zu betrachten oder damit zu interagieren. Ein bekanntes Beispiel ist die „Ikea Place“ App [1], in der virtuelle Möbelstücke im 3D-Raum platziert werden können, um so eine Vorschau der Möbel im eigenen Zimmer zu erhalten.

Die Auswahl ist hier eine wichtige Aufgabe bei der Interaktion mit 3D-Objekten [2]: denn möchten Nutzer diese virtuellen Objekte bearbeiten, müssen diese erst einmal ausgewählt werden. In Handheld-AR kann die Auswahl auf unterschiedlicher Weise durchgeführt werden. Diese sind jedoch immer nur auf die Auswahl von einem einzelnen Objekt limitiert, wie es auch bei der Ikea Place App der Fall ist.

Wird AR als kuratorisches Instrument betrachtet, dann wäre es nicht unüblich, dass sehr viele 3D-Objekte in einer Szene platziert sind und bearbeitet werden müssen. Zum Beispiel kann man sich hier einen Nutzer vorstellen, der sein Zimmer mit weiteren Möbelstücken erweitern will. Mithilfe von AR, platziert er die virtuellen Gegenstände seiner Wunschmöbelstücke in sein Zimmer. Ihm fällt auf, nachdem alles platziert wurde, dass ihm die Farben nicht gefallen und er alles gleichfarbig haben will. Damit das passieren kann, muss er alle Möbelstücke einzeln auswählen, um die Farben zu ändern. Schneller wäre es hier aber, wenn er mehrere Objekte gleichzeitig auswählen könnte und nur einmal die Farbe ändern muss. Für solche eine Auswahl benötigt der Nutzer Mehrfachauswahl-Methoden, die es ihm ermöglichen mehrere 3D-Objekte auszuwählen. Aus diesem Grund sind Untersuchungen von Mehrfachauswahl-Methoden für Handheld-AR von Interesse, um den Arbeitsprozess in solchen Design-Erlebnissen zu vereinfachen.

1.1 Zielsetzung

Es gibt zahlreiche Arbeiten die Auswahlmethoden in Handheld-AR erforschen [3], [4], doch diese beziehen sich auf die Auswahl von einzelnen 3D-Objekten. Während der Recherche für diese Arbeit wurden keine Arbeiten gefunden die Mehrfachauswahl-Methoden in Handheld-AR behandeln. Aus diesem Grund werden im Rahmen dieser Arbeit, auf Grundlage bekannter Methoden, vier Mehrfachauswahl-Methoden für Handheld-AR konzipiert und implementiert. Ziel dieser Arbeit ist es diese Methoden auf ihre Benutzerfreundlichkeit anhand einer Nutzerstudie zu überprüfen und folgende Fragen zu beantworten:

- *Wie unterscheidet sich die Benutzerfreundlichkeit zwischen 2D- und 3D-Mehrfachauswahlmethoden in Handheld-AR?*

Durch die unterschiedlichen Eingabemöglichkeiten von Handheldgeräten können Mehrfachauswahl-Methoden ihren Auswahlbereich direkt auf der Bildebene des Displays (2D-Mehrfachauswahl-Methode) oder im 3D-Raum (3D-Mehrfachauswahl-Methode) der AR-Anwendung definieren. Für weitere Entwicklungszwecke kann es sinnvoll zu wissen, welche Ansätze bei Mehrfachauswahl-Methoden in Handheld-AR verfolgt werden soll.

- *Inwiefern beeinflussen 2D Auswahl Metapher die 3D-Mehrfachauswahl-Methoden?*

Metapher können beim Nutzen von Interaktionsmethoden behilflich sein, aber inwiefern beeinflusst eine 2D-Metapher die 3D-Mehrfachauswahl?

1.2 Struktur der Arbeit

In Kapitel 2 werden zunächst die Grundlagen behandelt, die für das Verständnis der folgenden Arbeit benötigt wird. Anschließend werden in Kapitel 3 die Auswahl in 3D thematisiert, welche die Grundlage für die Konzeption der Mehrfachauswahl-Methoden bilden. In Kapitel 4 wird die Konzeption der vier Mehrfachauswahl-Methoden beschrieben. Das 5. Kapitel beschreibt die Implementierung der Mehrfachauswahl Methoden und die dabei genutzten Technologien. Anhand einer Nutzerstudie werden die vier Mehrfachauswahl-Methode im 6. auf ihre Benutzerfreundlichkeit evaluiert. Dabei werden die Ergebnisse vorgestellt und in der Diskussion aufgegriffen. Abgeschlossen wird diese Arbeit im 7. Kapitel mit dem Fazit und der Aussicht.

2 Grundlagen

2.1 Augmented Reality

Augmented Reality (AR) erlaubt es, einem Betrachter, die Realität mit virtuell erweiterten Inhalten zu erleben. Prinzipiell beschränkt sich AR nicht nur auf die Erweiterung der visuellen Wahrnehmung, sondern kann auch alle anderen Sinneswahrnehmungen ansprechen und erweitern [5]. Heutzutage aber beschränkt sich AR meist nur auf die Erweiterung der visuellen Wahrnehmung.

Das „Milgram Reality-Virtuality Continuum“ welches ein Spektrum zwischen der realen und der virtuellen Umgebung definiert, ordnet AR und Augmented Virtuality (AV) zwischen der realen Umgebung (real environment) und einer rein virtuellen Umgebung ein [6].

Wie in Abbildung 2.1 zu sehen ist, befindet sich AR näher an der realen Umgebung und AV näher an der virtuellen Umgebung. In einer virtuellen Umgebung, welches heute als Virtual Reality (VR) bezeichnet wird, taucht der Nutzer komplett in eine virtuelle Welt ein und nimmt die reale Welt nicht mehr wahr. Dagegen versucht AR nicht die reale Welt zu ersetzen, sondern diese mit Computer-generierten 3D-Inhalten zu vermischen. [7]

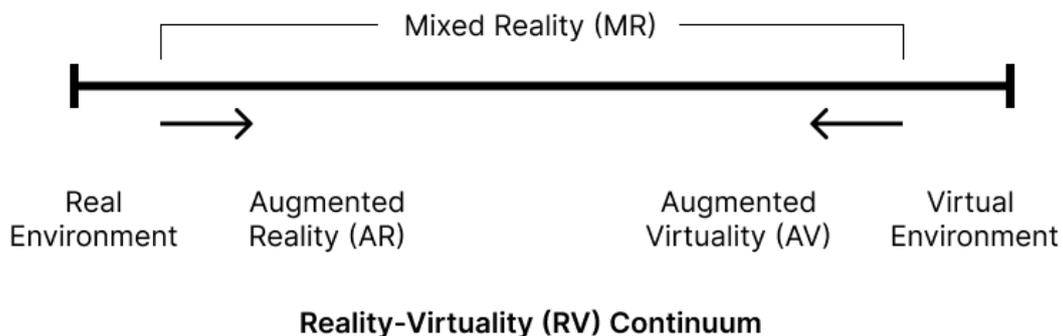


Abbildung 2.1 – Das Reality-Virtuality Continuum nach Milgram et. al [6]

Quelle: In Anlehnung an [6, Abb. 1]

Die Definition von AR, die von Azuma R. [8] aufgestellt wurde und weitestgehend in der Wissenschaft etabliert ist [7], [9], wird von folgenden Eigenschaften geprägt:

- 1) Es kombiniert Realität und Virtualität,
- 2) Es ist interaktiv in Echtzeit
- 3) Virtuelle Inhalte sind im 3D-Raum registriert

Die Vermischung von Realität und Virtualität wird größtenteils mit der Überlagerung von virtuellen Objekten über der Realität erreicht. Das heißt, dass der Nutzer die Realität und die damit vermischten virtuellen Elemente als ein einziges wahrnimmt. Mit diesen virtuellen Inhalten kann der Nutzer des AR-Systems interagieren. Die Registrierung der virtuellen Inhalte in 3D bedeutet, dass ein in AR platziertes

virtuelles Objekt seinen Platz in der realen Umgebung beibehält. Wenn sich die Perspektive des Nutzers ändert – beispielsweise durch die Änderung der Blickrichtung – bleibt das virtuelle Objekt am gleichen Ort bestehen. Das virtuelle Objekt verhält also sich wie ein reales Objekt. [7]

2.2 Tracking

Die kontinuierliche Berechnung der Rotation und Position eines Objektes wird als Tracking bezeichnet. Das Trackingverfahren ist ein grundlegender Baustein für das Funktionieren eines AR-Systems und beeinflusst, wie gut die 3D-Registrierung funktioniert und dadurch auch das Erlebnis des Nutzers in AR. Denn wenn ein AR-Gerät seine aktuelle Position und Rotation in der realen Welt nicht bestimmen kann, kann es dazu kommen, dass die Lage der virtuellen und der realen Welt nicht mehr übereinstimmen. Das hat zu Folge, dass platzierte virtuelle Objekte nicht mehr am Wunschort erscheinen. [7]

Für das Tracking von Betrachter, realen und virtuellen Objekten werden verschiedene Techniken angewendet. Im Folgenden werden die für diese Arbeit relevanten Tracking Methoden beschrieben.

2.2.1 Nicht visuelles Tracking

Abhängig davon, worin die Tracking-Anforderungen der AR Anwendung liegen, werden andere Tracking Methoden eingesetzt. Wird etwa die Position des Nutzers im globalen Kontext benötigt, kann z. B. Mittels GPS (Global-Positioning-System) die Position, mit einer Genauigkeit von ca. 7 m [10] bestimmt werden. Für den Kontext dieser Arbeit ist diese Art von Tracking aufgrund der Ungenauigkeit nicht weiter relevant. [7]

Eine wichtige Art des Trackings, die in vielen AR-Systemen benutzt wird, ist das Tracking mithilfe von Sensoren, die im Gerät eingebaut sind. Heutzutage sind sehr üblich benutzte Sensoren z. B. Drehratensensoren (Gyrosensoren), Beschleunigungssensoren und Magnetometer. Diese Sensoren werden in einer inertialen Messeinheit (IMU) eingebaut, um die Beschleunigung von Rotationen (Drehratensensor) und lineare Beschleunigung (Beschleunigungssensor) entlang einer Achse zu messen. Anhand dieser Messwerte kann die Position und die Orientierung eines Gerätes berechnet werden. Weil es bei diesen Berechnungen zu Messungenauigkeiten kommt, werden diese Sensoren mit einem Magnetometer ergänzt. Dieser kann mithilfe des Erdmagnetfeldes, auch die Rotation des Gerätes bestimmen und entsprechend Messungenauigkeiten kompensieren. [7]

2.2.2 Visuelles Tracking

Eine andere grundlegende Form des Trackings ist das visuelle Tracking. Mithilfe von Kameras wird die relative Position und Orientierung von Objekten zur Kamera bestimmt. Es kann hier zwischen zwei Technikarten unterscheiden werden.

Die erste Technik ist das Marker-basierte Tracking. Dazu gehören Trackingverfahren, die einen Marker für die Bestimmung von Position und Orientierung der Kamera oder Objekten benutzen. In AR sind diese Marker meistens Schwarz-Weiß-Bilder, wie beispielsweise in Abbildung 2.2 sehen ist. Diese Marker müssen im Vorfeld registriert sein, das bedeutet, dass die Größe und die Muster bekannt sind. Mithilfe von Softwarealgorithmen werden daraus Merkmale mit dem bekannten Bild verglichen. Erkennt die Software das Bild, so kann die Position des Markers bestimmt werden und dadurch die Position der Kamera in Relation zum Marker. Tracking mit diesen Markern hat eine hohe Genauigkeit [11], aber hat das Problem, dass der Bewegungsfreiraum limitiert ist. Denn sobald der Marker nicht im Bild ist, kann die Position des Markers nicht mehr verfolgt werden. [7]



Abbildung 2.2 – Typische Marken, wie sie für kamerabasiertes Tracking eingesetzt werden

Quelle: Abbildung von Virtual und Augmented Reality (VR/AR) [7]

Bei merkmalsbasiertem Tracking werden Merkmale im Kamerabild erfasst und mit bereits bekannten 2D oder 3D Modellen zugeordnet. Mit Merkmalsdetektoren, welche spezielle Algorithmen für das Erkennen und Beschreiben von Merkmalen sind [12], können Merkmale aus dem Kamerabild herausgelesen werden, die für den menschlichen Betrachter nicht direkt ersichtlich sind. Mittels dieser erkannten und erfolgreich zugeordneten Daten kann dann die Position und Orientierung der Kamera berechnet werden [7].

2.3 Displays

AR kann auf verschiedenen Geräten dargestellt werden. Dabei gibt es unterschiedliche Techniken für die Darstellung, die zum Einsatz kommen. In diesem Abschnitt werden zwei übliche Techniken vorgestellt. Zusätzlich werden dazu aktuelle Gerätetypen eingeordnet und beschrieben.

2.3.1 Video-See-Through

Bei der Video-See-Through Technik wird die Umgebung mit einer Videokamera aufgenommen. Das Videobild wird dann mit virtuellen Inhalten überlagert und daraufhin auf ein Display ausgegeben

(Abbildung 2.3) [8]. Hier kann bei der Implementation dieser Technik zwischen zwei Gerätekategorien unterscheiden werden [11].

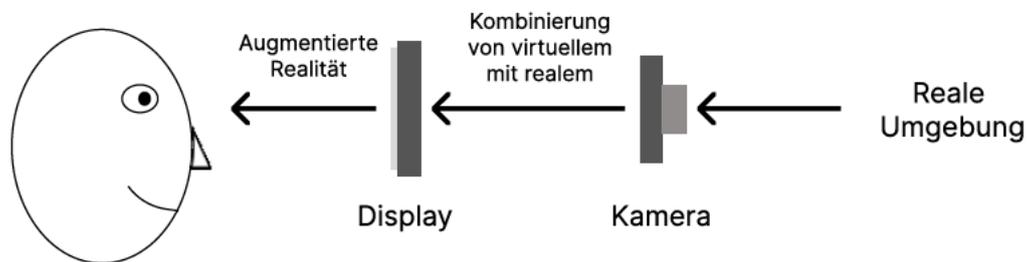


Abbildung 2.3 – Ein Video-See-Through System. Die Umgebung wird von einer Kamera aufgenommen und wird dann, mit virtuellen Inhalten, auf einem Display angezeigt [8].

Quelle: Eigene Darstellung

Zu der ersten Kategorie gehören die Handheldgeräte wie Smartphones und Tablets, die mit ausgestatteten Kameras die Umgebung aufnehmen und zusammen mit den virtuellen Inhalten auf dem Display anzeigen. Hierbei ist die Augmentierung nur zur Position und Orientierung (Rotation) Kamera perspektivisch korrekt. Das heißt das Bild, welches auf dem Display angezeigt wird, ist nicht gleich der Perspektive, die der Nutzer auf die Umgebung hat. Diese Gerätetypen nehmen zusätzlich nur einen minimalen Teil des Blickfeldes des Betrachters ein, welches die Immersion, zusammen mit der vorher beschriebenen Eigenschaft weiter verringert. [7]

Dennoch sind Handheldgeräte heutzutage die meistverbreiteten Geräte für den Einsatz von AR. Dies hat den Grund, da der Besitz Smartphone und Tablets und sehr üblich ist und dass entsprechende AR-Frameworks (ARKit oder ARCore) dafür existieren, die den Einsatz von AR deutlich vereinfachen. [7], [11]

Head-Mounted-Displays (HMD) wie VR-Brillen gehören zur anderen Kategorie. HMDs sind Displays, die direkt vor den Augen des Nutzers getragen werden. Mit diesen VR-Brillen kann der Nutzer die reale Umgebung nicht direkt wahrnehmen, denn nur das von der VR-Brille ausgespielte Bild wird wahrgenommen. Hier wird nämlich eine voll virtuelle Szene dargestellt. Diese Brillen können jedoch auch für den Einsatz von AR verwendet werden. Dabei sieht der Nutzer eine mit virtuellen Inhalten angereichertes Echtzeit Videobild der Umgebung, die mithilfe von integrierten Kameras aufgenommen wird. [7]

Im Vergleich zu AR-Systemen auf Handheldgeräten, bieten HMDs ein immersiveres Erlebnis an. Denn der Nutzer hat ein deutlich größeres Sichtfeld auf die augmentierte Umgebung als bei Handheldgeräten, aufgrund der Nähe zu den Displays. Zusätzlich ist die Perspektive auf die augmentierte Umgebung an die Kopfbewegung des Nutzers gebunden, wie es auch in der Realität teilweise der Fall ist. [11]

2.3.2 Optical-See-Through

Alternativ wird bei HMD-Geräten auch die Optical-See-Through (OST-HMD) Technik verwendet. Mit OST-HMDs wird die reale Umgebung vom Nutzer nicht durch ein ausgegebenes Videobild, sondern direkt wahrgenommen. Der Nutzer schaut in diesem Fall durch ein halb-transparentes und -reflektives Display, worauf die virtuellen Inhalte projiziert werden [8]. Eine Veranschaulichung ist in Abbildung 2.4 zu sehen.

Die Technik erlaubt dem Nutzer die reale Welt mit den eigenen Augen ohne Verzerrungen der Szene zu betrachten. Die Immersion ist bei dieser Display-Art daher am größten, hat jedoch den Nachteil, dass bei einer hellen Umgebung der Hintergrundkontrast sehr gering ist. Die virtuellen Inhalte werden also sehr schwach wahrgenommen, da diese mit steigender Helligkeit zunehmend transparent erscheinen [7]. Dazu kommt das dieser Display-Typ generell ein sehr geringes Sichtfeld hat [13] .

OST-HMD Geräte (oder auch optische AR-Brille) sind im Vergleich zu Handheldgeräten und VR-Brillen, aufgrund dieser genannten Problematiken und unter anderem wegen des hohen Anschaffungspreises, nicht so weit im Konsumentenbereich verbreitet.

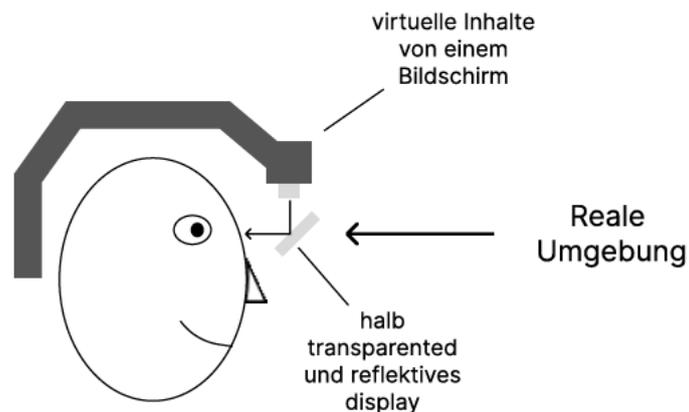


Abbildung 2.4 – Ein Optical-See-Through System in Anlehnung an Azuma R. [8]

Quelle: Eigene Darstellung

2.4 Interaktion in AR

Die Möglichkeit in AR mit virtuellen Elementen interagieren zu können ist, wie oben in Kapitel 2.1 schon beschrieben, ein wesentlicher Bestandteil. Daher werden die Interaktionsmöglichkeiten in AR im Folgenden näher betrachtet.

2.4.1 Interaktionsaufgaben

In AR kann auf verschiedenster Weise mit virtuellen Objekten interagiert werden. Der Nutzer interagiert mit virtuellen Objekten in einer augmentierten Realität, um sie auszuwählen und zu verändern, sowie seine Perspektive darauf zu verändern [7]. Es kann also, bei der Interaktion mit virtuellen Objekten in AR, zwischen drei Grundaufgaben unterscheiden werden: Auswahl, Manipulation und Navigation [2], [14].

Die Navigation, also die Änderung der Position und der Blickrichtung, ist in AR an die Kamera gekoppelt. Für den Nutzer orientiert sich die Wegfindung in AR an der üblichen Wegfindung in der Realität [7]. Das heißt, es können keine speziellen Bewegungstechniken verwendet werden, wie es in VR der Fall ist. Denn in VR kann der Nutzer sich zum Beispiel teleportieren [15], während dieser seine Position in der Realität nicht bewegen muss. In AR hat hingegen eine Änderung der Position in der Realität den gleichen Einfluss wie in der augmentierten Realität. Die Navigation in AR kann dennoch ergänzt werden, um diese zu vereinfachen, beispielsweise mit einem virtuellen Wegweiser, der den Nutzer bei der Wegfindung unterstützt [7].

Die Auswahl ist die Aufgabe, ein bestimmtes Objekt (Einzelauswahl) oder eine Mehrzahl von Objekten (Mehrfachauswahl) aus einer Menge von verfügbaren Objekten zu erfassen oder zu identifizieren [2]. Die Auswahlaufgabe wird im dritten Kapitel ausführlicher beschrieben, da die Mehrfachauswahl ein Teil dieser ist und das Hauptthema dieser Arbeit darstellt.

Die Manipulation eines virtuellen Objektes ist definiert als die interaktive Änderung der Parameter, welche das Objekt charakterisieren. Dazu gehört zum Beispiel die Erscheinung, Größe oder Position und Rotation im 3D-Raum [7]. Wobei die Positionierung, Rotation und Skalierung eine primäre Rolle bei der Manipulationsaufgabe einnehmen und zu den wichtigeren Teilaufgaben gehören [2].

2.4.2 Interaktions-Typen in Handheld-AR

Da diese Arbeit sich mit Mehrfachauswahl-Methoden befasst, die spezifisch für AR mit Handheldgeräten (Handheld-AR) konzipiert sind, ist es sinnvoll über die Interaktionstypen zu reden, die in Handheld-AR zu finden sind.

Bei der Auswahl und Manipulation von virtuellen Objekten in Handheld-AR können die verschiedenen Interaktionsmethoden in verschiedenen Kategorien eingeordnet werden. Diese sind: touchbasierte Interaktion, Mid-Air gestenbasierte Interaktion und gerätebasierte Interaktion. [4]

Mid-Air gestenbasierte Interaktion umfassen alle Techniken, bei denen getrackte Hand- oder Fingergesten für die Eingabe verwendet werden, die die Auswahl und Manipulation von 3D-Objekten ermöglichen. Diese Art von Interaktion wird in dieser Arbeit nicht weiter untersucht, aufgrund der schlechten Möglichkeit auf Tracking von Hand- und Fingergesten in einem Handheldgerät. [4]

Zu den Touch-basierten Interaktionen gehören Interaktion, bei dem Finger-Gesten auf dem Display benutzt werden, um zum Beispiel 3D-Objekte zu Manipulieren oder auszuwählen [4]. Aufgrund der Natur des Displays, ist die Touch-Eingabe auf eine zweidimensionale Ebene limitiert. Die Position der Fingergeste befindet sich also nur innerhalb einer x- und y-Achse und somit kann auf dem Display nur ein Punkt gemessen werden.

Die Kategorie der gerätebasierten Interaktion umfasst alle Interaktionsmethoden, die die physischen Eigenschaften des Gerätes für die Auswahl oder Manipulation von virtuellen Objekten verwenden [4]. Da das Handheldgerät die volle Bewegungsmöglichkeit im Raum hat, können hier die drei Bewegungsrichtungen entlang der x, y und z-Achsen und die drei Rotationen um diese Achsen gemessen werden. Die Anzahl dieser grundlegenden Bewegungsmöglichkeit im 3D-Raum, wird durch die Anzahl der Freiheitsgrade bezeichnet und wird oft mit DOF (degrees of freedom) abgekürzt. Daraus lässt sich verstehen, dass beim Handheldgerät sechs Freiheitsgrade gemessen werden, welches auch als 6DOF Tracking bezeichnet werden kann.

3 Auswahl in 3D

Die Auswahl ist einer der grundlegenden Interaktionsaufgaben in AR [2]. Außerhalb von einem AR-Kontext, wurden Auswahlmethoden in virtuelle Welten schon ergiebig untersucht [3]. Einige schon existierende Auswahlmethoden für 3D lassen sich auch für Handheld-AR übersetzen [3], doch diese sind auf die Auswahl von einzelne Objekten ausgelegt.

Um zu überprüfen, ob Mehrfachauswahl-Methoden in Handheld-AR übersetzt werden können, betrachtet diese Arbeit erstmal, wie die Auswahlaufgabe in 3D aufgebaut ist und wovon sie beeinflusst wird. Dann werden bekannte Auswahl Metaphern behandelt, die die Basis für den Entwurf von Auswahlmethoden stellen können. Schließlich werden bekannte Mehrfachauswahl-Methoden beschrieben.

3.1 Auswahlaufgabe

Die Auswahlaufgabe wird von einigen Einflüssen kontrolliert und diese werden in diesem Abschnitt beschrieben.

3.1.1 Auswahlparameter

Für die Auswahlaufgabe gibt es viele Variablen, welche die Benutzerfreundlichkeit beeinflussen [16]. Zum Beispiel hat die dichte der Objekte in der Umgebung des Zieles oder der Ziele oder die Größe des Objekts Einfluss darauf, welche Interaktionsmethode eingesetzt werden kann. [2]

Die Auswahlaufgabe umfasst einen sogenannten Aufgabenbereich (engl. task space), die mehrere Variationen dieser Aufgabe (Auswahlmethoden) enthält, welche wiederum von Aufgabenparametern (engl. task parameters) definiert sind. Je nachdem welche Parameter für die Auswahl relevant sind, können verschiedene Auswahlmethoden eingesetzt werden.

Dabei sind typische Auswahlparameter, die untersucht wurden, folgende: Entfernung und Richtung zum Ziel, Größe des Ziels, Dichte der Objekte in der Umgebung des Ziels, Anzahl der auszuwählenden Ziele und Verdeckung des Ziels. [2]

3.1.2 Einfluss der Eingabegeräte

Zwischen den Eigenschaften von Eingabegeräten und der Gestaltung von Interaktionsmethoden für die Auswahl besteht ein enger Zusammenhang und die Wahl des Eingabegerätes hat Einfluss darauf, welche Methoden verwendet werden können. So hat die Anzahl der DOF, die ein Gerät kontrollieren kann und die Anzahl der gleichzeitig kontrollierbaren DOF eine wichtige Rolle bei der Auswahlaufgabe [2].

Bei Eingaben auf einem Touch-Display können gleichzeitig 2DOF zu kontrolliert werden. Vergleichen kann man diese Eingaben-Art, mit einer Computermaus, welche auch nur 2DOF gleichzeitig

kontrollieren kann. Bei AR mit einem Touch-Display (Handheld-AR) könnten hier also auch traditionelle Auswahl-Methoden, wie die bekannte Klick Auswahl auf dem Desktop verwendet werden.

Während diese Art von 2D-Eingabe in einem 2D-Kontext, also auf einer grafischen Oberfläche, schon ausgiebig ausgetestet und auch standardisierter sind [7], ist eine solche Eingabe in einem 3D-Kontext mit anderen Schwierigkeiten behaftet. So kann es zum Beispiel sein, dass Objekte zu klein, zu weit weg oder von anderen Objekten verdeckt sind (siehe 3.1.1). Wie gut solch eine Auswahl in 3D funktioniert, hängt ganz davon ab wie diese Objekte im Raum stehen.

Bei gerätebasierten Eingaben (siehe 2.4.2) besteht die Möglichkeit 6DOF gleichzeitig zu kontrollieren. Viele Manipulationsmethoden und darunter auch Auswahlmethoden in virtuellen Welten benutzen Eingabegeräte, welche diese gleichzeitige Kontrolle voraussetzen. [2] In Abschnitt 3.2 werden einige Auswahlmethoden beschrieben, die auch ein 6DOF-Tracking voraussetzen.

3.1.3 Klassifizierung per Taxonomie

Es ist wichtig zu verstehen, wie Auswahl-Methoden in 3D aufgebaut sind, wenn diese genauer betrachtet und weiter entwickelt werden sollen [17], [18]. Dabei können die Auswahlmethoden in einzelne Teilaufgaben unterteilt werden [2], [17].

Es existieren einige Arbeiten, die versuchen Interaktionsmethoden per Unterteilung in kleine Teilaufgaben zu klassifizieren [18]–[20], um die Entwurfsphase von diesen zu vereinfachen beziehungsweise zu verbessern. Diese Unterteilung und Kategorisierung der Interaktionsmethoden wird als Taxonomie bezeichnet. Mithilfe dieser Taxonomien können die Möglichkeiten der Interaktionsmethoden untersucht werden [18]. Es werden dabei verschiedene Ansätze benutzt und es gibt kein einziges Schema, welches universell für alle Fälle benutzt werden kann.

Da diese Arbeit sich um die Mehrfachauswahl beschäftigt, benutzt diese Arbeit die Taxonomie von Lucas J. [17] (siehe Abbildung 3.1) welcher für Mehrfachauswahl von Objekten in einem 3D-Raum konzipiert ist. Mehrfachauswahl-Methoden für können darin eingeordnet, um ein Vergleich dieser Methoden einfacher zu gestalten. Diese Arbeit benutzt die gleiche Taxonomie (siehe Abbildung 3.1), um die Mehrfachauswahl-Methoden für Handheld-AR einordnen zu können und ein Vergleich dieser Methoden einfacher zu gestalten.

Ein wichtiger Bestandteil der Mehrfachauswahl für 3D ist das Anzeigeobjekt [17], welches vom Nutzer bedient wird, um den Auswahlbereich zu zeigen. Die Taxonomie zeigt unter anderem an welche Eigenschaften das Anzeigeobjekt (engl. indication object) hat.

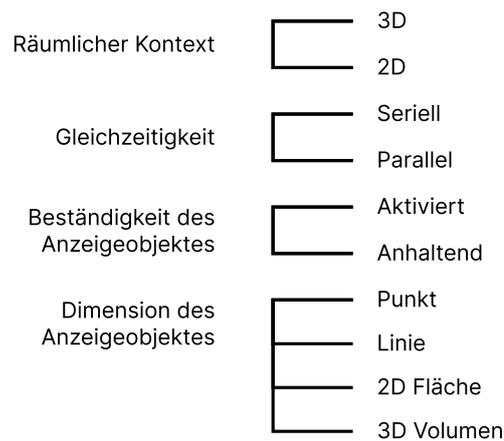


Abbildung 3.1 – Die Taxonomie für Mehrfachauswahl-Methoden in 3D von Lucas J. [17].

Quelle: Eigene Darstellung

Für diese Arbeit sind nur die ersten vier Punkte relevant und werden im Folgenden kurz erklärt:

- Der *räumliche Kontext* bezieht sich auf die Dimensionalität der Eingabe, welche für die Bestimmung des Anzeigobjektes eingesetzt wird. Im 3D Kontext muss das Anzeigobjekt, im 3D-Raum positioniert werden, um eine Auswahl zu ermöglichen und im 2D-Kontext wird das Anzeigobjekt auf einer Bildebene definiert.
- Die *Gleichzeitigkeit* beschreibt, ob eine Mehrfachauswahl nacheinander oder gleichzeitig durchgeführt. Dies wird im weiteren Verlauf genauer beschrieben.
- Anzeigobjekte können vom Nutzer, wie die Rechteck-Auswahl aus einer Desktop-Umgebung, *aktiviert* werden oder wie ein Mauszeiger auf einer Bildebene *dauerhaft* bestehen. Dies wird von der Beständigkeit des Anzeigobjektes beschrieben.
- Die *Dimension des Anzeigobjektes* beschreibt die Form des Anzeigobjektes

3.2 Auswahlmetapher

Wichtig für den Entwurf von Auswahlmethoden ist die ist Auswahl und eine gezielte Nutzung von Metaphern. Geeignete Metapher sind vor allem für 3D Interaktionen sehr wichtig [21]. Nutzer und Entwickler, sollen anhand dieser Metaphern verstehen können, was bei der Nutzung einer Interaktionsmethode möglich ist und wie diese Methode beschränkt ist [2]. Es entsteht also ein Mentales Modell, worauf ein Nutzer zugreifen kann, um eine Interaktionsmethode besser verstehen zu können [21].

Viele Auswahlmethoden basieren auf einigen grundlegenden Metaphern oder eine Kombination dieser Metaphern. So können neue Auswahlmethoden als eine neue Implementation dieser Auswahlmetaphern bezeichnet werden. [2] Einige grundlegende Metaphern, können als Basis für Mehrfachauswahl-Methoden in Handheld-AR benutzt werden. Deshalb werden in diesem Teil einige Metaphern, zusammen mit Beispielen vorgestellt.

3.2.1 Zeige-Methoden

Es gibt viele Auswahlmethoden, die dem Nutzer das Auswählen von virtuellen Objekten mit einem Zeigegerät ermöglichen. Mit diesem Zeigegerät, welches in der Regel ein Eingabegerät mit 6DOF Tracking ist, wird das auszuwählende Ziel oder Ziele vom Nutzer anvisiert und ausgewählt. [7]

Als *Ray-Casting* wird eine sehr simple und sehr beliebte Form dieser Zeige-Methode bezeichnet [3]. Inspiriert von einem Laserpointer, wird ein virtueller Strahl vom Zeigegerät in die Richtung geworfen, in die das Zeigegerät orientiert ist [21]. Bei einer Überschneidung des Strahles mit einem virtuellen Objekt kann dann die Auswahl mit einer Eingabe des Nutzers bestätigt werden [2]. Der Schnittpunkt des Strahles wird meist visualisiert, um zu verdeutlichen worauf gezeigt wird. Dies kann durch Hervorhebung des getroffenen 3D-Objektes oder durch die Positionierung eines 3D-Cursors am Schnittpunkt geschehen [7]. Eine Veranschaulichung des Ray-Casting ist in Abbildung 3.2 zu sehen.

Limitiert ist das Ray-Casting nicht nur für die Auswahl, sondern kann auch für die Manipulation von Objekten verwendet werden, indem virtuelle Objekte zum Beispiel entlang des Schnittpunktes positioniert werden. In Handheld-AR lässt sich diese Auswahlmetapher in vielen AR-Anwendungen wiederfinden, um 3D-Objekte im 3D-Raum zu platzieren, wie zum Beispiel bei der „Ikea Place“ App .

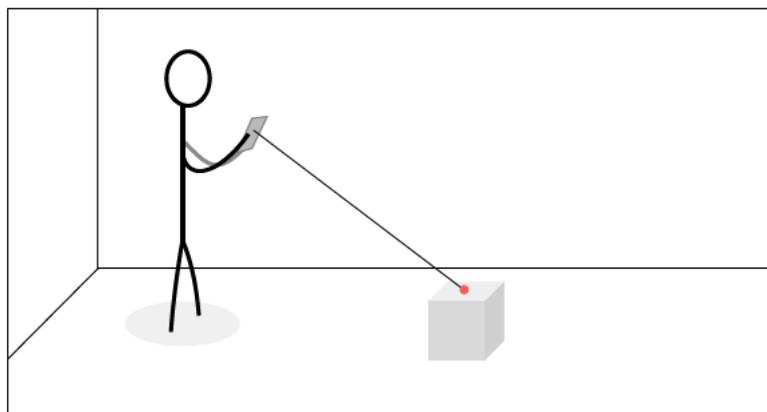


Abbildung 3.2 – Ray-Casting auf einem virtuellen Objekt mit einem Handheld Eingabegerät. Der Schnittpunkt des Vektors wird hier mit einem 3D-Cursor visualisiert.

Quelle: Eigene Darstellung

Weil das simple Ray-Casting nicht perfekt ist und Probleme bei der Auswahl von virtuellen Objekten hat, die zum Beispiel sehr klein oder weit entfernt sind [22], ist es nicht für alle Anwendungsfälle der 3D-Interaktion passend. Deshalb wurden viele verschiedene Auswahl-Methoden entwickelt, welche diese Metapher als Basis für eine verbesserte Auswahl benutzen.

Eine modifizierte Version des Ray-Casting ist das *Sphere-Casting*. Hier wird ein vordefiniertes Volumen beim Schnidepunkt des virtuellen Strahles positioniert. In diesem Fall ist das Volumen eine Kugel. Virtuelle Objekte, die innerhalb dieses Volumen liegen, können als auswählbar betrachtet werden [2]. Die SQUAD Methode [23] benutzt Sphere-Casting um eine initiale Auswahl an Objekten zu erhalten, um daraufhin eine feinere Auswahl zu ermöglichen.

Eine weitere Variante des Ray-Casting ist die *Flashlight Methode*. Anstatt eines virtuellen Strahles wird hier ein Kegel in den 3D-Raum geworfen, welcher mit einer Taschenlampe Vergleichbar ist. Objekte, die innerhalb des Kegels sind, stehen dann zur Auswahl zu Verfügung. [7]

3.2.2 Oberflächenbasierte Methoden

Es gibt auch Auswahlmethoden für Oberflächen, wie auf der Bildebene eines Touch-Displays. Diese Bildebene ist das Bild einer 3D-Szene (bzw. einer augmentierten Realität), die auf das Display projiziert wird [24].

Nutzer können direkt mit der 2D-Bildebene interagieren, um Inhalte zum Beispiel auszuwählen oder zu manipulieren. Aufgrund der verbreiteten Nutzung von Smartphones und Tablets, sind Nutzer sehr erfahren darin Touch-Gesten, für diese direkte Interaktion mit virtuellen Objekten, zu benutzen [2].Nutzer

Zudem ist vielen Nutzern diese Interaktion auf einer 2D-Oberfläche bei Desktopumgebungen bekannt, bei den eine Auswahl, mithilfe der Steuerung des Cursors (dt. Mauszeiger) und einem einfachen Linksklick auf der Computermaus, durchgeführt werden kann. In diesem Kontext ist dies zum Beispiel die Auswahl von Ordnern oder Dokumenten auf dem Desktop. Da diese Desktop-Metapher bei Nutzern bekannt ist und nur zwei DOF kontrolliert werden müssen, erlaubt dieser Interaktionstyp eine einfachere Kontrolle bei der Auswahl [7].

Der Linksklick des Cursors lässt sich durch einfaches Tippen auf dem Touch-Display realisieren. So kann mit dieser Geste, wie auf dem Desktop, eine direkte Auswahl von Inhalten durchgeführt werden. Der Nutzer kann so auf dieser Weise auch mit 3D-Inhalten auf der Bildebene interagieren.

3.3 Mehrfachauswahl-Methoden

Während die oben vorgestellten Auswahlmethoden, für die Auswahl von einzelnen Objekten konzipiert wurden, können diese auch für die Auswahl von mehreren Objekten benutzt werden. Dies ist möglich, indem die Auswahl nacheinander stattfindet [17]. Bei der Auswahl eines Objektes wird dieser in eine

Liste von Auswahlkandidaten gespeichert. Erst wenn der Nutzer die Auswahl abbricht, wird diese Liste gelöscht und alle Objekte werden abgewählt, sonst wird die Liste immer erweitert. Diese Mehrfachauswahl nennt man die *serielle* Auswahl (engl. serial selection)[17]. Vergleichen kann man dies mit dem Halten der Umschalttaste auf der Tastatur, um auf dem Desktop mehrere Objekte nacheinander auszuwählen [2] .

Die gleichzeitige Auswahl von mehreren Objekten, mithilfe von eines Anzeigeobjektes in Form einer Linie, Fläche oder eines Volumens, kann hingegen als *parallele* Mehrfachauswahl bezeichnet werden [17].

3.3.1 Mehrfachauswahl-Methoden im 2D-Kontext

Mehrfachauswahl-Methoden, welche aus Desktopumgebungen stammen und in bekannten 3D-Modellier- oder Bildbearbeitungsprogrammen eingesetzt werden, können für Handheld-AR einfach übersetzt werden. Da ein Handheldgerät eine interaktive 2D Oberfläche besitzt, können solche Auswahlmethoden, einfach übersetzen werden. Denn auf dem Touch-Display kann, wie in einer Desktopumgebung, in einem 2D-Kontext interagiert werden.

Eine typische Mehrfachauswahl-Methode, die bei Desktopumgebungen eine natürliche Rolle spielt, ist die Auswahl mit einem Rechteck [25]. Das Rechteck kann beliebig aufgezogen werden, um eine Menge von Inhalten, wie Dokumenten, parallel auszuwählen. Eingesetzt wird diese Mehrfachauswahl auch in vielen Programmen, wie in Blender, um 3D-Inhalte auszuwählen oder in Photoshop, um Pixelbereiche auszuwählen.

Eine ähnliche Auswahl ist die Mehrfachauswahl mit dem Lasso-Tool, welches üblicherweise in Bildbearbeitungsprogrammen zu finden ist. Hier wird eine geschlossene und frei geformte Kontur gezeichnet, um z. B. ein Abschnitt eines Bildes zu bewegen. Inspiriert von dieser Lasso-Metapher wurde in [17] eine Lasso-Auswahl in einer virtuellen Welt implementiert. In [25] wird die Pinsel-Auswahl, auf Desktopumgebungen beschrieben. Der Nutzer zieht eine vordefinierte Fläche über die Bildebene, um Inhalte auszuwählen. Bei Bildbearbeitungsprogrammen wird diese Pinsel-Metapher für die Auswahl von Bildabschnitten (Pixelbereiche), beim Pinsel-Tool verwendet.

3.3.2 Mehrfachauswahl-Methoden im 3D-Kontext

Einige vorgestellte Auswahlmethoden (Sphere-Casting, Flashlight, Aperture), die eigentlich für die Auswahl von einzelnen Objekten benutzt werden, können auch für die Mehrfachauswahl verwendet werden. Diese benutzen ein Volumen, um eine initiale Menge von Objekten auszuwählen, um dann in einem weiteren Schritt eine verfeinerte Auswahl zu treffen. Diese Methoden können also einfach für die Mehrfachauswahl modifiziert werden, in dem dieser Schritt ausgelassen wird [26]. Es können also alle Objekte in dem Volumen ausgewählt werden, die entweder vollständig vom Volumen umschlossen oder sich auch teilweise mit dem Volumen überlappen [2].

Außerhalb dieser bestehenden Methoden gibt es Mehrfachauswahl-Methoden, die ein Auswahlvolumen dynamisch definieren. Hierbei wird, wie bei dem Lasso-Tool oder die rechteckige Flächenauswahl, für jede Auswahl ein neuer Bereich erstellt. In [17] wird die Lasso-Metapher benutzt, um ein Volumen im 3D-Raum zu erstellen, indem die erstellte 2D Kontur, von der Kamerasicht, in die virtuelle Umgebung projiziert wird.

Ein weiterer Ansatz für die Mehrfachauswahl ist die Auswahl mit einem Volumen, welcher im 3D-Raum dynamisch positioniert werden kann. Zusätzlich kann der Nutzer die Form und Größe dieses Volumens frei bestimmen [2]. Ein Beispiel für diesen Ansatz ist die PORT-Methode [17], in der eine Box frei positioniert wird, indem die Positionen der sechs Seiten unabhängig voneinander bestimmt werden können.

4 Konzeption

Für den Entwurf der Mehrfachauswahl-Methoden in Handheld-AR war besonders der Vergleich des räumlichen Kontextes von Bedeutung, da hier verschiedene Eingabemöglichkeiten vorliegen. Diese Eingabemöglichkeiten erlauben die Auswahl durch Anzeigeobjekte auf der Bildebene des Displays (2D-Kontext) und Anzeigeobjekte im 3D-Raum (3D-Kontext).

Damit auf den ersten Blick deutlich wird, wie sich diese Methoden voneinander unterscheiden, wird hierfür die Taxonomie aus [17] verwendet. Die entworfenen Mehrfachauswahl-Methoden werden darin eingeordnet und die Unterschiede zwischen den Methoden werden hervorgehoben. Obwohl der Schwerpunkt des Entwurfs in seinem räumlichen Kontext lag, wurden die Mehrfachauswahl-Methoden sehr unterschiedlich voneinander entworfen und nehmen deshalb unterschiedliche Positionen in der Taxonomie (siehe Abbildung 3.1) ein. In Tabelle 4.1 kann auf einem Blick betrachtet werden, welche Unterschiede die Mehrfachauswahl-Methoden haben. Es wurden vier Mehrfachauswahl-Methoden für den Vergleich implementiert.

Methode	Räumlicher Kontext	Gleichzeitigkeit	Beständigkeit des Anzeigeobjekts	Dimension des Anzeigeobjekt
Sphere-Select	3D	Parallel	Dauerhaft	3D Volumen
Box-Select	3D	Parallel	Aktiviert	3D Volumen
Rect-Select	2D	Parallel	Aktiviert	2D Bereich
Drag-Select	2D	Seriell	Aktiviert	Punkt

Tabelle 4.1 - Einordnung der Methoden in die Taxonomie für die Mehrfachauswahl-Methoden in 3D

4.1 Entwurfs-Richtlinien

Für den Entwurf der Mehrfachauswahl-Methoden in Handheld-AR, wurden einige Richtlinien aufgestellt, die auf die eigene Motivation und auf die Vorteile des mobilen Kontextes von Handheld-AR basieren.

- 1) Der Ablauf der Mehrfachauswahl-Methoden soll möglichst wenige Schritte beinhalten, um die gewünschte Auswahl zu erreichen. Der Nutzer soll so schnell wie möglich eine Auswahl treffen können.
- 2) Um die Eigenschaft auszunutzen, dass ein Handheldgerät einhändig bedient werden kann, soll die Mehrfachauswahl ebenfalls nur mit einem Finger verwendet werden können. Die Möglichkeit der Multi-Touch Eingabe wird ignoriert.

- 3) Auswahl-Bereich bei parallelen Mehrfachauswahl-Methoden soll dynamisch verändert werden können, um eine präzise als auch eine grobe Auswahl zu ermöglichen.

4.2 Methoden im 3D Kontext

Im Folgenden werden die Mehrfachauswahl-Methoden vorgestellt, dessen Anzeigebjekt im 3D-Raum platziert sind und mit denen im 3D-Raum interagiert wird. Mithilfe des 6DOF-Tracking wird ein virtueller Strahl (Ray-Casting) gesteuert, um das Anzeigebjekt entlang einer erkannten Fläche (siehe 5.1) im 3D-Raum zu positionieren. Hier bestimmt der Schnidepunkt des Strahles mit der Fläche die Position des Anzeigebjektes. Das heißt, die Anzeigebjekte haben ihren Ursprung von dieser Fläche und werden in Relation dazu positioniert und definiert.

4.2.1 Sphere-Select

Die Sphere-Select Methode basiert auf der Sphere-Casting Metapher, welche eine Kugel eine Kugel am Schnidepunkt des virtuellen Strahles positioniert. Diese Implementation benutzt das Kugelvolumen, um alle virtuellen Objekte auszuwählen, welche sich ganz oder teilweise mit ihr überlappen. Es handelt sich hierbei also um eine parallele Mehrfachauswahl.

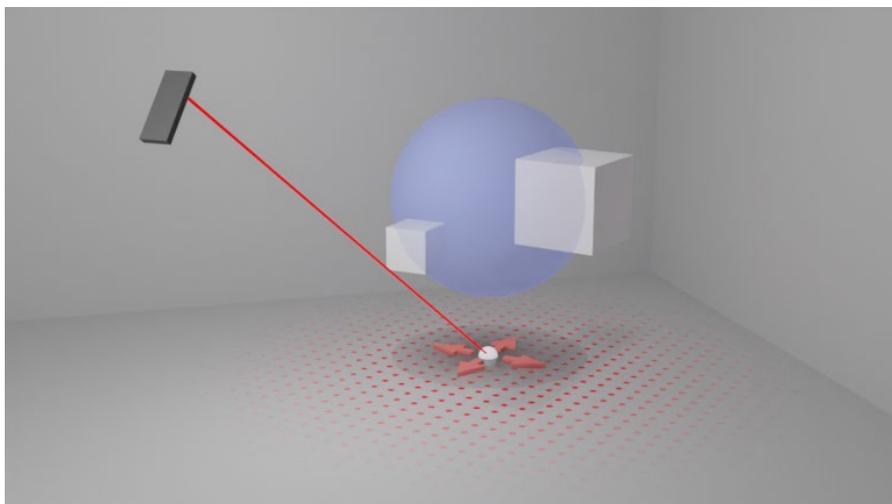


Abbildung 4.1 - Darstellung der Sphere-Cast Methode. Der Schnidepunkt wird mit einem 3D-Cursor dargestellt.

Quelle: Eigene Darstellung

Durch die Eigenschaft des virtuellen Strahles wird die Mehrfachauswahl aus der Ferne ermöglicht. Der Nutzer kann die virtuellen Objekte auswählen, ohne sich viel bewegen zu müssen. Mit einer minimalen Handgelenkbewegungen kann die Position des Anzeigebjektes bzw. der Kugel auf der erkannten Ebene verändert werden.

Die Sphere-Select Methode ermöglicht es außerdem die Höhenposition und die Skalierung des Anzeigeobjektes anzupassen. Statt herkömmliche Bedienelemente, wie Buttons oder Slider, werden hierfür Wisch-Gesten auf dem Touch-Display verwendet. So kann mit einer Wisch-Geste auf der x-Achse des Displays, die Größe der Kugel (siehe Abbildung 4.2a) verändert werden. Die Höhenposition hingegen kann mit einer Wisch-Geste auf der y-Achse variiert werden. Um die Wahrnehmung des Nutzers für die genaue Position der Kugel im 3D-Raum zu erleichtern, wird ein Schatten der Kugel auf dem Boden projiziert.

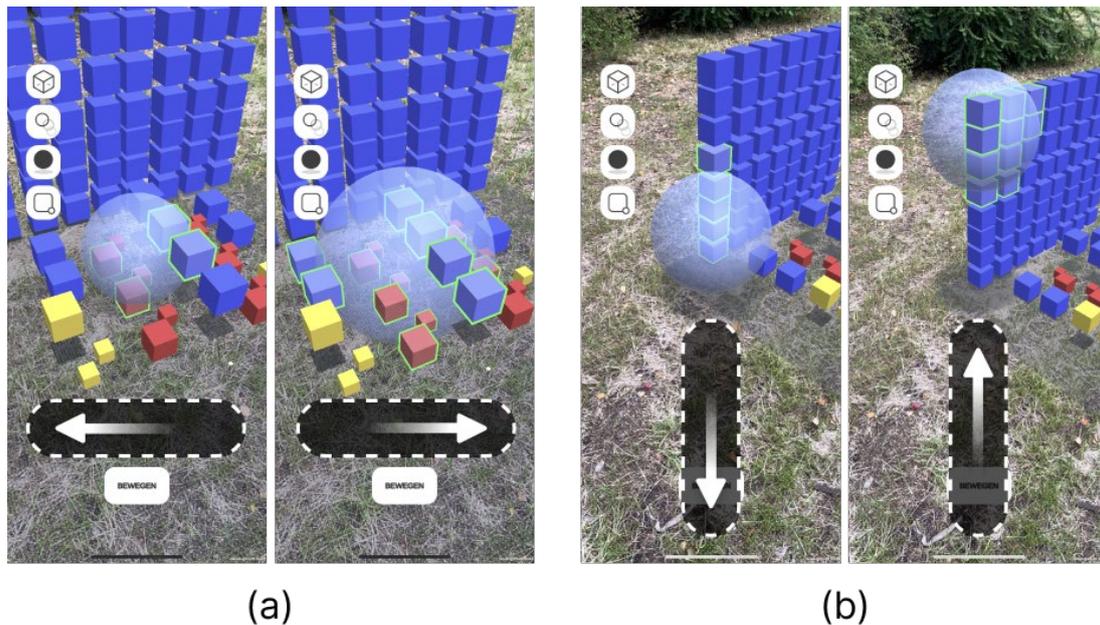


Abbildung 4.2 – (a) Darstellung der Größenänderung der Kugel durch einer Wisch-Geste auf der X-Achse.
 (b) Darstellung der Höhenänderung durch eine Wisch-Geste auf der Y-Achse.

Quelle: Bildschirmaufnahme aus der Anwendung

4.2.2 Box-Select

Die Box-Select Methode ist inspiriert von der Rechteck-Auswahl einer herkömmlichen Desktopumgebung. Einige Eigenschaften dieser Desktop-Auswahl werden von der Box-Select Methode in einem 3D-Kontext übersetzt. Dafür soll auf der erkannten Fläche, ein Quader für aufgezogen werden. Alle virtuellen Objekte, die sich mit diesem Volumen ganz oder teilweise überlappen, werden ausgewählt. Aufgrund dessen liegt hier eine parallele Mehrfachauswahl vor. Da die Auswahlmethode eine höhere Dimensionalität als das Rechteck der Desktop-Methode hat, muss das Aufziehen des Quaders in mehreren Schritten erfolgen. Die Box-Select Methode wechselt dafür zwischen drei Modi: Im ersten Modus (siehe Abbildung 4.3) wird ein Eckpunkt, auf der erkannten Fläche, positioniert. Solange dieser Modus aktiv ist, wird der Eckpunkt kontinuierlich am Schneidepunkt des virtuellen Strahles ausgerichtet. Das heißt es wird hierfür keine weitere Eingabe auf dem Touch-Display benötigt.

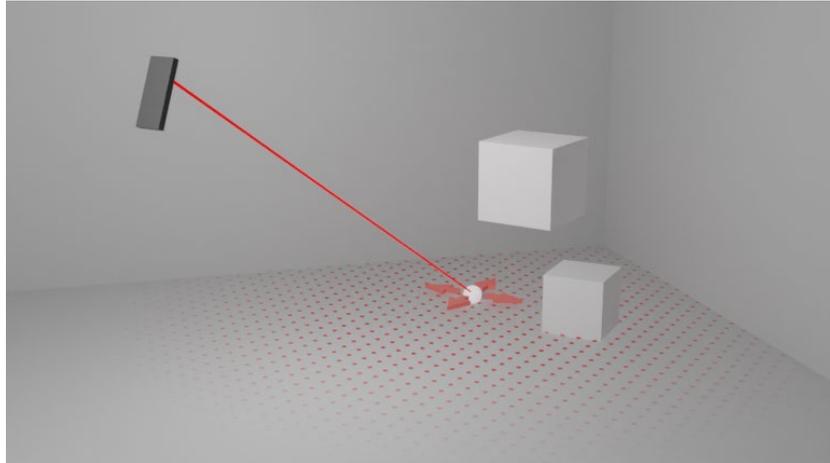


Abbildung 4.3 – Eine abstrakte Darstellung vom ersten Modus, mit Ausschnitt der Benutzeroberfläche. Die erkannte Fläche wird mit einer rot gepunkteten Ebene visualisiert.

Quelle: Eigene Darstellung

Im zweiten Modus wird zunächst die Bodenfläche des dreidimensionalen Anzeigeobjektes, durch das Bestimmen von zwei sich diagonal gegenüberliegenden Eckpunkten definiert. Mit einer halte Halte-Geste auf dem in Abbildung 4.6 dargestellten „Auswahl“-Button, wird der zweite Modus aktiviert. Die Position des ersten Eckpunktes wird bestätigt und die Positionierung des zweiten Eckpunktes beginnt auf gleicherweise wie im ersten Modus. Erreicht die durch die Verschiebung des zweiten Eckpunktes entstehende Bodenfläche des Quaders die gewünschte Größe, kann die Halte-Geste beendet werden. Die Orientierung dieser Fläche, wird beim Setzen des ersten Eckpunktes einmalig bestimmt. Solange die Halte-Geste besteht, bleibt dieser Modus aktiv.

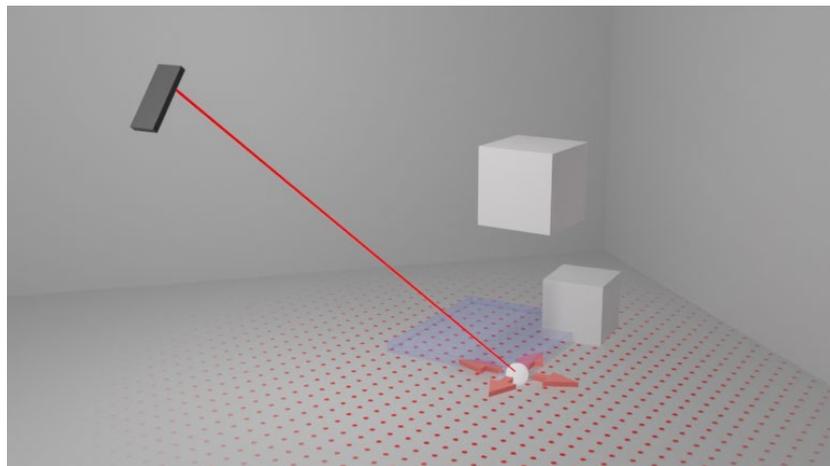


Abbildung 4.4 – Abstrakte Darstellung des zweiten Modus von der Box-Select Methode

Quelle: Eigene Darstellung

Der dritte Modus (siehe Abbildung 4.5) wird aktiviert, sobald die Halte-Geste auf dem „Auswahl“-Button beendet wird. Nun wird durch die Definition des dritten Eckpunktes die Höhe des Quaders bestimmt. Für die Bestimmung der Höhe wird der Schneidepunkt des virtuellen Strahles entlang einer unsichtbaren Ebene benutzt. Diese Ebene steht, ausgehend vom zweiten Eckpunkt, senkrecht zu der

erkannten Bodenfläche und bleibt stets zu Kamera hingerrichtet. So legt die Höhe des Schneidepunktes, die Höhe des Quaders fest. In Abbildung 4.5 ist diese senkrechte Ebene durch eine rot gepunktete Fläche visualisiert.

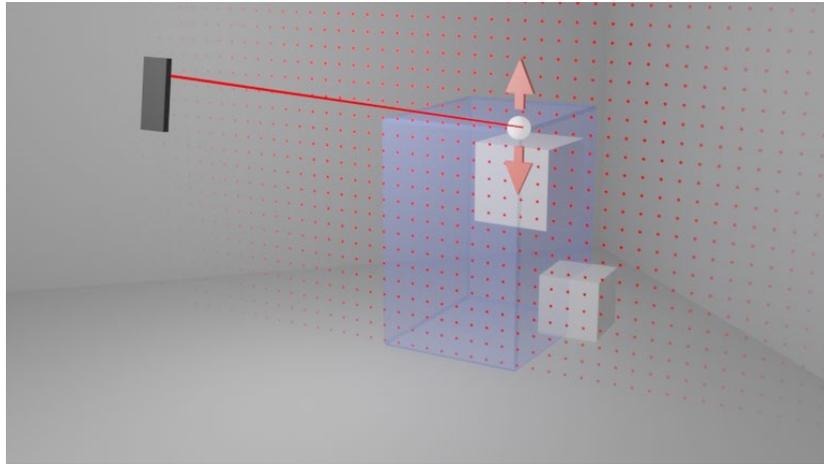


Abbildung 4.5 – Abstrakte Darstellung des dritten Modus der Box-Select Methode

Quelle: Eigene Darstellung

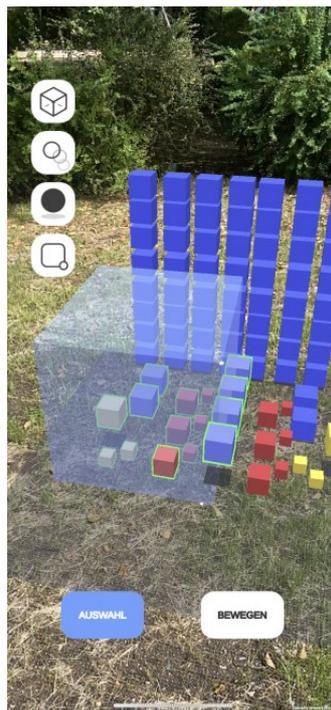


Abbildung 4.6 – Bildschirmaufnahme der Box-Select Methode mit „Auswahl“-Button (unten links) und „Beweg“-Button (unten rechts)

Quelle: Eigene Darstellung

Abschließend wird die Auswahl durch eine einzelne Eingabe auf dem Auswahl-B bestätigt. Die Auswahl-Methode geht dann wieder in den ersten Modus über. 3D-Objekte, die innerhalb des Quaders lagen, werden als ausgewählt markiert. Wie die Rechteck-Auswahl bei Desktopumgebungen, geht die Auswahl der virtuellen Objekte verloren, sobald eine neue Auswahl-Eingabe durchgeführt wird.

Zuletzt kann während jedem Schritt dieser Methode zum ersten Modus zurück gewechselt werden, wenn eine Tipp-Geste auf dem Display durchgeführt wird.

4.3 Methoden im 2D-Kontext

In diesem Teil werden Methoden vorgestellt, die das Touch-Display als Eingabe für die Mehrfachauswahl benutzen und das Anzeigeobjekt auf der Bildebene definieren.

4.3.1 Drag-Select

Die Drag-Select Methode, ist das Äquivalent zu einem Pinsel-Tool aus Bildbearbeitungsprogrammen, wie Adobe Photoshop [27]. Der Nutzer interagiert durch einer Wisch-Geste mit dem Touch-Display mit der Bildebene. Somit kann auf er direktem Wege virtuelle Objekte im 3D-Raum auswählen. Die Objekte werden in Mehrfachauswahl-Methode nacheinander ausgewählt, weshalb von einer seriellen Mehrfachauswahl gesprochen werden kann. Um eine neue Auswahl zu ermöglichen, kann die bestehende Auswahl der virtuellen Objekte, mit einer Tipp-Geste auf einer beliebigen Stelle des Displays, zurückgesetzt werden. In Abbildung 4.7 ist eine Darstellung dieser Methode zu sehen. Der Einbezug dieser Mehrfachauswahl-Methode in den Vergleich, ermöglicht den Vorteil von parallelen über seriellen Mehrfachauswahl-Methoden zu überprüfen.

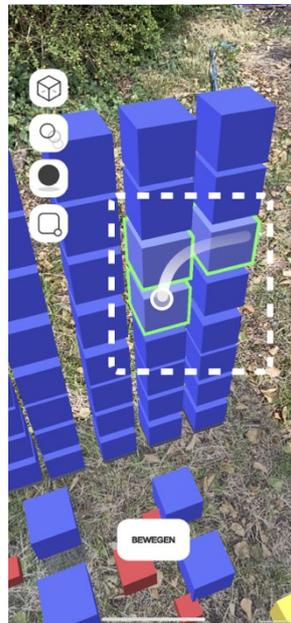


Abbildung 4.7 – Auswahl von 3D-Objekten mit der Drag-Select Methode. Innerhalb der weißen Umrandung wird die Wisch-Geste visualisiert.

4.3.2 Rect-Select

Die Rect-Select Methode ist eine direkte Implementierung der bekannten Rechteck-Auswahl aus Desktopumgebungen. Das Anzeigeobjekt dieser Methode verhält sich wie die Rechteck-Auswahl und wird auf ähnlicher Weise definiert.

Mit einer Wisch-Geste auf dem Touch-Display, wird ein rechteckiger Bereich auf der Bildebene aufgezogen. Der erste Berührungspunkt bestimmt die Position des ersten Eckpunktes. Der zweite Eckpunkt liegt unter der aktuellen Position der Touch-Eingabe.

Virtuelle Objekte, die auf der Bildebene innerhalb des Rechtecks liegen, zählen als Auswahlkandidaten. Sobald die Wisch-Geste beendet wird, beziehungsweise wenn der Finger gehoben wird, werden die Kandidaten als ausgewählt markiert. Durch eine Tipp-Geste auf dem Touch-Display oder einer neuen Auswahl wird die vorherige Auswahl aufgehoben. Da hier mehrere Objekte gleichzeitig ausgewählt werden, ist diese Methode als eine parallele Mehrfachauswahl zu definieren.

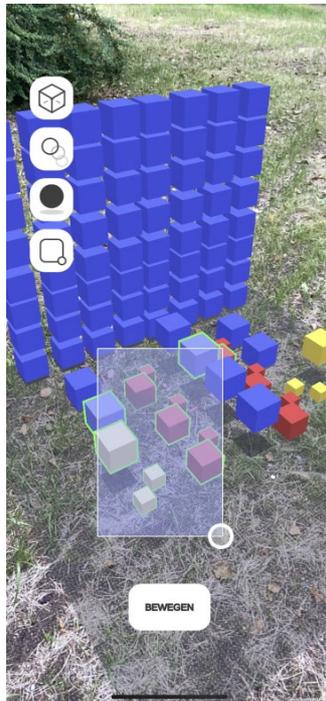


Abbildung 4.8 – Auswahl von 3D-Objekten mit der Rect-Select Methode. Der Halte-Geste wird mit einem Kreis visualisiert.

Quelle: Eigene Darstellung

4.4 Manipulation der Objekte

Nach der erfolgreichen Auswahl kann mit den Objekten weiter interagiert werden. Für weitere Testzwecke ist es möglich, die Position dieser Objekte zu verändern. Dies wird mithilfe der Ray-Casting Metapher umgesetzt. Die Objekte werden in Relation zum Schneidepunkt des virtuellen Strahles verankert: Wenn der Schneidepunkt sich einen Meter in eine bestimmte Richtung bewegt, so bewegen sich alle ausgewählten Objekte, um die gleiche Distanz und dieselbe Richtung. Dadurch, dass die Änderung der Position des Schneidepunktes immer auf der erkannten Oberfläche positioniert ist, gibt es keine Änderung in der Höhenposition. Die neue Positionierung wird bei allen Methoden mit einer Halte-Geste auf einem „Beweg“-Button aktiviert.

5 Implementierung

Dieses Kapitel beschreibt die Implementierung der Mehrfachauswahl-Methoden in Handheld-AR und geht auch auf einige Grundlagen ein.

5.1 AR-Framework

Für die Entwicklung von AR-Anwendungen in Handheld-AR stellen die großen Plattformen, wie iOS und Android, AR-Frameworks zur Verfügung. Mit diesen wird die Vermischung der realen Umgebung mit virtuellen Inhalten ermöglicht.

Wie bereits beschrieben, können für die kontinuierliche Bestimmung der Position und der Orientierung eines Gerätes in Relation zu einem bekannten Punkt, verschiedene Tracking-Verfahren verwendet werden (siehe 2.2). Die Frameworks von Android (ARKit) und iOS (ARCore) benutzen sogenannte SLAM (simultaneous localization and mapping) Systeme [28], [29], welches durch eine Kombination von inertialen und visuellen Tracking-Systemen umgesetzt wird.

Beim SLAM-Verfahren wird die Umgebung in Form einer Karte rekonstruiert, während gleichzeitig die Position und Orientierung des Gerätes darin eingeschätzt wird. Hierfür werden graduell erkannte Merkmale aus dem Kamerabild, mit vorher erkannten Merkmalen aus der Karte, abgeglichen und darin zugeordnet. Auf Grundlage der bereits bekannten Merkmale in der Karte, werden gleichzeitig die Position und die Orientierung der Kamera neu bestimmt. [7]

Mithilfe dieser Merkmale können, diese Frameworks, Oberflächen im Raum erkennen und diese in einer 3D-Szene als geometrische Ebenen visualisieren. Diese Flächen werden, in der für diese Arbeit programmierten AR-Anwendung, für die Verankerung der 3D-Szene in der realen Umgebung verwendet.

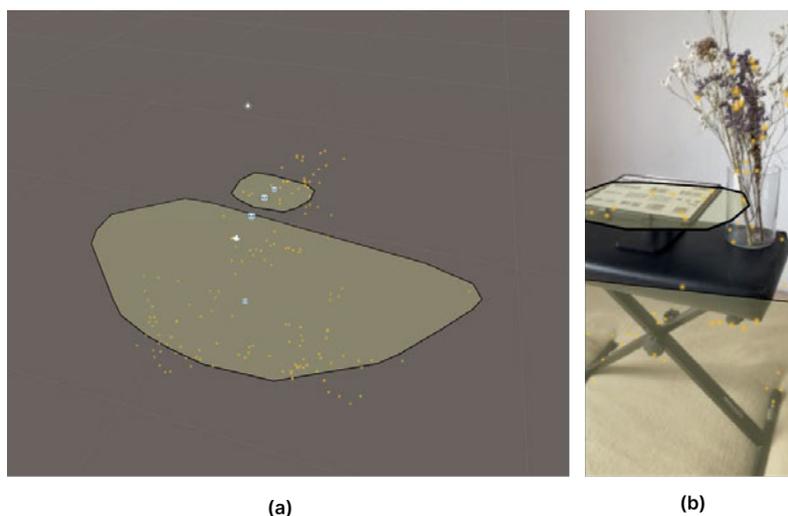


Abbildung 5.1 - Darstellung der erkannten Merkmale und Ebenen in der 3D-Szene (a) Darstellung (b) zeigt diese in Bezug zum realen Raum.

Quelle: Eigene Darstellung

5.2 Entwicklungsumgebung

Als Entwicklungsumgebung wurde die Spiele-Engine „Unity“ [30] verwendet, da sie viele Schnittstellen für die Entwicklung von AR-Anwendungen besitzt.

Das von Unity Technologies angebotene „ARFoundation“ Paket bietet Zugriff auf die wichtigsten Features, von AR-Frameworks der großen mobilen Plattformen (iOS und Android), mit einer einzigen Softwareschnittstelle an [31]. Dadurch kann die Entwicklung für viele Plattformen gleichzeitig durchgeführt werden, ohne auf plattform-spezifische Implementierungen der AR-Frameworks zu achten.

Außerdem stehen in einem sogenannten „Asset-Store“ für Unity zahlreiche Plugins online frei zur Verfügung, welche die Entwicklung von Anwendungen erleichtern. Das LeanTeach+ Plugin [32] ist eine in diesem Store angebotene Script-Sammlung, welche die Implementation der Erkennung von Touch-Gesten anbietet.

Um die Entwicklungszeit zu verkürzen, wurde das „ARFoundation Editor Remote“ [33] Plugin verwendet. Dieses Plugin erlaubt ein „In-Editor-Testing“ von AR-Anwendungen, indem ein Smartphone, die Engine mit Kamerabild- und Trackinginformationen zuführt. Es muss also kein Build der Anwendung erstellt werden, um diese zu testen, welchen zu einer deutlich verkürzten Iterationszeit führt.

5.3 Anwendung

Dieser Abschnitt beschreibt die Implementation der konzipierten Mehrfachauswahl-Methoden für Handheld-AR.

5.3.1 Platzierung der Szene

Die Grundvoraussetzung für die Anwendung der Mehrfachauswahl-Methoden in einem 3D-Raum ist das Erkennen einer Ebene, also einer horizontalen Fläche, in dem gewünschten Raum. ARFoundation erstellt, basierend auf den gefundenen Oberflächen durch das Tracking der AR-Provider (ARKit oder ARCore), Planes (flache Ebenen) in der 3D-Szene. Mit diesen Plane-Objekten kann dann weiter interagiert werden [31].

Es wird die Ray-Casting Implementation von Unity benutzt, um ein „Anchor-GameObject“ am Schnittpunkt des virtuellen Strahles, mit einem dieser Plane-Objekten, zu platzieren. Ein Anker-Objekt markiert einen gewünschten Punkt im 3D-Raum, welcher kontinuierlich getrackt wird. In diesem Anker-Objekt, werden alle 3D-Objekte hinterlegt, die ihre Position im 3D-Raum beibehalten sollen. Um die erkannte Bodenfläche zu visualisieren, wird darin unter anderem ein *10x10m* große Ebene darin hinterlegt.

Dazu gehört auch eine Plane, die zehn mal zehn Meter groß ist. Diese flache Ebene ist transparent und weiß gepunktet, um die Bodenfläche zu visualisieren (siehe Abbildung 5.2).

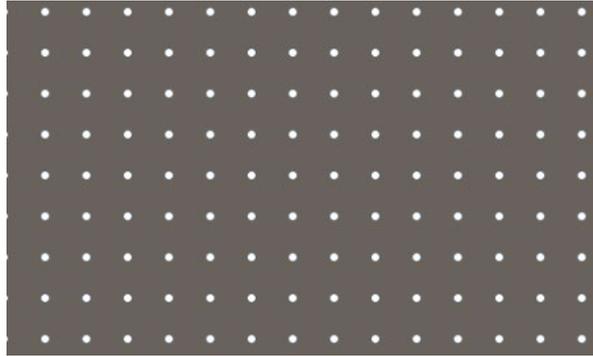


Abbildung 5.2 - Darstellung der transparent und weiß gepunkteten Fläche.

Quelle: Eigene Darstellung

5.3.2 Ray-Casting

Für die Positionierung der 3D-Anzeigeobjekte wurde ein „RayCast“ Script erstellt. Hier wird ein Vektor (virtueller Strahl), von der Mitte des Kamerabildes, gerade in dem Raum projiziert. Schneidet sich dieser Vektor mit einer Oberfläche, wird ein „Event“ ausgelöst, welches die Position des Schnidepunktes ausgibt. Scripts, welche die Logik der Mehrfachauswahl-Methoden beinhalten und diese Position benötigen, können dieses „Event“ abonnieren. Das heißt, sobald eine Oberfläche getroffen wird, werden alle Abonnenten des Events darüber informiert.

5.3.3 Implementierung der Mehrfachauswahl-Methoden

5.3.3.1 Sphere-Select

Bei der Sphere-Select Methode wird, am Schnidepunkt des virtuellen Strahles, das Anzeigeobjekt und ein Pseudo-Schatten fortwährend platziert. Um den Pseudo-Schatten zu erhalten, wurde ein Bild eines Kreises als Material Farbe einer Plane gesetzt.

Die Touch-Gesten für die Skalierung der Kugel und Änderung der Höhenposition werden mit dem LeanTouch+ Plugin implementiert. Die Änderungsraten können wie folgt zusammengefasst werden: Die Distanz des Berührungspunktes in Pixel seit dem letzten Frame-Update wird mit der kürzesten Bildschirmseite in Pixel multipliziert. Dieser wird nochmals mit 0.7 multipliziert. Das Ergebnis ist die Änderungsrate der Größen- und Höhenparameter des Anzeigeobjekts pro Frame.

Die Kugel hat zu Beginn einen Durchmesser von 10cm Abstand. Um der Kugel eine Halbtransparente Oberfläche zu geben wurde der Alphawert des Materials auf 0.5 gesetzt. Der 2D Schatten ist immer am Schnidepunkt positioniert und hat immer den gleichen Durchmesser wie die Kugel.

Unity bietet Funktionen an, die die Kollision zwischen zwei Objekten mit „Collider“ Komponenten testet. Da für die 3D-Anzeigeobjekte nur die primitiven Formen von Unity verwendet werden, haben die „Collider“ die gleiche Position, Form und Größe, wie das 3D Model. Wird eine Kollision mit einem

auswählbaren Objekt erkannt, wird das jeweilige Objekt durch einen Materialwechsel in Form einer grünen Umrandung gekennzeichnet. Verlässt das Objekt den „Collider“ wird wieder das originale Material übernommen.

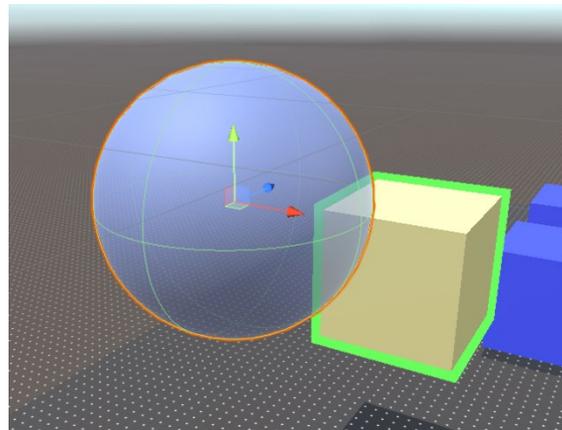


Abbildung 5.3 – Kollision des Anzeigeobjektes der Sphere-Select Methode mit einem auswählbaren 3D-Objekt.

Quelle: Eigene Darstellung

5.3.3.2 Box-Select

Für das Aufziehen des Quaders wird ein normalisierter 3D-Cube verwendet. Das heißt der Ursprungspunkt wurde um den Vektor $(-0.5, -0.5, -0.5)$ verschoben und die Dimensionen auf 1m gesetzt. (siehe Abbildung 5.4). Das hat zur Folge, dass nicht der Mittelpunkt (originaler Ursprungspunkt) des 3D-Objektes am Schnittpunkt des virtuellen Strahles positioniert wird, sondern der neue Ursprungspunkt. Der als erster bestimmte Eckpunkt entspricht aufgrund dessen stets der Position des Ursprungs.

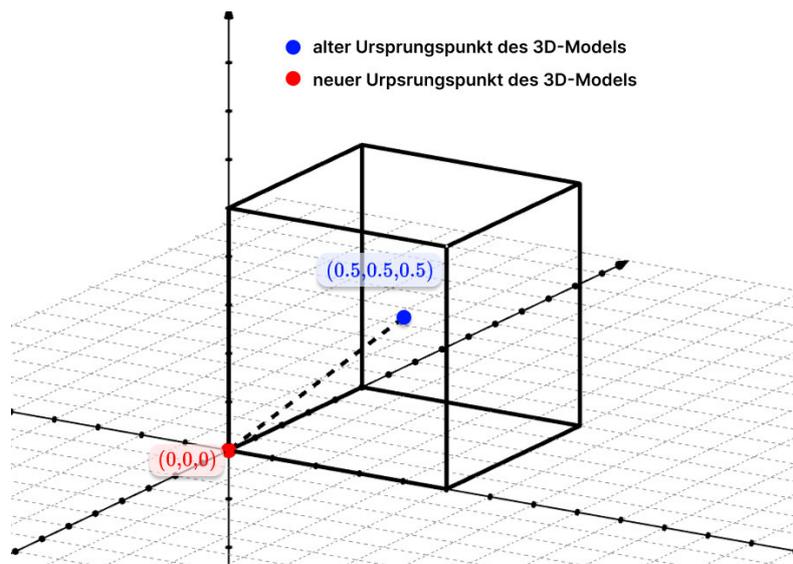


Abbildung 5.4 – Normalisierter Kubus in einem Koordinatensystem. Der Ursprungspunkt des Koordinatensystems ist am Schnittpunkt des virtuellen Strahles.

Quelle: Eigene Darstellung

Mithilfe des Richtungsvektors, vom gesetzten Eckpunkt-1 (E1) zum Eckpunkt-2 (E2), kann die gewünschte Skalierung des Quaders bestimmt werden. Da hier ein normalisierter Kubus vorliegt, kann dieser Richtungsvektor als Skalierungsfaktor verwendet werden, um die korrekten Dimensionen zu erhalten. In Abbildung 5.5 wird dies genauer veranschaulicht. Damit diese Skalierung, nach der Rotation des aufgezogenen Volumens, funktionieren kann, muss der zweite Eckpunkt (E2), um die gleiche Rotation (mit E1 als Drehpunkt), wieder zurückrotiert werden. Dieser Eckpunkt wird daraufhin für die Berechnung des Richtungsvektors und somit des Skalierungsfaktors verwendet.

Das Material des Volumens ist gleich die der Sphere-Select Methode. Die Kollision wird auf gleicherweise implementiert.

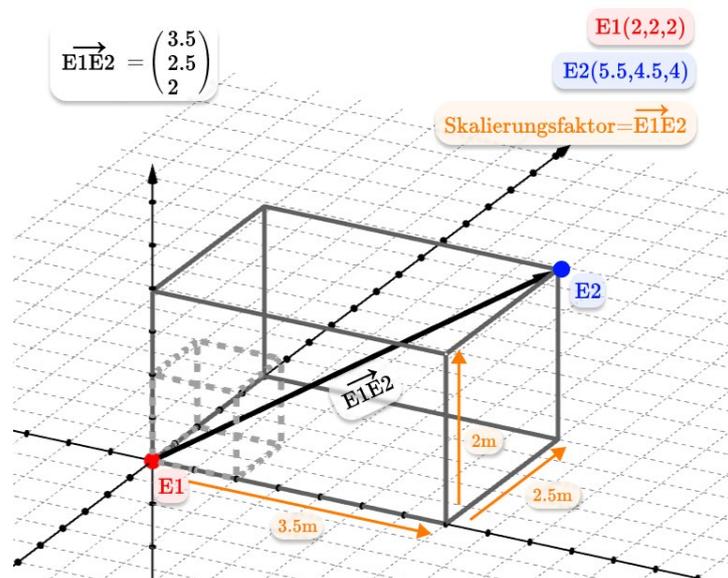


Abbildung 5.5 –Skalierung des Kubus in einem Koordinatensystem. Der normalisierte Kubus wird durch die Veränderung von E2 bis zu der gewünschten Größe skaliert.

Quelle: Eigene Darstellung

5.3.3.3 Rect-Select

Für die Implementierung der Methode wurden große Teile des Codes von einem Blog-Post übernommen [34]. Hier wird eine Grafik in Form eines Rechtecks auf der Bildebene definiert und mit der „Viewport-Point“ Position der 3D-Objekte abgeglichen. Ist ein Punkt innerhalb der der Grafik, gilt es als Auswahlkandidat. Sobald die „Viewport-Point“ Position des 3D-Objektes den Bereich verlässt, wird es abgewählt. Mit dem Beenden der Eingabe wird das Grafik-Element gelöscht. Die Auswahl besteht fort und wird erst gelöscht, sobald eine neue Eingabe gestartet wird.

Das rechteckige Grafik-Element hat einen niedrigen Alphawert von 0.3, um das Blockieren größerer Flächen der 3D-Szene zu vermeiden.

5.3.3.4 Drag-Select

Die Drag-Select Methode wurde ausschließlich mithilfe der LeanTouch+ [32] Script Sammlung implementiert. Das Erkennen einer Wisch-Geste führt zu Auswahl aller sich unterhalb der Berührungspunkte befindender Objekte. Um die Auswahl zu erhalten, wird mit der Raycast-Methode von Unity ein virtueller Strahl in den Raum projiziert, um einen Treffer mit auswählbaren Objekten zu erkennen. Der virtuelle Strahl hat seinen Ursprung am aktuellen Berührungspunkt der Wisch-Geste und wird in die Richtung projiziert, in der die Kamera schaut.

6 Nutzerstudie

Anhand einer Nutzerstudie werden die in Handheld-AR implementierten Mehrfachauswahl-Methoden miteinander verglichen. In einer Aufgabenstellung wird überprüft, wie sich der 2D-Kontext und der 3D-Kontext der Methoden auf die Benutzerfreundlichkeit auswirken. Darüber hinaus wird getestet, wie die Mehrfachauswahl-Methoden in verschiedenen Szenarien abschneiden.

6.1 Teilnehmer

An dieser Studie haben zehn Personen (neun männlich, eine weiblich, Alter 23-39, Mittelwert = 27,6) teilgenommen. Neun Teilnehmer haben angegeben, dass sie eine Tätigkeit im Bereich der Informatik haben. Eine Person hat angegeben, dass sie linkshändig ist. Auf einer Skala von 1 bis 5 wurde angegeben, wie viel Erfahrung sie mit 3D-Modelliersoftware und AR Anwendungen haben. Die Erfahrung der Teilnehmer mit AR-Erlebnissen und mit 3D-Modelliersoftware waren vielfältig. Neun Teilnehmer haben angegeben, dass sie mindestens einmal die Woche 3D-Videospiele spielen. Die Gesamtzeit pro Teilnehmer, einschließlich Vorfragebögen, Anweisungen, Aufgabenstellung, Nachfragebögen und Nachbesprechung, betrug 40 - 50 Minuten.

6.2 Testplattform

Für die Aufgabenstellung wurde ein iPhone 11 verwendet, welches eine Bildschirmdiagonale von 15,5cm hat. Die Auflösung beträgt 1792 x 828 Pixel bei 326 ppi und hat eine Bildwiederholungsrate von 60 Hz. Das Tracking des Gerätes läuft mithilfe von Unity über ARKit.

6.3 Aufgaben

Um die Mehrfachauswahl-Methode in verschiedenen Szenarien testen zu können, wurden sechs Aufgaben entworfen, bei dem die 3D-Objekte auf unterschiedlicher Weise angeordnet waren. Ziel war es verschieden farbige 3D-Objekte auszuwählen und diese auf die jeweils gleichfarbige Plattform zu platzieren. Es wurden drei Aufgabetypen entwickelt:

Starter: Diese Aufgabe dient zur Eingewöhnung in das Aufgabensystem. Die Plattformen haben eine Größe von $50 \times 50 \text{ cm}$. Die 3D-Objekte haben die Geometrie eines Würfels und eine Seitenlänge von 8 cm. Diese Geometrie und Größe der 3D-Objekten ist in allen Aufgaben gleich. In folgender Abbildung ist eine Darstellung der Anordnung zu sehen:

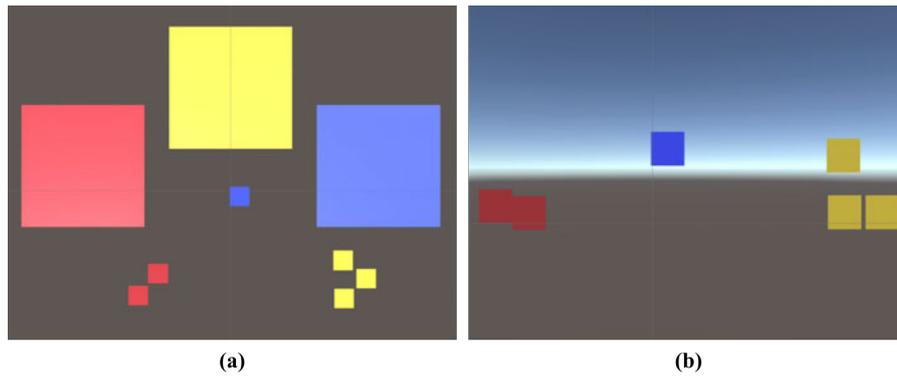


Abbildung 6.1 - Darstellung der Starteraufgabe. (a) zeigt eine Draufsicht der Aufgabe und (b) eine Ansicht von vorne.

Quelle: Eigene Darstellung

Geordnet: Für diesen Aufgabentyp wurden drei Aufgaben erstellt, die sich einander ähneln. Hierbei variiert nur die Anzahl der 3D-Objekte, um zu prüfen, wie sich die Menge der auszuwählenden Objekte auf die Mehrfachauswahl-Methoden auswirkt. In Abbildung 6.2 ist die Objektanordnung der drei Aufgaben zu sehen. Bei Aufgabe (6.2a) und (6.2c) sind fünf Objekte übereinander positioniert, während bei (6.2b) nur vier Objekte übereinander positioniert sind. Die Aufgaben haben also folgende Objektanzahlen: (6.2a) („geordnet hoch“) $5 \times 5 \times 3 = 75$, (6.2b) („geordnet mittel“) $4 \times 3 \times 3 = 36$, (6.2c) („geordnet wenig“) $5 \times 1 \times 3 = 15$. Bei allen diesen Aufgaben liegen die Abstände zwischen den Objekten bei 3cm . Die Plattformen haben eine Größe von $60 \times 60\text{cm}$.

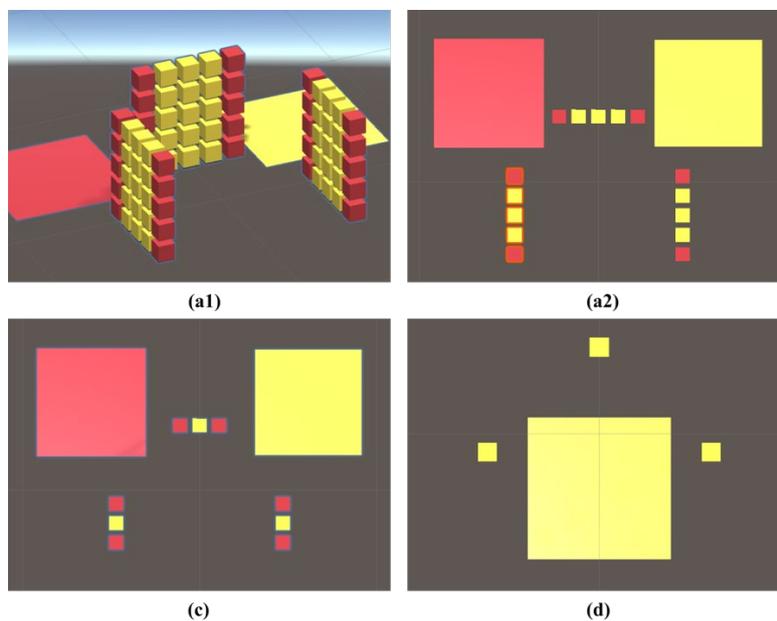


Abbildung 6.2 – Darstellung der Reihen-Aufgaben in Unity. (a1) zeigt eine schiefe Ansicht der Aufgabe. (a2), (b) („geordnet mittel“) und (c) („geordnet wenig“) zeigen eine Draufsicht der drei „geordnet“-Aufgaben.

Quelle: Eigene Darstellung

Ungeordnet: Bei diesen Aufgaben soll überprüft werden, wie sich die Mehrfachauswahl-Methoden verhalten, wenn die auszuwählenden Objekte ungeordnet positioniert sind. In zwei verschiedenen Aufgaben wurde die Anzahl der Objekte, und somit auch die Dichte der Objekte um das Objekt herum, variiert. Aufgabe (6.3a) („ungeordnet“) hat 22 Objekte und Aufgabe (6.3b) („ungeordnet wenig“) 13 Objekte, die auf die Plattformen positioniert werden müssen. Die Plattformen haben auch hier eine Größe von $50 \times 50 \text{ cm}$.

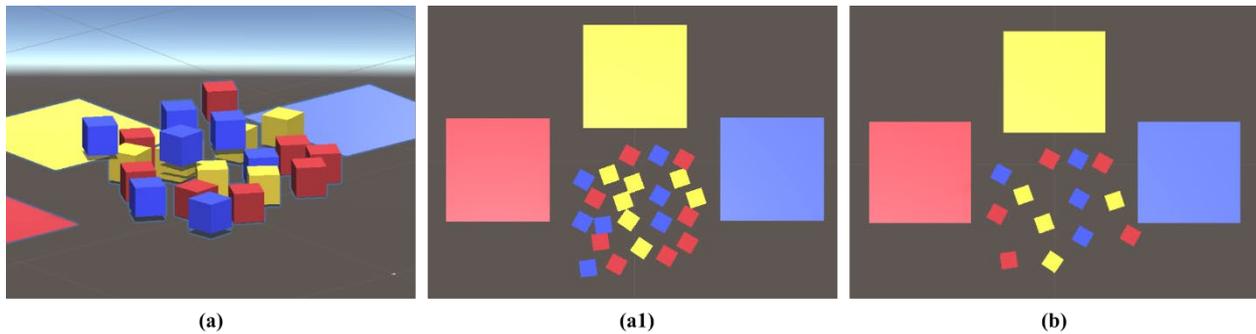


Abbildung 6.3 – (a) zeigt die „ungeordnet“ Aufgabe und (b) die „ungeordnet wenig“ Aufgabe

Quelle: Eigene Darstellung

Wichtig zu erwähnen ist, dass verschieden farbige Objekte nicht übereinander positioniert sind. Dies hat den Grund, dass die Box-Select Methode nicht selektiv einzelne Objekte in der Höhe auswählen kann, ohne andere Objekte darunter auszuwählen.

6.4 Methode

Für die Studie haben alle Teilnehmer dieselben vier Mehrfachauswahl-Methoden benutzt. Die Teilnehmer mussten für jede Methode jeweils sechs Aufgaben lösen. Pro Teilnehmer gab es also 24 Aufgaben, die absolviert werden mussten. Die Methoden wurden in einer zufälligen Reihenfolge vorgelegt, um einen Lerneffekt entgegenzuwirken. Während zu Beginn eines Durchlaufs stets die Starter-Aufgabe abgespielt wurde, hatten die restlichen Aufgaben auch eine zufällige Reihenfolge.

Vor der Aufgabenstellung haben die Teilnehmer ein Bogen mit demografischen Fragestellungen ausgefüllt.

Zunächst wurden in einer Sandboxumgebung mit unterschiedlich angeordneten 3D-Objekten, alle Mehrfachauswahl-Methoden vorgestellt. Es wurde erklärt, wie die Mehrfachauswahl-Methoden funktionieren und dabei wurden die besten Strategien für die Nutzung dieser Methoden nicht aufgeführt. Die Methoden durften so lange ausprobiert werden, bis die Teilnehmer empfanden, dass sie mit der Anwendung der Methoden zurechtkommen würden. Circa fünf bis zehn Minuten haben sich die Teilnehmer in diesem Modus aufgehalten.

Daraufhin wurde das Aufgabensystem für die Aufgaben initiiert und das Handheld-Gerät (iPhone 11) dem Teilnehmer übergeben. Die Aufgaben wurden nacheinander - also ohne Pause - abgeschlossen.

Teilnehmer durften sich während der Aufgabenstellung frei bewegen. Allerdings mussten sie vor dem Start einer Aufgabe wieder zurück an die Startposition gehen. Nach Durchlauf der Aufgaben mit einer Mehrfachauswahl-Methode durfte eine kurze Pause eingelegt werden. Die Pause dauerte üblicherweise nicht länger als eine Minute an. Kommentare der Teilnehmer zu den Mehrfachauswahl-Methoden und eigene Beobachtungen wurden während des Ablaufs dokumentiert.

Da für diese Arbeit von Interesse ist wie die Benutzerfreundlichkeit der Mehrfachauswahl-Methoden ausfällt, haben die Teilnehmer nach der Aufgabenstellung einen System-Usability-Scale (SUS) [35] Fragebogen ausgefüllt. Somit kann die Benutzerfreundlichkeit schnell und zuverlässig gemessen werden [36]. Zusätzlich wurden die Teilnehmer gebeten die Methoden auf Basis ihrer Präferenz in einer Rangfolge zu ordnen. Schließlich wurde die Zeit für das Lösen der einzelnen Aufgaben gemessen.

6.5 Auswertung

Bei der Auswertung werden, neben der subjektiven und objektiven Ergebnisse, auch die Kommentare der Nutzer und die und die eigene Beobachtung mit Einbegriffen, um ein genaueres Bild über die Methoden und die möglichen Schwierigkeiten zu erhalten. Die Ergebnisse werden im Folgenden in den subjektiven und objektiven Ergebnissen, getrennt, vorgestellt.

6.5.1 Subjektive Ergebnisse

Anhand der SUS-Fragebogen wurde eine Punktzahl ausgerechnet, welche die Benutzerfreundlichkeit der Mehrfachauswahl-Methoden bewertet. Eine höhere Punktzahl wird dabei als eine bessere Wertung betrachtet. Punktezahlen, die unter 68 sind, werden als unterdurchschnittlich bewertet. Um die Ergebnisse besser zu deuten, wird empfohlen die Werte zu normalisieren [34]. Diese Arbeit nutzt für die Deutung der Werte eine der von Jess S. [37] empfohlenen Interpretationsmöglichkeit. Den Werten wird hier ein Adjektiv zugeordnet [38]. In Tabelle 6.1 werden die Mittelwerte der Punktzahlen mit den Standardabweichungen aufgelistet. Die Standardabweichung beschreibt die durchschnittliche Abweichung der Ergebnisse um ihren Mittelwert.

	M	SD	Adjektiv
Rect-Select	92,5	12.99	Best möglich
Drag-Select	86,75	12.29	Best möglich
Sphere-Select	63	12.74	Ok
Box-Select	47,25	17.4	Mangelhaft

Tabelle 6.1 - SUS Punktzahlen der Methoden. (M) Zeigt den Mittelwert der Ergebnisse pro Methode an. (SD) zeigt die Standardabweichung der Punktezahlen an.

Die Rect-Select und die Drag-Select haben bei der Aufgabenstellung besonders gut abgeschnitten und deren Benutzerfreundlichkeit kann als „Best möglich“ beschrieben werden. Von den Teilnehmern

wurden diese Methoden sehr gut aufgenommen. Viele Teilnehmer haben bei diesen Methoden einen Vergleich zu den bekannten Auswahl-Methoden aus der Desktop-Umgebung gezogen und dies erwähnt. Ein Problem bei der Drag-Select Methode war jedoch, dass oft vergessen wurde die Auswahl der Objekte abzuwählen, wenn eine neue Auswahl durchgeführt werden sollte. Einige Teilnehmer haben sich gewünscht, dass die Auswahl automatisch abgewählt wird, wenn eine neue Auswahl getätigt wird und eine bestimmte Zeit überschritten wurde oder wenn die Objekte einmal schon bewegt wurden.

Die Sphere-Select Methode liegt in der Wertung unter dem Durchschnitt und wird mit „Ok“ bewertet. Während bei dieser verstanden wurde, wie Änderung des Anzeigeobjektes funktioniert, hatten einige Teilnehmer Probleme damit, diesen korrekt im 3D-Raum zu positionieren.

Abschließen die Benutzerfreundlichkeit der Box-Select Methode, mit einem SUS-Wert von 47,5, als „Mangelhaft“ bezeichnet werden. Die Box-Select Methode wurde von vielen als sehr frustrierend empfunden. Ein auffallendes Problem war, dass das Aufziehen des Auswahl-Tools öfter fehlgeschlagen ist, weil der „Auswahl“-Button nicht gedrückt gehalten, sondern getippt wurde - dadurch wurde ungewollt der dritte Modus aktiviert. Teilnehmer haben auch erwartet, dass die Bestätigung der Auswahl auch mit einer Eingabe auf den „Bewegen“-Button durchgeführt wird. Das heißt: während die Höhe des Auswahl-Tools bestimmt wurde, haben die Teilnehmer versucht die 3D-Objekte auszuwählen und zu bewegen.

Die Teilnehmer sollten die Methoden auf Basis ihrer Präferenz in einer Rangliste ordnen. Die Rangordnung ist wie folgt ausgefallen:

Rang	Methode
1	Rect-Select
2	Drag-Select
3	Sphere-Select
4	Box-Select

Tabelle 6.2 – Rangordnung der Methoden basierend auf der Präferenz der Teilnehmer.

6.5.2 Objektive Ergebnisse

In Abbildung 6.4 sind die Mittleren Lösungszeiten der Aufgaben für jeder Methode aufgeführt. Hier wurde die Starter Aufgabe bei der Auswertung ausgelassen, denn wegen Auffrischungszwecken durften die Auswahl-Methoden, bei dieser Aufgabe, nochmal ausprobiert werden. Die Aufgabe in die Wertung einzubeziehen, würde das Ergebnis verfälschen. Mit der Drag-Select Methode wurden die Aufgaben am schnellsten gelöst, obwohl es Probleme bei der Auswahl der Objekte gab. Ausnahme war hier die „geordnet wenig“ Aufgabe. Ganz dicht dahinter ist die Rect-Select-Methode.

Wird die Lösungszeit der Drag-Select-Methode als Referenzwert benutzt, fällt auf, dass die Sphere-Select-Methode bei den ungeordneten Aufgaben deutlich besser abschneidet als bei den geordneten

Aufgaben. Die Teilnehmer waren mit der Sphere-Select Methode bei den ungeordneten Aufgaben ähnlich schnell, wie mit den 2D-Mehrfachauswahl-Methoden. Dafür haben die Teilnehmer bei den geordneten Aufgaben am längsten gebraucht diese zu lösen.

Mit der Box-Select Methode haben die Teilnehmer, bei den „ungeordnet“ Aufgaben, im Vergleich zu den anderen Methoden, deutlich länger gebraucht um diese lösen. Die geordneten Aufgaben wurden mit dieser Methode schneller als die Sphere-Select Methoden gelöst, jedoch langsamer als die 2D-Mehrfachauswahl-Methoden.

Interessant war zu beobachten, dass die Teilnehmer die 3D-Mehrfachauswahl-Methoden unterschiedlich effizient nutzen konnten. Einige haben mit wenig Einarbeitung die Methoden effizient nutzen können und andere konnten bis zum Schluss die 3D-Mehrfachauswahl-Methoden nicht korrekt verwenden. Dies ist auch deutlich in Abbildung 6.4 wieder zu erkennen. Dies spiegelt sich auch in den Ergebnissen dieser Methoden wider. Anhand der breiten Standardabweichung kann erkannt werden, dass die Lösungszeiten mit der Methode stärker variieren.

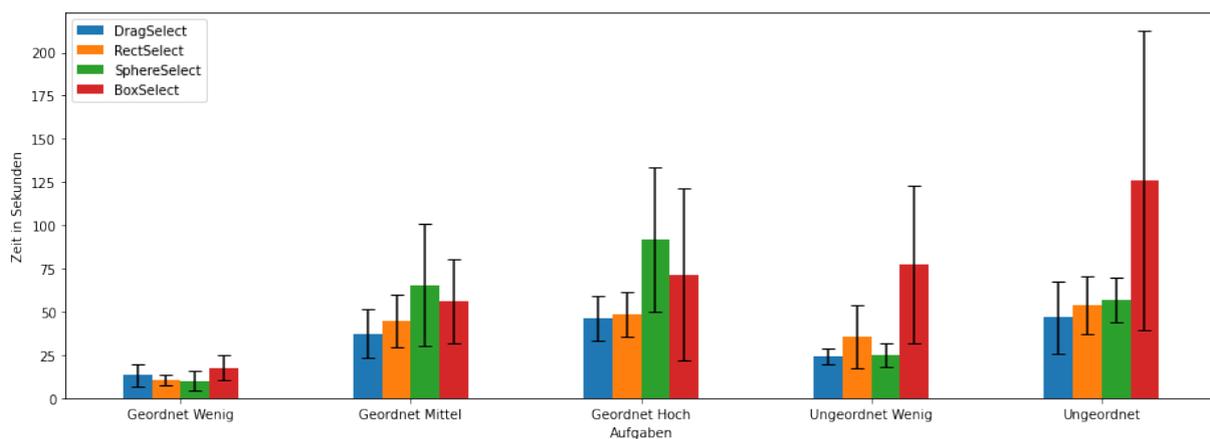


Abbildung 6.4 - Mittlere Lösungszeiten der Aufgaben pro Methode. Die x-Achse stellt die Methoden pro Aufgabe dar. Die y-Achse die Zeit für das lösen der Aufgabe.

6.6 Diskussion

Die Ergebnisse zeigen, dass die 2D-Methoden sich angemessen für die Mehrfachauswahl von 3D-Objekte eignen und wurden deshalb von den Teilnehmern hoch bewertet. 2D-Methoden waren schneller zu erlernen und effizienter als die 3D-Methoden. Mit der Rect-Select Methode hatten die Teilnehmer keine Probleme, da diese Methode für die Teilnehmer bekannt war und wie erwartet funktioniert hat. Die Drag-Select Methode wurde von dem Teilnehmer im SUS Fragebogen schlechter bewertet, waren damit aber schneller beim Lösen der Aufgaben, obwohl die Auswahl öfter neugestartet werden musste. Aus den Beobachtungen geschlossen werden, dass die Erfahrung mit dem Umgang von Touch-Displays

und Ähnlichkeit der Methode zu den bekannten Desktop-Metaphern eine große Rolle spielen, wenn es um die Benutzerfreundlichkeit der 2D-Methoden geht.

Im Gegensatz dazu, haben die 3D-Mehrfachauswahl-Methoden eine unterdurchschnittliche Bewertung der Benutzerfreundlichkeit erhalten. Diese Bewertung kann unterschiedliche Gründe haben:

Zuerst kann man aus den Beobachtungen folgern, dass die Erfahrungen der Teilnehmer mit der Ray-Casting Metapher und somit auch der gerätebasierten Eingabe, eine wichtige Rolle gespielt hat. Einige Teilnehmer haben die Methoden effizienter nutzen konnten als andere und der Unterschied Lösungszeit war dabei größer als bei den anderen Methoden. Würden die Teilnehmer mehr Training erhalten, könnte erwartet werden, dass die Bewertung der Methoden nach oben gehen würde.

Ein weiteres Problem ist, dass die 3D-Methoden aufgrund der Dreidimensionalität komplexer sind und mehr Interaktionsschritte für die Definition des Auswahlbereiches benötigen. Durch diese zusätzlichen Schritte konnten die Teilnehmer, bei der Definition der 3D-Anzeigeobjektes öfter Fehler ausführen Ein auffallendes Beispiel war hier das Aufziehen des Volumens bei der Box-Select Methode. In diesem Fall wurde der zweite Modus ungewollt übersprungen, weil keine Halte-Geste, sondern ein eine Tipp-Geste verwendet wurde, beim Versuch die Bodenfläche zu bestimmen. Teilnehmer haben dies als sehr frustrierend empfunden, welches sich bei der Bewertung der Benutzerfreundlichkeit auch ausgewirkt hat.

Auch spielte die Identifizierung der Auswahlvolumens im 3D-Raum eine Rolle bei der Bewertung. Dies war bei der Sphere-Select Methode deutlich zu erkennen. Teilnehmer konnten manchmal nicht erkennen wo die Kugel im 3D-Raum positioniert ist und haben deshalb länger gebraucht die gewünschte Auswahl durchzuführen.

Beim Entwurf der Box-Select Methode wurde die Rechteck-Auswahl als Vorbild genommen. Dadurch hat diese Methode bei der Definition des Anzeigeobjektes ähnliche Züge, aber nicht dieselbe Funktionalität. Teilnehmer haben, wie bei der Rect-Select Methode erwartet, dass sie die Auswahl nicht noch einmal abwählen müssen. Sie haben versucht, die ausgewählten Objekte zu bewegen, ohne den letzten Schritt der Bestätigung durchzuführen. Dies deutet darauf hin, dass diese Teilnehmer versucht haben, die bekannte Rechteck-Auswahl Metapher auf die 3D-Methode zu übersetzen. Ähnliches

Verhalten konnte auch bei der Drag-Select Methode beobachtet werden. Die meisten Teilnehmer haben bei der Auswahl erwartet, dass sie einzelne Objekte abwählen können, obwohl das nicht möglich war. Hier könnte die Ähnlichkeit zu der Touch-Auswahl der Grund dieser Erwartung sein. Außerdem konnte bei dieser Methode beobachtet werden, dass nachdem die Objekte bewegt wurden, die alte Auswahl nicht abgewählt wurde.

Es kann jedoch nicht daraus geschlossen werden, ob hier eine Erwartung der Teilnehmer nicht erfüllt wurde oder ob sie vergessen haben, die Abwahl durchzuführen.

Letztlich ist aus den Lösungszeiten zwischen den 2D-Methoden kein erkennbarer Einfluss, bei den verschieden geordneten Aufgabentypen, zu erkennen. Deutliche Unterschiede bei den Lösungszeiten gab es dafür aber zwischen der Sphere-Select und Box-Select Methode. Mit der Box-Select Methode konnten die „geordnet“ Aufgaben schneller gelöst werden als mit der Sphere-Select Methode. Da die Sphere-Select Methode keinen länglichen Bereich definieren kann, konnten auch keine länglich platzierten Objekte ausgewählt werden. Das Anzeigeobjekt musste dadurch klein gehalten werden, um die richtige Auswahl zu treffen können. Dies hat die Lösung der Aufgabe deutlich verlangsamt. Anhand der verschiedenen Aufgaben kann bei den 3D-Mehrfachauswahl-Methoden nochmal bestätigt werden, dass die Benutzerfreundlichkeit der Auswahl auch von den Auswahlparametern, die in Kapitel 3.1.1 behandelt werden, abhängig ist.

7 Fazit und Aussicht

Im Rahmen dieser Arbeit wurden vier verschiedene Mehrfachauswahl-Methoden basierend auf bekannten Metaphern und Mehrfachauswahl-Methoden in Handheld-AR implementiert. Diese sind folgende:

- Rect-Select Methode mit der Rechteck-Auswahl Metapher als Grundlage
- Drag-Select Methode mit der Pinsel Metapher als Grundlage
- Box-Select Methode mit der Rechteck-Auswahl Metapher als Grundlage, aber in 3D Form
- Sphere-Select Methode mit der Sphere-Cast Metapher als Grundlage

Anhand einer Nutzerstudie wurden die vier Methoden evaluiert, um herauszufinden wie sich die Benutzerfreundlichkeit zwischen den 2D-Mehrfachauswahl-Methoden und 3D-Mehrfachauswahl-Methoden in Handheld-AR unterscheidet.

Die Ergebnisse zeigen, dass im Vergleich zu 3D-Mehrfachauswahl-Methoden die 2D-Methoden, für Nutzer, die wenig Erfahrung mit Handheld-AR haben, eine bessere Benutzerfreundlichkeit besitzen können. Der Nutzer muss keine neue Eingabemöglichkeit erlernen und kann bekannte Metaphern der 2D-Auswahl direkt übersetzen. 3D-Mehrfachauswahl-Methoden hingegen sind aufgrund ihrer höheren Dimensionalität komplexer welche die Nutzung erschweren kann. Dazu kommt, dass die Position des Auswahl-Tools im 3D-Raum identifiziert werden muss, welches eine weitere Schwierigkeit mit sich bringt. Werden die Probleme der Komplexität und Identifizierung des Auswahl-Tools weiter untersucht und die Methoden weiterentwickelt, so kann hier die Benutzerfreundlichkeit für die Auswahl verbessert werden. Letztlich kann hier aber behauptet werden, dass 2D-Mehrfachauswahl-Methoden für simple Anwendungsfälle, wie dieser der in der Einleitung ausgemalt wurde, eine ausreichende Gebrauchstauglichkeit haben.

Weiter sollte in dieser Arbeit untersucht werden, welchen Einfluss die Auswahl Metapher auf die Nutzung der Mehrfachauswahl-Methoden hat. Aus den Beobachtungen zeigte sich, dass Mehrfachauswahl-Methoden, die ähnlich zu den bekannten Auswahl Metaphern sind, dem Nutzer eine gewisse Erwartung auf die Funktionalität der Methode geben. Wird eine bekannte 2D-Mehrfachauswahl-Metapher in Handheld-AR genutzt, so kann der Nutzer diesen schneller verstehen. Die Erwartung zeigte sich auch bei der Box-Select Methode, die eine Ähnlichkeit zu der Rechteckauswahl-Metapher hat. Fraglich ist aber noch, inwieweit 2D-Metapher bei der Nutzung von 3D-Methoden behilflich sind. Weitere Untersuchungen sind hier nötig, um eine Aussage treffen zu können.

Für weiteren Untersuchungen der Mehrfachauswahl-Methoden in Handheld-AR wäre es interessant zu erfahren, wie die Identifizierung der Position von Auswahl-Tools, die im 3D-Raum positioniert sind, verbessert werden kann. Ein Ansatz wäre es mit visuellem Feedback die Entfernung des Auswahl-Tools anzudeuten.

Damit die Funktionalitäten von Handheld-AR voll ausgenutzt werden können, sollte auch die Multi-Touch Fähigkeit bei Entwicklung von weitere Mehrfachauswahl-Methoden in Betracht gezogen werden.

Die 3D-Methode, die für diese Arbeit entwickelt wurde, benutzen die Ray-Cast Metapher, um die Anzeigeobjekte im Raum zu platzieren. Es bietet sich hier an andere bekannte Manipulationsmethoden zu benutzen um, die Anzeigeobjekte im Raum zu bewegen. Wie zum Beispiel die Go-Go Methode, welche eine nicht lineare Positionierung von 3D-Objekten mit erlaubt [39].

Außerdem wurde während der Studie von einigen Teilnehmer bemängelt, dass das visuelle Feedback für die Auswahl der Objekte nicht ausreichend war und sie sich ein haptisches Feedback gewünscht haben, wenn eine Auswahl bestätigt wurde. Deshalb sollte dieser Effekt von Feedback auf die Mehrfachauswahl weiter untersucht werden.

Schlussendlich kann die Mehrfachauswahl in Handheld-AR sehr viel Potenzial haben, denn durch die immer besser werdenden Technologien können auch immer komplexere Szenen in AR dargestellt werden. Neue Anwendungsfälle können zum Vorschein kommen, bei dem die Mehrfachauswahl, als wichtiges Instrument, gebrauch findet.

Literaturverzeichnis

- [1] „IKEA Place“. Zugegriffen: 25. Juli 2022. [Online]. Verfügbar unter: <https://apps.apple.com/us/app/ikea-place/id1279244498>
- [2] J. J. LaViola, E. Kruijff, R. P. McMahan, D. Bowman, und I. P. Poupyrev, *3D User Interfaces: Theory and Practice*. Addison-Wesley, 2017. Zugegriffen: 28. Mai 2022. [Online]. Verfügbar unter: <https://pub.h-brs.de/frontdoor/index/index/docId/3009>
- [3] J. Yin, C. Fu, X. Zhang, und T. Liu, „Precise Target Selection Techniques in Handheld Augmented Reality Interfaces“, *IEEE Access*, Bd. 7, S. 17663–17674, 2019, doi: 10.1109/ACCESS.2019.2895219.
- [4] E. S. Goh, M. S. Sunar, und A. W. Ismail, „3D Object Manipulation Techniques in Handheld Mobile Augmented Reality Interface: A Review“, *IEEE Access*, Bd. 7, S. 40581–40601, 2019, doi: 10.1109/ACCESS.2019.2906394.
- [5] R. Azuma, Y. Baillot, R. Behringer, S. Feiner, S. Julier, und B. MacIntyre, „Recent advances in augmented reality“, *IEEE Comput. Graph. Appl.*, Bd. 21, Nr. 6, S. 34–47, Nov. 2001, doi: 10.1109/38.963459.
- [6] P. Milgram, H. Takemura, A. Utsumi, und F. Kishino, „Augmented reality: A class of displays on the reality-virtuality continuum“, *Telem manipulator Telepresence Technol.*, Bd. 2351, Jan. 1994, doi: 10.1117/12.197321.
- [7] R. Dörner, W. Broll, P. Grimm, und B. Jung, *Virtual und Augmented Reality (VR/AR)*, 2. Aufl. Springer. Zugegriffen: 31. Mai 2022. [Online]. Verfügbar unter: <https://link.springer.com/book/10.1007/978-3-662-58861-1>
- [8] R. T. Azuma, „A Survey of Augmented Reality“, *Presence*, Bd. 6, Nr. 4, S. 355, 1997.
- [9] M. Billinghurst, A. Clark, und G. Lee, „A Survey of Augmented Reality“, *Found. Trends® Human–Computer Interact.*, Bd. 8, Nr. 2–3, S. 73–272, März 2015, doi: 10.1561/11000000049.
- [10] „GPS SPS Performance Standard“, United States. Dept. of Defense, Technische Dokumentation 5. Edition. Zugegriffen: 20. Juni 2022. [Online]. Verfügbar unter: <https://www.gps.gov/technical/ps/#spgps>
- [11] D. Chatzopoulos, C. Bermejo, Z. Huang, und P. Hui, „Mobile Augmented Reality Survey: From Where We Are to Where We Go“, *IEEE Access*, Bd. 5, S. 6917–6950, 2017, doi: 10.1109/ACCESS.2017.2698164.

- [12] J. Bauer, N. Sünderhauf, und P. Protzel, „Comparing several implementations of two recently published feature detectors“, 2007.
- [13] C.-M. Tsai, J.-Y. Li, P. Han, und C.-T. Yen, „Design and Evaluation of Optical See-Through Head-Mounted Display With Wide FOV Based on Dihedral Corner Reflector Array“, *IEEE Access*, Bd. 9, S. 118977–118984, 2021, doi: 10.1109/ACCESS.2021.3107476.
- [14] Y. Ghazwani und S. Smith, „Interaction in Augmented Reality: Challenges to Enhance User Experience“, in *Proceedings of the 2020 4th International Conference on Virtual and Augmented Reality Simulations*, New York, NY, USA, Feb. 2020, S. 39–44. doi: 10.1145/3385378.3385384.
- [15] E. Bozgeyikli, „Locomotion in Virtual Reality for Room Scale Tracked Areas“, *Tampa Grad. Theses Diss.*, Nov. 2016, [Online]. Verfügbar unter: <https://digitalcommons.usf.edu/etd/6470>
- [16] J. D. Foley, V. L. Wallace, und P. Chan, „The human factors of computer graphics interaction techniques“, *IEEE Comput. Graph. Appl.*, Bd. 4, Nr. 11, S. 13–48, Nov. 1984, doi: 10.1109/MCG.1984.6429355.
- [17] J. F. Lucas, „Design and Evaluation of 3D Multiple Object Selection Techniques“, Thesis, Virginia Tech, 2005. Zugegriffen: 28. Mai 2022. [Online]. Verfügbar unter: <https://vtechworks.lib.vt.edu/handle/10919/31769>
- [18] D. A. Bowman und L. F. Hodges, „Formalizing the Design, Evaluation, and Application of Interaction Techniques for Immersive Virtual Environments“, *J. Vis. Lang. Comput.*, Bd. 10, Nr. 1, S. 37–53, Feb. 1999, doi: 10.1006/jvlc.1998.0111.
- [19] F. Argelaguet und C. Andujar, „A survey of 3D object selection techniques for virtual environments“, *Comput. Graph.*, Bd. 37, Nr. 3, S. 121–136, Mai 2013, doi: 10.1016/j.cag.2012.12.003.
- [20] I. Poupyrev und T. Ichikawa, „Manipulating Objects in Virtual Worlds: Categorization and Empirical Evaluation of Interaction Techniques“, *J. Vis. Lang. Comput.*, Bd. 10, Nr. 1, S. 19–35, Feb. 1999, doi: 10.1006/jvlc.1998.0112.
- [21] B. Preim und R. Dachsel, *Interaktive Systeme*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2015. doi: 10.1007/978-3-642-45247-5.
- [22] A. Steed und C. Parker, „3D Selection Strategies for Head Tracked and Non-Head Tracked Operation of Spatially Immersive Displays“, S. 8.
- [23] R. Kopper, F. Bacim, und D. A. Bowman, „Rapid and accurate 3D selection by progressive refinement“, in *2011 IEEE Symposium on 3D User Interfaces (3DUI)*, März 2011, S. 67–74. doi: 10.1109/3DUI.2011.5759219.

- [24] J. S. Pierce, A. S. Forsberg, M. J. Conway, S. Hong, R. C. Zeleznik, und M. R. Mine, „Image plane interaction techniques in 3D immersive environments“, in *Proceedings of the 1997 symposium on Interactive 3D graphics*, New York, NY, USA, Apr. 1997, S. 39-ff. doi: 10.1145/253284.253303.
- [25] G. J. Wills, „Selection: 524,288 ways to say ‚this is interesting‘“, in *Proceedings IEEE Symposium on Information Visualization '96*, Okt. 1996, S. 54–60. doi: 10.1109/INFVIS.1996.559216.
- [26] R. Stenholt, „Efficient selection of multiple objects on a large scale“, in *Proceedings of the 18th ACM symposium on Virtual reality software and technology*, New York, NY, USA, Dez. 2012, S. 105–112. doi: 10.1145/2407336.2407357.
- [27] „Adobe Photoshop“. Adobe. Zugegriffen: 26. Juli 2022. [Online]. Verfügbar unter: <https://www.adobe.com/de/products/photoshop.html>
- [28] *Understanding ARKit Tracking and Detection - WWDC18 - Videos*. Zugegriffen: 9. Juli 2022. [Online Video]. Verfügbar unter: <https://developer.apple.com/videos/play/wwdc2018/610/?time=916>
- [29] „Fundamental concepts | ARCore“, Google LLC. Zugegriffen: 9. Juli 2022. [Online]. Verfügbar unter: <https://developers.google.com/ar/develop/fundamentals>
- [30] „Unity“. Unity Technologies. [Online]. Verfügbar unter: <https://unity.com/de>
- [31] „AR Foundation 4.2.2“, Unity Technologies, Manual 32131231. Zugegriffen: 8. Juli 2022. [Online]. Verfügbar unter: <https://docs.unity3d.com/Packages/com.unity.xr.arfoundation@4.2/manual/index.html>
- [32] C. Wilkes, „Lean Touch+“. Zugegriffen: 8. Juli 2022. [Online]. Verfügbar unter: <https://assetstore.unity.com/packages/tools/input-management/lean-touch-72356>
- [33] K. Kuzyk, „AR Foundation Editor Remote“. Zugegriffen: 8. Juli 2022. [Online]. Verfügbar unter: <https://assetstore.unity.com/packages/tools/utilities/ar-foundation-editor-remote-168773>
- [34] J. Zimmer, „RTS Style Unit Selection in Unity 5“, *Jeff Zimmer*, 28. April 2015. <https://hyunkell.com/blog/rts-style-unit-selection-in-unity-5/> (zugegriffen 10. Juli 2022).
- [35] J. Brooke, „SUS: A quick and dirty usability scale“, *Usability Eval Ind*, Bd. 189, Nov. 1995.
- [36] A. S. for P. Affairs, „System Usability Scale (SUS)“, 6. September 2013. <https://www.usability.gov/how-to-and-tools/methods/system-usability-scale.html> (zugegriffen 17. Juli 2022).
- [37] J. S. PhD, „5 Ways to Interpret a SUS Score – MeasuringU“. <https://measuringu.com/interpret-sus-score/> (zugegriffen 19. Juli 2022).
- [38] „Determining What Individual SUS Scores Mean: Adding an Adjective Rating ScaleJUX“, 7. Mai 2009. <https://uxpajournal.org/determining-what-individual-sus-scores-mean-adding-an-adjective-rating-scale/> (zugegriffen 19. Juli 2022).

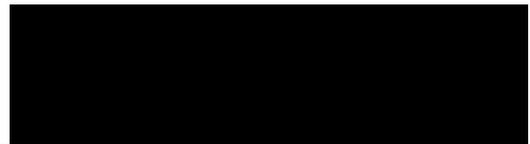
[39] I. Poupyrev, M. Billinghurst, S. Weghorst, und T. Ichikawa, „The go-go interaction technique: non-linear mapping for direct manipulation in VR“, in *Proceedings of the 9th annual ACM symposium on User interface software and technology*, New York, NY, USA, Nov. 1996, S. 79–80. doi: 10.1145/237091.237102.

Eigenständigkeitserklärung

Hiermit versichere ich, dass ich die vorliegende Bachelorarbeit mit dem Titel:

selbständig und nur mit den angegebenen Hilfsmitteln verfasst habe. Alle Passagen, die ich wörtlich aus der Literatur oder aus anderen Quellen wie z. B. Internetseiten übernommen habe, habe ich deutlich als Zitat mit Angabe der Quelle kenntlich gemacht.

Datum



Unterschrift